6DoF Motion Estimation for UAV Landing on a Moving Shipdeck using Real-Time On-Board Vision

Shanggang Lin¹, Matt Garratt¹, Andrew Lambert¹, and Ping Li¹
¹UNSW Australia, Canberra, ACT 2612, Australia
{Shanggang.Lin,Ping.Li@student.adfa.edu.au},{M.Garratt,A.Lambert}@adfa.edu.au

Abstract

We present a vision system for UAV shipboard landing applications. It aims to detect and track the international landing marker and measures the relative 6DoF pose of the shipdeck with respect to the vehicle. An additional vision sensor is adopted to increase redundancy. Real-time on-board flight tests with achieved closed-loop control demonstrate that the vision system is fast, precise and capable of estimating 6DoF motion of a moving shipdeck while performing target tracking, which proves the feasibility and quality of our approach.

1 Introduction

We aim at developing a feasible way for autonomously landing an Unmanned Aerial Vehicles (UAV) on a ship’s moving flight deck, after it completes operations such as coastline patrol, reconnaissance, surveillance and payload delivery. Among those different types of UAVs, fixed-wing UAVs travel at higher speeds and fly for longer ranges than rotary-wing UAVs (RUAVs), but require runway that small ships don’t have for taking-off and landing in many scenarios. In contrast, RUAVs including multi-rotor UAVs and conventional helicopters are able to perform vertical take-off and landing (VTOL), low-speed flight, low-altitude hovering and are highly maneuverable. These unique abilities make them particularly suitable for small ships, thus we target RUAV operations in our work.

The whole landing procedure consists of several stages, each stage comes with critical challenges. First, a proper landing area must be pre-located before approaching to the deck. Second, once located the UAV should be able to lock on that specific area and start following the ship. Then the relative pose between the UAV and the deck is to be properly estimated. Finally, a safe trajectory must be generated to finalize landing. We consider the first and second stages as landing target recognition and tracking problem, treat the third stage as pose estimation problem, and regard the final stage as a combination of path planning and quiescent period prediction problem. At the current stage of development, we aim to solve the first two problems and leave the last one for our future research.

We address the necessity of developing a vision system to assist UAVs landing on shipdeck, which provides reliable target tracking and pose estimation, whilst being able to deal with a partially occluded target caused by rain, fog or shipdeck infrastructures. In this paper a further improved vision system based on our previous development is proposed [Lin et al., 2015]. While maintaining robustness, precision, real-time implementation and the capability of dealing with occlusion, the vision system has the ability to track the international landing marker and estimate the relative 6 degrees of freedom (6DoF) pose between the UAV and the deck during continuous shipdeck movements, where its feasibility is demonstrated via closed-loop flight tests with on-board vision processing. To our best knowledge, other research groups haven’t reached this stage yet.

This paper is constructed as follows: in Section 2 the relevant state of the art is introduced, then the vision system is explained in detail in Section 3. Section 4 and 5 show the experiment platform and discuss the flight test results.

2 Related Work

Some previous achievements require additional on-deck infrastructures to assist VTOL UAVs landing, which include the UAV Common Automatic Recovery System (UCARS) provided by Sierra Nevada Corp. and other systems based on high precision differential GPS (DGPS) and beacons [S. N. Corporation, 2006]. Although these systems can provide precise relative pose measurement of the shipdeck, they need to have a sub-system on the deck to transmit the measured data to the UAV by radio signals. Such configuration prevents an UAV from landing on an unequipped ship, which is
the main disadvantage of these systems. Moreover, the expensive cost prohibits the widespread use of these systems.

Instead, using vision sensors in guiding an UAV toward the shipdeck and land has become a research hotspot recently. The advantages of vision sensors are light weight, low cost, and easy operation. They can provide rich information about the shipdeck environment for visual processing. Moreover, they can couple with other sensor setups such as Inertial Measurement Units (IMU) or GPS to form more reliable measurements.

