

# A 6D-pose estimation method for UAV using known lines

Wenxin Liu<sup>1</sup>, Shuo Yang<sup>2</sup>, Ming Liu<sup>1</sup>

**Abstract**—This paper introduces two efficient global localization and attitude estimation (6D global pose estimation) algorithms for an unmanned aerial vehicle (UAV) over a known rectangular field by detected lines using monocular cameras. The algorithms are designed in the context of International Aerial Robotics Competition (IARC) Mission-7, which has been considered as one of the most prestigious up-to-date UAV challenges. The first algorithm achieves 6D global pose estimation by detection of three different lines, and the second algorithm is by observing a known corner of the playground. The simulation and experimental results validate the accuracy and robustness of the proposed algorithms.

## I. INTRODUCTION

### A. Motivation and IARC

Localization is the basis for all position-based robotic activities. Specifically, when a UAV is concerned, the 6D pose estimation of the platform is usually required. Over the past decade, pose estimation for UAVs has attracted great attention [1], [2]. Several works took sensor fusion into account [3], [4], [2]. In most cases, sensor fusion adopts a loosely coupled approach, in which the precision of frame-to-frame pose estimation or frame-to-model is crucial. However, because of the lack of global consistent reference, the estimation error will usually accumulate inevitably over time.

IARC is one of the most challenging competitions for UAV nowadays, and it is the longest running collegiate aerial robotics challenge in the world. It aims at tackling challenges that currently no flying robots owned by a government or an industry can achieve. For the history of the competition, please refer to <http://www.aerialroboticscompetition.org>. The current challenge (Mission-7) includes a multi-robot scenario in which a UAV needs to navigate over a predefined playground and precisely control several ground robots to move around. As required by the competition, only “mapless” approaches are allowed for Mission-7. Therefore, classical methods such as PTAM[5]-based methods are not feasible. In addition, the UAV needs to move always at a high speed over the playground. The playground was a “featureless surface” as

required by the first released version of game rule. Consider that the competition is in indoor environment with limited luminance conditions, the motion-blur for down-looking cameras and the minimal features leave minor possibility for stable tracking of the features on the ground.<sup>1</sup> Therefore, optical flow-based pose estimation is not feasible either. Without global localization, the error accumulated in these incremental approaches can lead to a mission failure. In this paper, we aim at using the limited information about the playground to globally estimate the 6D pose of a UAV directly from image observation. Our platform is shown in Figure 1, which won two awards for the first-year competition in August, 2014.



Fig. 1. Our platform in the middle of competition. Please note that the playground was featured, which was contradictory to our assumption that only border-lines can be observed, due to last-time changes of rules.

### B. Visual Odometry and Navigation

Visual odometry is the key method to estimate the pose of a UAV [6], [7]. There are two main categories of methods, namely absolute pose estimation [8] and relative pose estimation, e.g. [9]. In typical robotic scenarios, relative pose estimation has been widely used due to the relatively higher precision. However, errors can be incrementally accumulated over time. For a long-run experiment, the failure of pose estimation can easily lead to mission failure.

Another related taxonomy is visual servoing (or called visual homing for mobile robots [10], [11]). Using a given set of reference features, the robot can converge to predefined reference positions. The classical visual servoing methods are based on point features [12]. Some variations such as those that are based on lines [13] have been reported. However,

<sup>1</sup>Wenxin Liu and Ming Liu are with the Department of Electronic & Computer Engineering, Hong Kong University of Science & Technology {wliuad, eelium}@ust.hk

<sup>2</sup>Shuo Yang is from DJI Innovations and the Department of Electronic & Computer Engineering, Hong Kong University of Science & Technology syangag@ust.hk

This work was partially supported by the HKUST Project IG13EG03; partially sponsored by the Research Grant Council of Hong Kong SAR Government, China, under project No. 16206014 and National Natural Science Foundation of China No. 6140021318. All grants were awarded to Prof. Ming Liu.

<sup>1</sup>Please note that this was the original definition of the rules. Due to requests, the playground was designed as featured surface since April 2014. This paper deals with the localization with only boundary lines, i.e. before the rule changes.

recovery of the metric information is impossible using such approaches.

### C. Localization with Known Visual Hints

Several related works used points as primary features in localization, e.g. Perspective-Three-Point (P3P). They aim at determining the position and orientation of a camera in a reference frame from point correspondences [8], [14]. Specifically for IARC Mission-7, these methods are not feasible, because in most cases the point features are not detectable as the UAV must maneuver at different heights over a feature-less ground. Even if some point features can be tracked locally and temporarily, global localization is still necessary at the mission planning level. These observations do infer that localization by point features are not applicable for the mission. On the other hand, inspired by some early works in RoboCup competitions [15], [16], since the line markers on the playground are specifically highlighted (see: section II), line-based methods can be designed accordingly. Please note that the points at the crossing of lines cannot be observed due to the large area of playground, therefore existing jointly point-line-based methods are not applicable such as [17], [18]. This work is stimulated by the fact that lines are as cross product results from planes instead of as extensions of connected points. This assumption leads to interesting results that a set of parallel lines can be easily represented using the known playground boundaries. This set of parallel lines assist to finally refine the pose of the UAV.