Some vision-based approaches for locating the landing area rely on having artificial, customized markers on the deck, with distinguishing features that have a strong contrast to the background (usually painted in black color within a white background or vice versa). Common shapes used for markers are squares, circles and other geometric patterns [Xu et al., 2006; Lee et al., 2012]. After thresholding, certain image processing techniques such as corner detection and labeling, ellipse fitting and contour detection are applied to locate the marker within the image. For example, Xu et al. make use of a "T" shaped pattern with an infrared camera to detect the pose and position of the ship deck [Xu et al., 2009]. Arora et al. apply a LIDAR-based technique for tracking a standard US Naval shipdeck marking, which declares that it is infrastructure free, but multiple sensors are needed including active laser to increase the redundancy [Arora et al., 2013]. Previously our research group developed a system based on LIDAR in conjunction with a single beacon to locate the shipdeck [Garratt et al., 2009]. Although the works summarized here have demonstrated the capability of the vision systems, having a customized marker on the deck may interfere with manned landing operations which have strict standards for deck markings and lighting arrangements.

An alternative to custom markers is to use the international landing pattern, which consists of a letter “H” and a surrounding circle. Beneficially, RUAVs can land on ships equipped with such a pattern whilst not interfering with the existing manned helicopter operations. An early approach by Saripalli et al. performed invariant moment calculation on a landing marker which has an individual “H” letter, but the method is sensitive to size variation and rotation [Saripalli et al., 2003]. Shi et al. treat a similar “H” marker by means of corner labeling and rectangle fitting, using a method which is able to recognize a slightly incomplete marker [Shi and Wang, 2009]. However, both of the works only deal with the “H” letter. None of them has proven applicable to recognize an international landing marker. Yang et al. apply the solution of solving the sign detection problem to detect the international landing marker [Yang et al., 2009]. Evidence show that they can locate the marker lying within a cluttered background, but it has to be fully presented (Figure 1(a)). Sanchez-Lopez et al. introduce an Artificial Neural Network (ANN) with a 7-layers decision tree to recognize the marker in a cluttered environment (Figure 1(b)), but only achieve 6fps on a laptop computer equipped with an Intel i3 CPU due to the heavy-processing load and no on-board vision processing or closed-loop flight tests is proposed [Sanchez-Lopez et al., 2014]. We address the need to have a marker detection method which is accurate, robust to random occlusion and running at a much faster speed.

Figure 1: The international landing marker: (a) Marker detection based on connected component method [Yang et al., 2009]. (b) Marker recognition with occlusion using Artificial Neural Network [Sanchez-Lopez et al., 2014].

In order to land, the vision system should have the ability to estimate the relative pose between the shipdeck and itself after successfully locating the landing marker. The precision of pose estimation is of crucial importance to the control system. Yakimenko et al. solve the pose estimation problem by means of contributing a solution to the P3P (Perspective-3-Points) problem based on three infrared reference points on a runway [Yakimenko et al., 2002]. In Saripalli’s work it obtains 3DoF pose estimation from vision, but relies on the GPS’s vehicle height information as a precondition [Saripalli et al., 2003]. Their method for estimating x and y positions is also adopted in Shi’s implementation [Shi and Wang, 2009]. Xu et al. derive the 3D positions and yaw angle of their landing marker based on stereo triangulation, but no pitch or roll movement is involved [Xu et al., 2006]. Eberli et al. propose a 5DoF pose estimation method based on an ANN with the features of two concentric circles, whereas the heading of the UAV still remains ambiguous [Eberli et al., 2011]. In [Yang et al., 2009], the 5DoF pose estimation is achieved by means of ellipse fitting, whilst the yaw angle is calculated from the “H” pattern. However, it may produce a much larger error if distortion occurs on the ellipse, especially when the ellipse is occluded. The works summarized above assume the landing marker to be stationary or only has several DoF motion. Sanchez-Lopez’s work is the first to consider landing on a 6DoF moving platform [Sanchez-Lopez et al., 2014]. The relative pose
estimation is mainly based on solving the Perspective-n-Points (PnP) problem, which requires at least the “H” pattern to be complete so that the co-planar corresponding points can be located [Lepetit et al., 2009]. Nevertheless, PnP is very sensitive to image pixel error. In our work, we only introduce PnP as a supplementary method to work out pitch and roll angles, whilst adopting another vision-based approach to calculate the relative 3D positions and yaw angle. Moreover, the target detection output minimizes the pixel error of the corresponding points, which guarantees the precision of measurement. An evaluation scheme in conjunction with an additional vision sensor are adopted to further increase the precision of pose estimation and redundancy of the system.