## II. PRELIMINARIES

IARC Mission-7 requires autonomous flying of a UAV above a 20x20 meter square area. Four lines form the area on the ground. A pair of the opposite lines are white colored (depicted in black in Figure 2; the other two lines are colored in red and green separately. We denote the frame of the world  $\omega$ , which is originated at the cross point of the green line and one of the white line (the one specifically defined in Figure 2). The frame of the UAV is denoted by  $\pi$ .

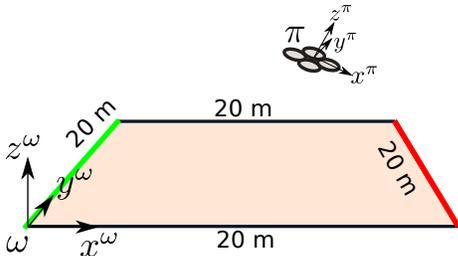


Fig. 2. Playground for the IARC Mission 7

The position of the UAV is  $\mathbf{x}^\omega := \{Px^\omega, Py^\omega, Pz^\omega\}$ . The attitude of the UAV is  $\mathbf{q}$ . Although both  $\mathbf{x}$  and  $\mathbf{q}$  are time-variant, the time component is omitted in the representation here for simplicity.

Consider that the four lines provide  $N$  detectable points, each point  $i^\omega$  in the world frame can be projected to the local

frame of the UAV through fully calibrated camera(s). Let the frame of a camera be  $c$ . The point  $i$  can be represented in  $\pi$  frame (UAV) by:

$$\vec{p}_i^\pi = \mathcal{T}_c^\pi \mathcal{T}_\omega^c i^\omega$$

where  $\mathcal{T}_c^\pi$  and  $\mathcal{T}_\omega^c$  are the transformation from camera to the UAV and world to camera respectively. For fully calibrated cameras, the transformation from world to camera frame is:

$$\mathcal{T}_\omega^c = \begin{pmatrix} \mathcal{R}_\omega^c & T \\ \mathbf{0} & 1 \end{pmatrix} K$$

where  $K$  is the intrinsic camera matrix;  $\mathcal{R}_\omega^c$  and  $T$  define the extrinsic parameters.  $\mathcal{T}_c^\pi$  is the result of extrinsic calibration. A simple sketch is shown in Figure 3.

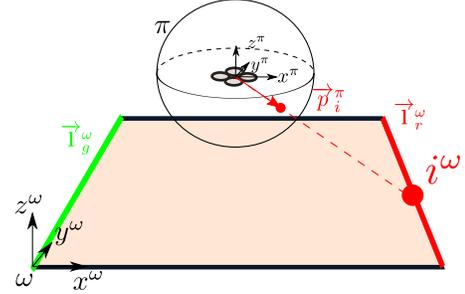


Fig. 3. Projected point vector in local  $\pi$  frame.

The two specially colored lines can be seen as composition of points in the world frame. Here, the green line is denoted by  $\vec{I}_g^\omega := \{i^\omega | i^\omega \in \text{points of green color}\}$ ; the red line is  $\vec{I}_r^\omega := \{i^\omega | i^\omega \in \text{points of red color}\}$ . Note that the observation vectors  $\{\vec{p}_i^\pi\}$  are generally normalized directions in the UAV frame.

## III. POSE ESTIMATION BY OBSERVING THREE LINES

A two-phase algorithm is used. Firstly, we can find the rotation between world frame and body(UAV) frame by observing two parallel lines and gravity measurement. Secondly, using the rotation matrix, we can transform important vectors observed in UAV frame into world frame, then with simple geometry we can derive the exact position of UAV in the arena. We abuse  $\vec{I}_g^\omega$  and  $\vec{I}_r^\omega$  as the two parallel lines in this section. The other two white can also be used in the same analysis.

### A. Finding the rotation

1) *Intersection calculation in  $\pi$* : The following two observations can be derived:

*Observation 1*: As shown in Figure 4, considering all planes pass through  $\vec{I}_g^\omega$  and  $\vec{I}_r^\omega$ , the intersection is always a line that is parallel to both  $\vec{I}_g^\omega$  and  $\vec{I}_r^\omega$ , because  $\vec{I}_g^\omega$  and  $\vec{I}_r^\omega$  are parallel.

*Observation 2*: The plane formed by any two different vectors with both their heads are points from either  $\vec{I}_g^\omega$  or  $\vec{I}_r^\omega$  lies on one of the mentioned planes in *Observation 1*.