3 The Vision Approach

Our vision system consists of six stages: image pre-processing, line segment processing, cascade filtering, target refinement, information gathering with pattern reconstruction and pose estimation. The output of the system will be the 3D positions as well as the pitch and roll angles of the landing marker w.r.t the UAV, and the heading (yaw angle) of the vehicle. Figure 2 is a flow chart of the vision system.

3.1 Landing Marker Analysis

Due to the symmetry of the “H” marker, we can categorize the 12 line segments into four different groups due to their positions and lengths. Moreover, the perpendicularity and parallelity of the line groups are other important properties. During large shipdeck motions, perspective distortion may degrade these features. But we can still use such properties to a certain extent. Before the experiment we take a scaled international landing marker and measure the physical length of each line group as well as the proportion of lengths of every two connected lines, as illustrated in Figure 3 and Table 1. We will make use of them in the following sections.

<table>
<thead>
<tr>
<th>color</th>
<th>blue</th>
<th>yellow</th>
<th>green</th>
<th>red</th>
</tr>
</thead>
<tbody>
<tr>
<td>value</td>
<td>4.27</td>
<td>1.146</td>
<td>1.45</td>
<td>0.688</td>
</tr>
</tbody>
</table>

Figure 3: Properties of the international landing marker.

Table 1: Look-up table for proportion calculation

3.2 Image Pre-processing

It’s a common approach to paint the landing marker in bright color within a dark shipdeck background. Therefore, two significant peaks appear in the histogram. The pixels gathered around the left most peak stand for the background and those gathered around the right most peak represent the marker and other bright objects in the scene. Our vision system calculates an adaptive threshold by averaging the recorded intensity values of the peaks and uses it to binarize the image (See Figure 4(a)-(c)).

3.3 Line Segment Processing

The intensity-based features or corners are less readily detected after image binarization. However, the lines remain easily segmented. We present a novel solution for finding the markers representation by means of line segment detection and feature point mapping. The image is processed through the Edge Drawing line segment detection algorithm (EDLines) [Akinlar and Topal, 2011]. The dominant advantage is that it balances the processing speed and the quality of the produced line segments, which outperforms other parameter-based approaches such as Hough Transform and meets our task’s real-time requirement. The output image after line segment extraction can be seen in Figure 5(a).
Each detected line segment contains a start-point \((x_1, y_1)\) and an end-point \((x_2, y_2)\) (units in pixel coordinates). The slope \(k\) and length \(l\) of it are calculated using (1) and (2):

\[
k = \frac{y_2 - y_1}{x_2 - x_1}. \tag{1}
\]

\[
l = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}. \tag{2}
\]

In order to find the line segments representing the whole marker and the “H” pattern, we propose a spatial feature point mapping method to find the key expression of a line segment, then apply density-based clustering followed by a coarse-to-fine filtering approach to isolate the marker from other objects. The mid-point of each line segment in associated with the calculated information are regarded as a feature point, where the mapping results is shown in Figure 5(b) (each feature point is indicated by a small white circle). A yellow circle and a red ellipse are used to indicate the feature points from the whole marker and the “H” pattern, respectively. As contamination is more likely to happen on the outer part of the marker, we are more interested in the “H” pattern itself since the number of points and the shape and distribution of the point pattern are robust to scaling, rotation and perspective distortion. For clustering the points using density feature, there are two involved parameters: \(\text{eps}\) and \(\text{minPts}\), which stand for the maximum distance between point pairs (unit in pixels) and the minimum number of points required to form a cluster, respectively. We choose a fixed value 7 for \(\text{minPts}\), and introduce (3), which is derived from the marker-altitude characterization process in the experiment, to calculate an adaptive \(\text{eps}\):

\[
\text{eps} = \text{round}(\frac{22 \times h + 62.36}{h - 0.4291}). \tag{3}
\]

where \(h\) stands for the altitude of the vehicle.

3.4 Cascade Filtering

We design four layers of filtering as a coarse-to-fine approach to gradually isolate the marker from other irrelevant objects. The first layer make uses of the measured number of feature points within the cluster to reject the clusters which are either too large or too small. Hence, we use (4) obtained in the experiment to calculate the criteria within an upper and lower bound:

\[
s_{1_{\text{num}}} = \text{round}(\frac{290}{h + 3.117} \pm \delta_1) \quad (\delta_1 = 20). \tag{4}
\]

where \(h\) denotes the altitude of the vehicle and \(\delta_1\) is the volatility. We choose 20 to be the optimal \(\delta_1\) value according to various attempts.

The second layer constructs a small region of interest (ROI), centre of which coincides with each candidate cluster centre and size of which is calculated according to (5):

\[
s_{2_{\text{roi}}} = \text{round}(\frac{606.9}{h + 0.2071}). \tag{5}
\]

where \(h\) is the altitude of the vehicle. The proportion of the white pixels \(p_w\) w.r.t the total number of pixels in the ROI \(p_t\) is selected as the criteria for the second layer using (6). Experiments show that a 30% value of threshold, which is the value that of a perfectly fitted ROI with the marker inside, with a 10% of volatility \((\delta_2)\) is sufficient for filtering.

\[
s_{2_{\text{prop}}} = \frac{p_w}{p_t} \pm \delta_2 \quad (\delta_2 = 0.1). \tag{6}
\]

The third layer makes use of the knowledge that the majority of the line segments from the marker should be
within the ROI. If more than 70% of the line segments are partially or completely inside the ROI, we consider the cluster as a candidate.

The fourth layer, which is also the final and most important one, relies on the geometrical properties of the marker and finalizes target recognition based on the analysis of connectivity, perpendicularity and parallelity analysis. Every two connected line segments are evaluated by the included angle. For the “H” pattern, the angle value is close to 90° (with a tolerance $\sigma$) and the lines are perpendicular to each other. All the connected line segments result in several line segment “chains” belonging to different parts of the marker. By examining them the vision system is able to uniquely identify the “H” marker. In Figure 6 different line segment chains are detected and illustrated, where the chains of the “H” pattern, circular pattern and other unidentified objects are indicated in red, blue, and green color, respectively.

3.5 Target Refinement

Since line segments of the “H” pattern have been located, we suggest that it is sufficient to adopt the vertices of the pattern to fully represent the pattern itself. An ideal vertex is one of the “corners” of the pattern, which can also be treated as the intersection of two connected line segments. Due to the detection error in the EDLines algorithm’s output, however, we can’t directly use one of the intersecting points from the two line segments as the vertex, since they aren’t usually coincident (see Figure 7(a)). By re-calculating the intersection and eliminating the bias in the output of line segment detector, we obtain the new vertices’ positions. Not limited to this, we address the necessity to calibrate the camera barrel distortion, which effects the size of the target and leads to inaccurate altitude measurement. Instead of calibrating the whole image, which takes approximately 40ms to process one frame using our quadrotor’s on-board computer, only those vertices are corrected based on the camera matrix $R$ and distortion coefficient matrix $D$, resulting in processing time less than 0.1ms. The result is satisfactory whilst minimizing computation. Figure 7(b) shows the corrected vertices plotted on an undistorted image for validation purpose, where the obtained new vertices are precisely located at the desired positions. Thanks to this approach, the pixel error is minimized, which benefits our pose estimation process.

![Figure 6: Line segment chains after filtering. Red, blue and green color chains indicate the “H” pattern, the circular pattern and other objects.](image6.png)

![Figure 7: Vertices refinement: (a) Vertices incorrectly located. (b) Corrected vertices plotted on an undistorted image.](image7.png)

3.6 Information Gathering with Pattern Reconstruction

Once we have the calibrated vertices, our vision system loops through all of them to gather the key information of the “H” pattern. That is, with the re-calculated line parameters, we identify each line segment and categorize it into its own group by means of checking the line length proportion between the current one and the next one, according to Table 1. While implementing this, the concept, so called pattern reconstruction, is also proposed to recover the missing part of the pattern, as the missing lines can be artificially “rebuilt” using the gathered information of the existing lines. Figure 8(a) and 8(b) illustrate the “H” pattern before and after reconstruction, where all the calculated line segments match the actual ones quite well. To our best knowledge, this approach is the first work in this endeavor.