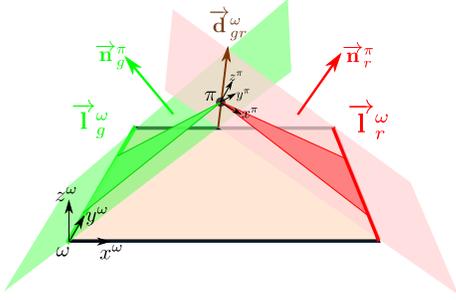


Fig. 4. Computation frame for Section III.A

By these two observations, we can calculate the intersection line  $\vec{d}_{gr}^\omega$  by:

$$\vec{d}_{gr}^\pi = \frac{\vec{n}_r^\pi \times \vec{n}_g^\pi}{\|\vec{n}_r^\pi \times \vec{n}_g^\pi\|} \quad (1)$$

where  $\vec{n}_g^\pi$  is the normal direction of the plane passes  $\vec{I}_g^\omega$ , while  $\vec{n}_r^\pi$  is the normal direction of the plane passes  $\vec{I}_r^\omega$ . They can be easily derived from:

$$\begin{cases} \vec{n}_g^\pi = \frac{\vec{p}_{g1}^\pi \times \vec{p}_{g2}^\pi}{\|\vec{p}_{g1}^\pi \times \vec{p}_{g2}^\pi\|} \\ \vec{n}_r^\pi = \frac{\vec{p}_{r1}^\pi \times \vec{p}_{r2}^\pi}{\|\vec{p}_{r1}^\pi \times \vec{p}_{r2}^\pi\|} \end{cases} \quad (2)$$

where  $\{\vec{p}_{g1}^\pi$  and  $\vec{p}_{g2}^\pi\}$  are two observation vectors sampled from  $\vec{I}_g^\omega$ , and  $\{\vec{p}_{r1}^\pi$  and  $\vec{p}_{r2}^\pi\}$  are two observation vectors sampled from  $\vec{I}_r^\omega$ . Take care of the sequence in computing the cross product to make sure that the normal directions are upwards.

As discussed in *Observation 1*,  $\vec{d}_{gr}^\pi$  is parallel with both  $\vec{I}_g^\omega$  and  $\vec{I}_r^\omega$ , so we can find the rotation from  $\pi$  frame to a new frame having its y axis aligned with  $\vec{d}_{gr}^\pi$ . We define the new frame to be  $\zeta$ , so the  $x^\zeta$ - $z^\zeta$  plane is perpendicular to  $\vec{I}_g^\omega$  and  $\vec{I}_r^\omega$ .

A subtle problem is that the pitch angle is not directly observable from the two lines. In this case, a gravity direction can be adopted (if possible) to compensate for the error in  $z^\pi$  direction. Here we assume the pitch angle is observable by gravity direction or the UAV can stabilize itself. Next, we can find the rotation between the local frame and the global frame. This assumption is proved feasible for the platform that we used for the challenge.

2) *Rotation*: Firstly, we need to find the rotation between  $\pi$  and  $\zeta$ . Figure Figure 5 is a sketched illustration.

In order to align the two axes, the two highlighted angles are required, i.e.

$$\begin{cases} \theta = \arctan \frac{\vec{d}_{gr}^\pi|_{x^\pi}}{\vec{d}_{gr}^\pi|_{y^\pi}} \\ \psi = \arctan \frac{\vec{d}_{gr}^\pi|_{z^\pi}}{\sqrt{\vec{d}_{gr}^\pi|_{x^\pi}^2 + \vec{d}_{gr}^\pi|_{y^\pi}^2}} \end{cases} \quad (3)$$

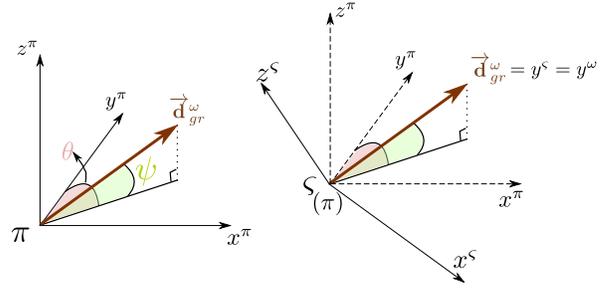


Fig. 5. Alignment of  $y^\pi$  and  $y^\omega$ , so that a new frame  $\zeta$  is generated.

where  $\vec{d}_{gr}^\pi|_{x^\pi}$ ,  $\vec{d}_{gr}^\pi|_{y^\pi}$ , and  $\vec{d}_{gr}^\pi|_{z^\pi}$  indicate the components along  $x^\pi$ ,  $y^\pi$  and  $z^\pi$  axes respectively.