![Figure 8: “H” pattern before and after reconstruction.](image8.png)

3.7 Pose Estimation

XYZ Positions Estimation

Owing to an on-board pan-tilt unit (PTU) stabilizing the camera in flight, we have a downward-looking
monochrome camera whose image plane can be considered parallel to the ground. If the target size is known, we can work out the altitude, then the relationship between image pixels and horizontal distances in both $x$ and $y$ directions can be derived as well. A previous characterization has been done on altitude from vision. Our quadrotor was tied with a rope and moved from 1m to 2.6m at 0.1m intervals, which is the valid operating range for the monochrome vision approach. Within this range we obtain the optimal filtering and recognition results by tuning the parameters in (3)-(6), whilst record the corresponding size of the “H” pattern at each interval. Meanwhile, the horizontal distances from the pattern centre to camera centre and the corresponding pixels at the specific altitude are also logged. Using these data, the Matlab Curve Fitting Toolbox is applied to generate the empirical equations (7) and (8):

$$h = -\frac{284.3}{s_h - 1.142}\text{ or } -\frac{249.4}{s_w + 2.17}. \quad (7)$$

$$d = 0.307 + 0.135p_n + 0.067h_n + 0.029h_nh_n + 0.002h^2_n \quad (8)$$

where $s_w$ and $s_h$ are the width and height of the “H” pattern, $h$ and $h_n$ are the original and normalized altitude of the UAV, $d$ is the physical distance from the marker centre to the image centre in either $x$ or $y$ direction, and $p_n$ is the corresponding normalized pixel values of $d$.

**Pitch, Roll and Yaw Angles Estimation**

As mentioned above, using the circular pattern to estimate the pitch and roll angles based on ellipse fitting is problematic due to deformation caused by occlusion. Thanks to the detected vertices and the fact that they are co-planar, the Perspective-N-Points ($PnP$) method can be applied to work out the relative attitude based on 2D-3D point correspondences. Due to the experiments proposed by Sanchez-Lopez et al., however, such a method has a relative poor performance when the object plane is parallel to the image plane and is sensitive to image pixel error [Sanchez-Lopez et al., 2014]. Therefore, in their work they expect a large approaching angle between the camera and the deck to achieve an acceptable measurement. Unfortunately, measuring the relative attitude at a large aspect angle is less important than when hovering above the deck. Here, we propose a method which combines $PnP$ and a novel pattern evaluation scheme to achieve a more robust measurement for above-the-deck hover.

First, the yaw angle of the “H” pattern is defined as its major axis, which can be directly derived from slope of the “blue” colour line. We align the image $y$ axis to coincide with the pattern’s major axis by adjusting the heading of the UAV. A vertex map including the marker centre, vertices of which are labeled in clockwise-direction order, is shown in Figure 9. We verify each vertex’s unique position (number) by checking its relative position w.r.t to the other vertices and assign its pixel coordinate to this map. Then, the necessity of using $PnP$ is examined via the proposed pattern evaluation scheme. As the downward looking camera has a relatively small field of view, we would like to approximate the deformation of the “H” pattern as affine transformation. There are four parameters standing for two types of transformation in matrix $F$:

$$F = \begin{bmatrix} W & A & 0 \\ B & H & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (9)$$

where $W$ and $H$ are scaling, $A$ and $B$ are shearing in $x$ and $y$ directions. These parameters are examined based on the gathered information described above and thresholds $\Delta_1$ and $\Delta_2$ are applied to them. If thresholds are exceeded, there exists a relatively large pitch or roll angle, or a combination of both. Therefore, as we know the actual physical size of the pattern, the 2D-3D point correspondences can be used to estimate the pitch and roll angles as well as the value $\hat{h}$ in $z$ direction. Hence, $\hat{h}$ is substituted into (8) as a feedback scheme to obtain a more accurate estimation of $d$. Otherwise, we consider that the marker plane is close to parallel to the image plane. The 6DoF pose estimation is degraded to 4DoF, which includes estimating $x$, $y$, $z$ positions and yaw angle.

**Dual Vision to Increase Redundancy**

In the previous works we verified the timing performance of the vision system running on the quadrotor’s on-board computer with an image size of $640 \times 480$ pixels, achieving 20fps while having other essential software packages running. However, the monochrome camera has a limited operating range on account of its inherent nature. For a longer range, GPS can be used as the main device for guiding the UAV to the vicinity of the ship. When close to the shipdeck, the landing marker may becomes either too large or out of field of view that the vision system can easily lose track of it. Thus, deploying a
Time-of-Flight (ToF)-based PMD (Photonic Mixer Device) range camera as a complementary sensing method increases the redundancy of our vision system. Having another camera on-board requires additional computational power, so we decide to lower the framerate of the monochrome camera to 6fps, which is still feasible since we can fuse vision with the on-board IMU to achieve a much faster update rate.