The rotation matrix from frame  $\pi$  to  $\zeta$  is calculated by a rotation about  $z^\pi$  for  $-\theta$ , followed by a rotation about  $x^\pi$  for  $\psi$ , so

$$\mathcal{R}_\pi^\zeta = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & \sin \psi \\ 0 & -\sin \psi & \cos \psi \end{pmatrix} \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4)$$

After that, the frame  $\zeta$  can be easily again rotated to align with  $\omega$ , since now  $y^\zeta$  is aligned with  $y^\omega$  already. Here we need the gravity measurement  $\vec{g}^\pi$  from the local frame of the UAV. The missing alignment to  $z^\omega$  axis is achieved by a pitch angle  $\tau$ .

$$\tau = \arccos \left( \vec{g}^{\zeta T} (-\vec{z}^\zeta) \right) = \arccos \left( (\mathcal{R}_\pi^\zeta \vec{g}^\pi)^T \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right) \quad (5)$$

where  $T$  denotes the transpose of vector. Therefore, the attitude transformation from  $\pi$  to  $\omega$  is shown as Equation 5.

## B. Localization

1) *Solution to a line*: Taking the sketch shown in Figure 6 as an instance, the height  $z^\omega$  and offset in between  $\vec{I}_g^\omega$  and  $\vec{I}_r^\omega$  i.e.  $x^\omega$  can be calculated. The transformed plane normals

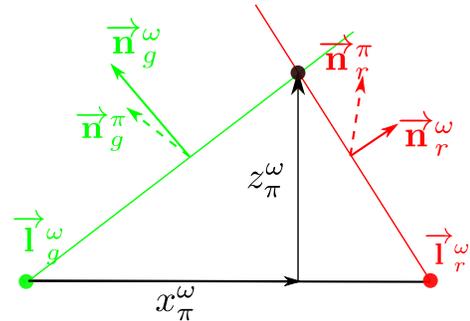


Fig. 6. Solution to offset and height

are:

$$\begin{cases} \vec{n}_g^\omega = \mathcal{R}_\pi^\omega \vec{n}_g^\pi \\ \vec{n}_r^\omega = \mathcal{R}_\pi^\omega \vec{n}_r^\pi \end{cases} \quad (7)$$

$$\mathcal{R}_\pi^\omega = \mathcal{R}_\zeta^\omega \mathcal{R}_\pi^\zeta = \begin{pmatrix} \cos \tau & 0 & -\sin \tau \\ 0 & 1 & 0 \\ \sin \tau & 0 & \cos \tau \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & \sin \psi \\ 0 & -\sin \psi & \cos \psi \end{pmatrix} \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (6)$$

By simple triangular geometry, the following relations can be easily derived:

$$\begin{cases} \frac{Pz_\pi^\omega}{Px_\pi^\omega} = \frac{-\vec{\mathbf{n}}_g^\omega|_{x^\omega}}{\vec{\mathbf{n}}_g^\omega|_{z^\omega}} \\ \frac{Pz_\pi^\omega}{20 - Px_\pi^\omega} = \frac{-\vec{\mathbf{n}}_r^\omega|_{x^\omega}}{\vec{\mathbf{n}}_r^\omega|_{z^\omega}} \end{cases} \quad (8)$$

where 20 is the width of the playground (in meters);  $\vec{\mathbf{n}}_g^\omega|_{x^\omega}$ ,  $\vec{\mathbf{n}}_g^\omega|_{z^\omega}$ ,  $\vec{\mathbf{n}}_r^\omega|_{x^\omega}$  and  $\vec{\mathbf{n}}_r^\omega|_{z^\omega}$  denote the component in  $x^\omega$  and  $z^\omega$  of  $\vec{\mathbf{n}}_g^\omega$  and  $\vec{\mathbf{n}}_r^\omega$  respectively.

The analytical solution is:

$$\begin{cases} Px_\pi^\omega = \frac{-20}{\left(\frac{\vec{\mathbf{n}}_g^\omega|_{z^\omega}}{\vec{\mathbf{n}}_g^\omega|_{x^\omega}} + \frac{\vec{\mathbf{n}}_r^\omega|_{z^\omega}}{\vec{\mathbf{n}}_r^\omega|_{x^\omega}}\right)} \\ Pz_\pi^\omega = Px_\pi^\omega \left(\frac{\vec{\mathbf{n}}_g^\omega|_{z^\omega}}{\vec{\mathbf{n}}_g^\omega|_{x^\omega}}\right) \end{cases} \quad (9)$$

Please note that  $\vec{\mathbf{n}}_g^\omega|_{y^\omega}$  should be close to zero based on *Observation 1*.