The monochrome camera and the PMD range camera are mounted on the same plane, as seen in Figure 10(a). From approximately 1.0m to 2.0m, these two cameras share a common operating range that we can make use of the outputs from both to achieve a more reliable pose estimation. Therefore, the same chessboard corners in the captured images have been linked, whose pixel coordinates have been used for camera calibration (shown in Figure 10(b)), yielding the rotation vector \( \mathbf{r} \) (unit in degree) and translation vector \( \mathbf{t} \) (unit in millimeter) as:

\[
\begin{bmatrix}
0.0004 \\
0.0295 \\
0.0088
\end{bmatrix}
\begin{bmatrix}
11.18 \\
38.95 \\
25.47
\end{bmatrix}
\]  
(10)

in which we can assume that there is no rotation but only translation between the two cameras.

![Image of dual vision system](image)

**Figure 10:** Dual vision system: (a) A monochrome camera in association with the PMD range camera mounted on a plane. (b) Camera calibration using corresponding chessboard corners.

Note that the range camera has a much larger field of view than the monochrome camera, resulting in the landing marker occupying a very small area in the depth image. Accordingly, the region of interest (ROI) comprising the marker area must be specifically located so that the corresponding point cloud can be employed for plane fitting whilst minimizing computation. The polygon formed by linking vertices 1, 6, 7, and 12 in Figure 9 together with straight lines is selected as the ROI. The basis of finding the vertices in the PMD camera is point projection.

Suppose the centre of the “H” pattern has a coordinate of \( X_o = (x_o, y_o, z_o) \) in the monochrome camera’s coordinate frame, there exists a relationship of:

\[
n_x x_o + n_y y_o + n_z z_o = d
\]  
(11)

where \( N = [n_x, n_y, n_z]^T \) denotes the surface normal and \( d \) is the distance from the camera’s optical centre to the plane. Recall from the vision system output we have already obtained \( X_o \), whilst \( N \) can also be written as:

\[
N = [-\sin\beta, \sin\alpha\cos\beta, \cos\alpha\cos\beta]^T
\]  
(12)

where \( \alpha \) and \( \beta \) stand for the pitch and roll angles of the deck, respectively. Thus \( d \) is obtained from (11). For any other co-planar vertices \( X_n \) we also have:

\[
n_x x_n + n_y y_n + n_z z_n = d, \quad n = 1, 6, 7, 12.
\]  
(13)

and the relationship between \( X_n \) and \( X_o \) is partially known as:

\[
\begin{bmatrix}
x_o/cos\beta \\
y_o/sin\alpha \\
z_o
\end{bmatrix}
= \begin{bmatrix}
\cos\gamma & -\sin\gamma \\
\sin\gamma & \cos\gamma \\
0 & 0
\end{bmatrix}
\begin{bmatrix}
x
y
\end{bmatrix}
+ \begin{bmatrix}
x_n \\
y_n
\end{bmatrix}
\]  
(14)

where \( \gamma \) is the yaw angle, \( \bar{x} \) and \( \bar{y} \) are the relative physical distances between \( X_o \) and \( X_n \) in x and y directions, respectively. Hence, the unknown \( z_n \) is solved after substituting the values into (13). Now we have the coordinate of each \( X_n \) in the monochrome camera’s coordinate frame. Meanwhile we can simply obtain that coordinate \( \hat{X}_n \) in the PMD camera’s coordinate frame by:

\[
\hat{X}_n = X_n - t \quad n = 1, 6, 7, 12.
\]  
(15)

where \( t \) is the translation vector in (10). Finally the vertices in the PMD camera are calculated by 3D-2D back-projection:

\[
\begin{bmatrix}
\hat{x}_i \\
\hat{y}_i \\
1
\end{bmatrix}
= A
\begin{bmatrix}
\hat{x}_n/z_n \\
\hat{y}_n/z_n \\
1
\end{bmatrix}
\]  
(16)

in which \( (\hat{x}_i, \hat{y}_i) \) is the pixel coordinate of vertex \( i \) and \( A \) is the camera matrix of the PMD range camera. Figure 11 indicates the ROI in the PMD image after projection. Then, RANSAC based plane fitting method embedded in the Point Cloud Library (PCL) is introduced to process the point cloud for pose estimation after further rejecting the invalid outliers in the ROI.