2) *Complete 3D solution*: In order solve the third dimension  $Py_\pi^\omega$ , a third condition, such as observations from the third line is required. As shown in Figure 7, a third point is

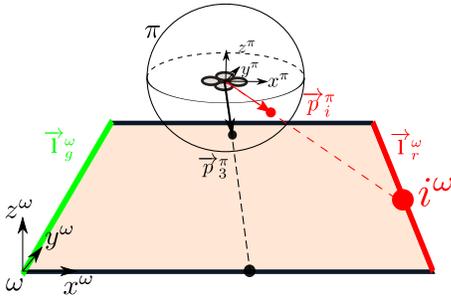


Fig. 7. A third observation vector

observed by the UAV as vector  $\vec{p}_3^\omega$ . The vector in  $\omega$  frame is

$$\vec{p}_3^\omega = \mathcal{R}_\pi^\omega \vec{p}_3^\pi \quad (10)$$

It can be derived by triangulation that the third dimension  $Py_\pi^\omega$  is:

$$Py_\pi^\omega = Pz_\pi^\omega \cdot \frac{\vec{p}_3^\omega|_{y^\omega}}{\vec{p}_3^\omega|_{z^\omega}} \quad (11)$$

where  $\vec{p}_3^\omega|_{y^\omega}$  and  $\vec{p}_3^\omega|_{z^\omega}$  represent the component of  $\vec{p}_3^\omega$  in  $y^\omega$  and  $z^\omega$  respectively.

So far, we have the full solution of the UAV position in 3D with respect to the playground, as well as the attitude given by  $(\mathcal{R}_\pi^\omega)^T$ .

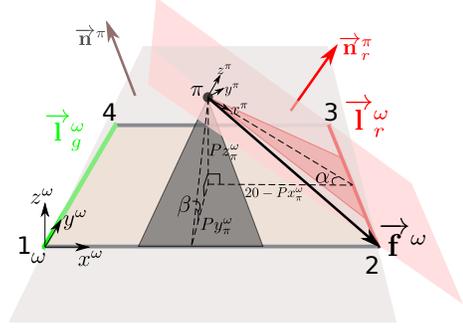


Fig. 8. Localization by observing a corner

#### IV. POSE ESTIMATION BY OBSERVING A CORNER

If a height measure is available, by observing a corner, the 3D position localization is also easily achievable. Please note that, it is not necessary to literally observe a corner. Partial observations of two adjacent lines would be sufficient, since the corner coordinate in the UAV frame can be obtained by cross product of the two line equations. In Figure 8, the four corners are labeled by 1 to 4. Taking corner 2 as an example, the 3D position is calculated as follows.

##### A. Position solution

Similarly, we can first localize the UAV to a line. The following relations can be achieved by triangulation:

$$\begin{cases} \alpha = \arccos(\vec{\mathbf{n}}_r^\pi)^T(-\mathbf{g}^\pi) \\ \cot \alpha = \frac{\vec{\mathbf{n}}_r^\omega|_{z^\omega}}{\vec{\mathbf{n}}_r^\omega|_{x^\omega}} = \frac{20 - P\hat{x}_\pi^\omega}{P\hat{z}_\pi^\omega} \\ \beta = \arccos(\vec{\mathbf{n}}_g^\pi)^T(-\mathbf{g}^\pi) \\ \cot \beta = \frac{\vec{\mathbf{n}}_g^\omega|_{z^\omega}}{\vec{\mathbf{n}}_g^\omega|_{x^\omega}} = \frac{P\hat{y}_\pi^\omega}{P\hat{z}_\pi^\omega} \end{cases} \quad (12)$$

where  $\vec{\mathbf{n}}^\omega$  and  $\vec{\mathbf{n}}_r^\omega$  are calculated similarly to Equation 2. 20 is the width of the playground in meter;  $\vec{\mathbf{n}}_r^\omega|_{x^\omega}$ ,  $\vec{\mathbf{n}}_r^\omega|_{z^\omega}$  denote the component in  $x^\omega$  and  $z^\omega$  of  $\vec{\mathbf{n}}_r^\omega$  and  $\vec{\mathbf{n}}^\omega$  respectively. The  $y^\omega$  component of  $\vec{\mathbf{n}}_r^\omega$  and  $x^\omega$  component of  $\vec{\mathbf{n}}^\omega$  are arbitrary and won't affect the calculation in Equation 10. Let the normalized components of vector  $[P\hat{x}_\pi^\omega, P\hat{y}_\pi^\omega, P\hat{z}_\pi^\omega]^T$  be  $[Px_0^\omega, Py_0^\omega, Pz_0^\omega]^T$ . The line equation passes through the origin of  $\pi$  frame and corner 2 is:

$$\lambda = \frac{x^\omega - Px_0^\omega}{20 - Px_0^\omega} = \frac{y^\omega - Py_0^\omega}{0 - Py_0^\omega} = \frac{z^\omega - Pz_0^\omega}{h^\omega - Pz_0^\omega} \quad (13)$$

where  $h^\omega$  is the measured height of the UAV in the world frame, by which the position of the UAV is determined. Similarly the lines passes through corner 1,3,4 can be derived in a similar way, which is omitted here.

Note that if the height measure is not available, the UAV position can be only determined to the level that it lies on the line defined by Equation 11. However, the correspondence observation vectors ( $\vec{f}^\omega$  and  $\vec{f}^\pi$ ) are known by this step.