**4 Experiment Setup**

The UAV involved in our research is the Pelican quadrotor from Ascending Technology GmbH. An Autopilot board acting as the flight control unit provides inner-loop stabilization and attitude command based control. It is equipped with a 3-axes accelerometers, rate gyroscopes, a pressure sensor, a magnetometer, a GPS
module and various communication interfaces. An on-board computer is comprised of an Intel Atom processor (1.6GHz) with 1GB of RAM, which has a pre-installed Ubuntu Linux operating system. Control commands are sent from the Atomboard to the Autopilot via serial interface. A light weight UI-1221-LE monochrome camera with a resolution of 752 × 480 pixels and a maximum framerate of 87.2Hz in conjunction with the PMD Camboard Nano range camera with a resolution of 165 × 120 and a maximum framerate of 90Hz are mounted on the PTU within the UAV frame. The optical lens attached to the monochrome camera is a 6mm S-mount IR cut filter lens, with 58° field of view (FOV). The operating range of the PMD camera is approximate from 10cm to 2m with an appropriate integration time setup. After camera calibration, we have \( f_m \cdot K_{xm} = 988.4475 \), \( f_m \cdot K_{ym} = 1002.34475 \), \( C_{xm} = 299.34709 \), and \( C_{ym} = 215.66086 \) for the monochrome camera matrix, and the distortion coefficient vector \( d_m \) is \([-0.57281, 0.25384, 0.01285, 0.00845, 0]\); \( f_r \cdot K_{xr} = 98.5608 \), \( f_r \cdot K_{yr} = 98.0349 \), \( C_{xr} = 77.4132 \), and \( C_{yr} = 57.6554 \) for the PMD camera matrix, the distortion coefficient vector \( d_r \) is \([-0.073, -0.5752, 0.01276, -0.00207, 0]\).

There has been a working inner-loop controller implemented on the Autopilot. In order to perform target tracking, we design three PID controllers to individually control the x, y and z axes positions, plus a PD controller for yaw control. The controller gains are roughly tuned according to experiments conducted using our VICON motion tracking system. An Extended Kalman Filter (EKF) with 12 degrees of freedom was designed to fuse the position data from the vision system with the IMU outputs to achieve a better estimation. For each axis, the position, velocity, acceleration and sensor bias are recorded.

5 Results and Discussion

In our previous research we have demonstrated the ability of the on-board systems to perform closed-loop target tracking and set-point hover in an imitated cluttered shipdeck environment with marker occlusion and illumination variation. For more detailed information, please refer to this video: [http://www.youtube.com/watch?v=x_QqVPBKads](http://www.youtube.com/watch?v=x_QqVPBKads). In this paper we try to evaluate the performance of our integrated system in tracking a moving deck and estimating its 6DoF motion. Again, for safety and feasibility, the Pelican quadrotor UAV is first guided by the VICON-simulated indoor GPS system to navigate to the vicinity of the shipdeck, then switched to vision in search of the landing marker. During target tracking, if the vision system loses track of the marker over a certain period, it will immediately switch back to GPS and fly to the nearest waypoint and re-initialize the target detection procedure. Both the quadrotor UAV and the shipdeck have VICON-markers attached to them, so that we are able to measure the exact 3D positions and poses of them. The landing marker was manually actuated to reach a maximum pitchroll angle of 15 degrees, or a combination of both. Figure 12(a) and (b) show some snapshots of the Pelican quadrotor hovering above the imitated moving shipdeck.

Since the vision system measures relative positions and the VICON system gives absolute positions, in our algorithm we rotate the measured 3D positions of the marker according to the heading of the UAV, add them to the marker’s absolute 3D positions to obtain the UAV’s absolute 3D positions, then compare them with VICON measurements. Due to the limitation of length, we only show one of the flight tests results.

The North-East-Down (NED) coordinate frame is involved in our work, thus the initial global coordinate of the marker centre is given as (-0.305m, -0.315, -0.06m).
We set the hover altitude at 1.8m above the target. The UAV was manually took off at approximately 4s, then guided by GPS at about 7s to fly to the waypoint. At 12s it reached the waypoint and the vision system located the landing marker 2s before this. After hovering for about 3s, the vehicle switched to vision-based control and descended to reach the desired hover altitude. From 22s to 38s, we manually actuated the shipdeck for 5 cycles to simulate pitching. At 47s, the vision system lost track of the marker for 1s, then the vehicle immediately switched back to GPS-based control to fly back to the waypoint. Started from 52s, we manually landed the UAV by remote control and terminated the experiment.

From Figure 13 we can observe that both raw vision and vision-INS EKF give reasonably precise results which coincide with VICON quite well. Especially when the marker was moving, the vision system correctly responded to the deck heave motion, resulting in variations in the UAV’s altitude during that period. The quadrotor oscillated around the setpoint, which is also the marker’s coordinate, in both x and y directions, whilst having an overshoot when descended from the GPS waypoint to the vision setpoint, reflecting the potential for better performance by improving the on-board controller design. Moreover, the loss of visual track at 47s was due to the late response of the controller, leading to more than half of the marker being lost from the field of view. Therefore, the vision system outputted constant values for 1s, followed by the vehicle switching back to GPS-based control mode.

![Figure 13: Vision-based closed-loop test: 3D positions estimation.](image)

In Figure 14, although the pitch and roll measurements by raw vision are noisy, they follow the trend of the deck motion. Specifically, when the pitch and roll angles are small, the estimation becomes worse (see 25s, 28s, 32s, 34s and after 37s in both measurements). That’s due to the inherent drawback of PoP, which is when the target plane is close to parallel to the image plane, it becomes unstable. Moreover, the measurement is affected by the on-board PTU, as it introduces an error while stabilizing the camera that the image plane is not always parallel to the ground. The performance of the pattern evaluation scheme is not as good as that at a lower altitude, thus fails to shave the peaks that shouldn’t be there in Figure 14. That’s because when the marker becomes smaller, the deformation is less significant for the vision system to detect. But we affirm that at a lower altitude (usually less than 1.5m), the variation is obvious enough for our vision system to detect, resulting in a much better performance. Meanwhile, the PMD range camera gives reasonable results in sensing the pitch angle, but has noisy measurements in roll angle. One reason is that due to high altitude, the point cloud has less points than when it is close and the measurement is noiser. Thus, the PMD range camera is more preferable operating at a closer range. Finally, the yaw measurement is much better than the two, since the line segment features are more robust and intuitive. Note that there is a constant bias of 2.4° between the vision and VICON measurements, which is due to the yaw angle initialization error in creating the object for the quadrotor in the VICON coordinate. This bias has already been subtracted in Figure 14. We also calculate the Root Mean Square Error (RMSE) in Figure 13, which are 4.06°, 3.61° and 3.52° for raw vision and 3.96°, 3.61° and 3.22° for vision-INS EKF in x, y and z directions, respectively, which reflects a certain degree of improvement in using sensor fusion. For Figure 14, the RMSE for pitch, roll measurements are 5.26° and 2.62° for raw vision, and 3.33° and 6.78° for PMD range camera, respectively. The RMSE for yaw angle is 0.45°. According to the analysis, we suggest that the UAV hovers at a lower altitude to achieve a better estimation, whilst adopting sensor fusion will also improve the result. Videos of our flight tests can be found at: https://www.youtube.com/watch?v=cqLinA4C-wM and https://www.youtube.com/watch?v=D6TS0HgfJNM.

6 Conclusion

We successfully extend our previous work in [Lin et al., 2015] to estimate full 6DoF pose of a shipdeck with motions. Real-time on-board flight tests have demonstrated the feasibility of the integrated system. In our future research we aim to further improve the precision of the system, especially in estimating pitching and rolling of the shipdeck, which is of great importance to predict periods of quiescent ship motion so that landings can be executed only at times when deck motion is within safe limits. A sensor fusion scheme is also required to perform
information integration at different landing stages, which can provide more robust measurements of the shipdeck motion.

References