### B. Attitude estimation

The attitude of the UAV can be estimated in the following way, as sketched in Figure 9. We have two pairs of known

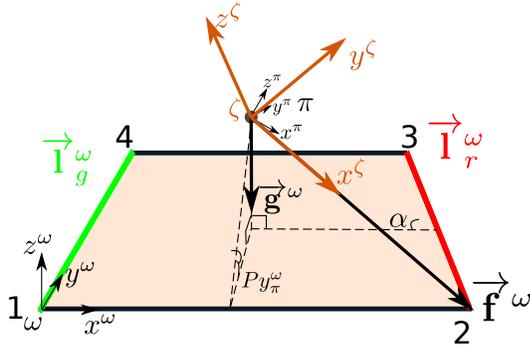


Fig. 9. Attitude recovery when observing a corner

vectors that are corresponding each other in  $\pi$  and  $\omega$ , i.e. the vector ( $\vec{f}^\omega$  and  $\vec{f}^\pi$ ) and ( $\vec{g}^\omega$  and  $\vec{g}^\pi$ ). Consider an intermediate frame  $\zeta$  as shown in Figure 9, the three basis vectors for the rotation  $\mathcal{R}_\zeta^\omega$  are defined as:

$$\begin{cases} \vec{x}_\zeta^\omega = \frac{\vec{f}^\omega}{\|\vec{f}^\omega\|} \\ \vec{y}_\zeta^\omega = \frac{\vec{f}^\omega \times \vec{g}^\omega}{\|\vec{f}^\omega \times \vec{g}^\omega\|} \\ \vec{z}_\zeta^\omega = \frac{\vec{x}_\zeta^\omega \times \vec{y}_\zeta^\omega}{\|\vec{x}_\zeta^\omega \times \vec{y}_\zeta^\omega\|} \end{cases} \quad (14)$$

The rotation  $\mathcal{R}_\zeta^\omega = [\vec{x}_\zeta^\omega \ \vec{y}_\zeta^\omega \ \vec{z}_\zeta^\omega]$

Similarly, for rotation from  $\zeta$  to  $\pi$  is spanned by:

$$\begin{cases} \vec{x}_\zeta^\pi = \frac{\vec{f}^\pi}{\|\vec{f}^\pi\|} \\ \vec{y}_\zeta^\pi = \frac{\vec{f}^\pi \times \vec{g}^\pi}{\|\vec{f}^\pi \times \vec{g}^\pi\|} \\ \vec{z}_\zeta^\pi = \frac{\vec{x}_\zeta^\pi \times \vec{y}_\zeta^\pi}{\|\vec{x}_\zeta^\pi \times \vec{y}_\zeta^\pi\|} \end{cases} \quad (15)$$

The rotation  $\mathcal{R}_\zeta^\pi = [\vec{x}_\zeta^\pi \ \vec{y}_\zeta^\pi \ \vec{z}_\zeta^\pi]$

As a result, the rotation transformation from frame  $\omega$  to  $\pi$  is:

$$\mathcal{R}_\omega^\pi = \mathcal{R}_\zeta^\pi \mathcal{R}_\zeta^\omega = \mathcal{R}_\zeta^\pi (\mathcal{R}_\zeta^\omega)^T = [\vec{x}_\zeta^\pi \ \vec{y}_\zeta^\pi \ \vec{z}_\zeta^\pi] [\vec{x}_\zeta^\omega \ \vec{y}_\zeta^\omega \ \vec{z}_\zeta^\omega]^T \quad (16)$$

The attitude of the UAV is obtained by  $\mathcal{R}_\omega^\pi = (\mathcal{R}_\omega^\pi)^T$ .

Note here we assume the visual observation is a more reliable measure than gravity estimation. Therefore, the visual feature vector is taken as the reference axis in  $\zeta$  frame. Another interesting observation is that the attitude is able to be recovered by only observing a corner, deriving from Equation 13 and Equation 10.

## V. VALIDATION AND DISCUSSION

So far, we have introduced two global pose estimation methods using the known boundary lines. To further validate the methods, we adopt *gazebo*<sup>2</sup> to simulate the behaviors of the UAV. Then we introduce the experiment on real system.

### A. Simulation Setup

Five cameras are mounted on the UAV at four equally distributed directions as well as the down-looking direction. We assume camera models are both intrinsically and extrinsically calibrated, such that the observed points from each camera can be projected as rays pointing to the center of the UAV. In practical cases, the intrinsic parameters are obtained by standard calibration procedure using a check-board and the extrinsic parameters are obtained by a Motion Capture system with sub-millimeter precision.

Due to limited space, we will not discuss the influence of line detection accuracy. The accuracy of line detection and its effect on the pose estimation will be discussed with real experiments in our further reports. However, we show the general effect of primary noise factors such as the gravity direction estimation and height estimation.

The validation adopts two major criteria: Euclidean error for position estimation and a Riemannian measure on  $SO(3)$  group for attitude. For the Riemannian metrics and the computation of geodesics, please further refer to [19]. Here we adopt the most widely used Riemannian metric to calculate the geodesic of two  $SO(3)$  rotation matrices  $R_1$  and  $R_2$  as,

$$d_R(R_1, R_2) = \frac{1}{\sqrt{2}} \|\log(R_1^{-1} R_2)\|_F \quad (17)$$

where  $F$  denotes the Frobenius norm. We use  $d_R(\hat{R}, R^*)$  to represent the difference between an estimated attitude  $\hat{R}$  and the ground-truth  $R^*$ .

For all the following validations, the simulated UAV flies following the same trajectory (sequence of poses). We analyze the accuracy and behavior both the algorithms introduced in section III and section IV, namely “three-line algorithm” and “corner-based algorithm” respectively. We assume the norm of the projected the estimated gravity direction (a unit vector) onto horizontal plane is with Gaussian noise  $\mathcal{N}(0, 0.01)$ . The height estimation used by the corner-based algorithm is with centimeter accuracy. Both parameters are selected according to the characteristics of accessible hardware setup.

### B. Simulation for three-line algorithm (section III)

Using the setup of subsection A, with additional noise in gravity estimation, Figure 10 depicts a simulated trajectory

<sup>2</sup><http://gazebo.org>

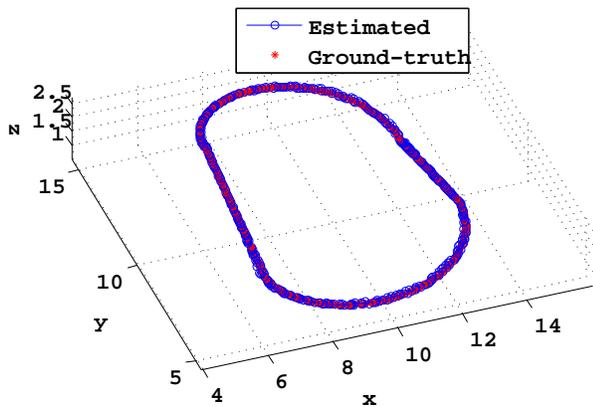


Fig. 10. Pose estimation by three lines with inaccurate gravity estimation. Units are in meter.

of the UAV using three-line algorithm, where blue circles indicate the estimated position against the red ground-truth. The estimated trajectory generally overlays on the ground-truth. In terms of position accuracy, Figure 11 shows the errors in x, y and z directions respectively along time axis. The position errors are generally small (within 0.5 meter). Note that the error in z direction (denoted by red curve) is significantly smaller than the horizontal directions. A possible reason is that the noise in gravity estimation is magnified in horizontal directions due to triangle relations. Conversely, for the estimation in versicle direction, the noise is balanced by the two estimated normals. At the same time, we see that the estimation error in x and y directions vary significantly over time. By studying the retraced data, the high errors happen when the UAV is close to either border of the playground. The observation noise increases in such cases, leading to sub-optimal results. The cyclic movement of the UAV causes the tide-like errors in amplitudes. In terms of orientation accuracy, Figure 12 shows the Euler angular errors along time. In general, the angular estimation is accurate. The maximum estimation error is within 0.4 degree. It shows the yaw estimation is much better than the other two. The reason is similar to the position estimation. Since the reference lines are taken from the horizontal playground, yaw estimation benefits the inherent consistency in feature tracking. Conversely, the estimations of roll and pitch rely much on the accuracy of gravity direction estimation.

In order to better understand how the accuracy in gravity estimation can affect the pose estimation, we plot both the position errors and attitude errors in Figure 13 and Figure 14 respectively. Third-ordered linear regression with linear kernel is also implemented to extract the tendency of the data. It can be seen in Figure 13 that on the one hand the position error generally increases with a worse estimation of gravity direction; on the other hand, the majority estimation results are with low errors (clustered at the bottom of the plot), which indicates the robustness of the proposed algorithm. However, the general estimation of attitude will not be much affected by increased error in gravity estimation. Please note that this result is not contradict with that in

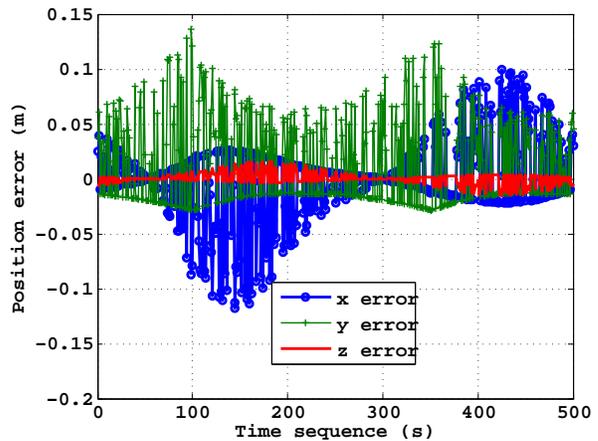


Fig. 11. Position error along simulation time

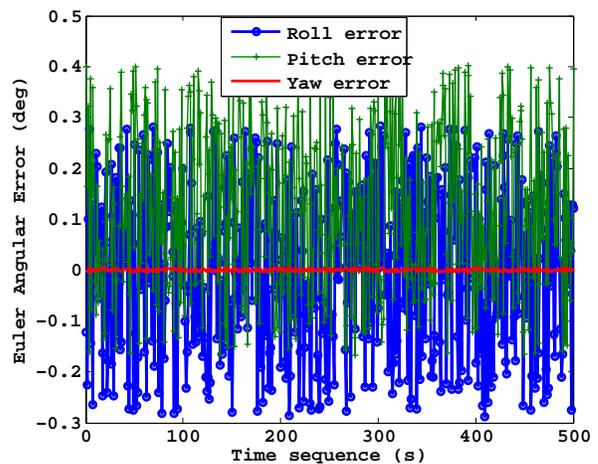


Fig. 12. Angular error along simulation time

Figure 12, because Figure 12 only indicates yaw estimation is superior than roll and pitch estimation, and Figure 14 implies better gravity estimation (within the test range) does not imply significantly better overall attitude estimation. Please be reminded that the behavior is different for corner-based algorithm shown in ??.

Due to page limits, the simulation for corner-based algorithm by section IV is omitted in this paper. The readers are referred to our full-length tech report at [20].

### C. Summary

In general, we compared the two proposed algorithms to estimate the pose of a UAV using the playground for IARC Mission-7. We see that the three-line algorithm is more resistant to observation noise, especially it only requires one additional measure other than vision. Since the absolute distance between two boarder-lines is used, this additional information also helps to form a more precise solution.

## VI. EXPERIMENT ON TEST-FIELD

We set up a smaller square field (5m x 5m) to mimic the IARC competition arena. When the quadrotor is moving in

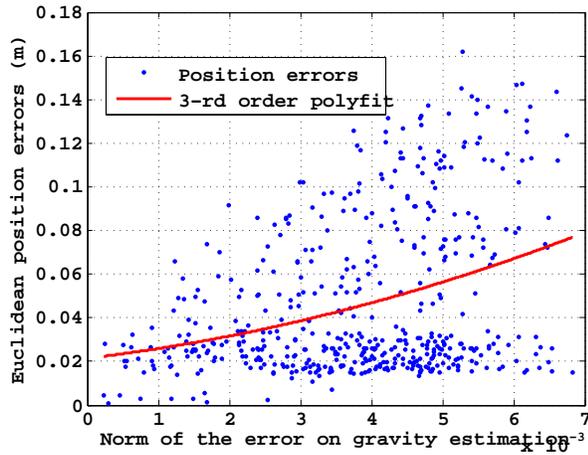


Fig. 13. Position error over the error in gravity direction estimation

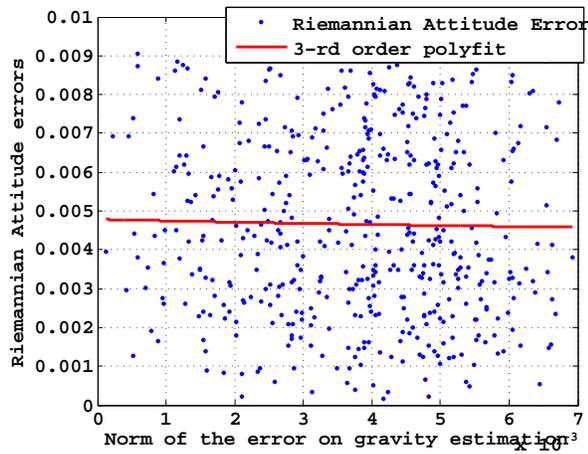


Fig. 14. Attitude error against gravity direction estimation

the field, its four cameras send images to the ground station, where its absolute position is calculated. Figure 15 shows the experimental field and screenshots from the control station. Four cameras used by the algorithm are mounted on the four sides of the UAV, looking at four equally distributed directions as shown in Figure 16. Please note that we do not aim at on-board global localization solution for IARC, since the heavily-loaded data from four HD cameras need to be processed in real-time. However, the flight control is taken care of on-board. For interested readers, please refer to [21] for further detail.

Two tricks are further introduced to ensure the implementation stability. They are briefly described as follows:

- **Rejection-based boundary line detection:** The boundary line detection is based on Hough-transform. Usually, plenty of outliers can be detected. We introduced a selection process based on the gradients (to reject lines with large slopes), lengths (to reject short segments) and locations (to reject lines above the ground). The remaining group votes for the optimal estimation based on the line saliency.

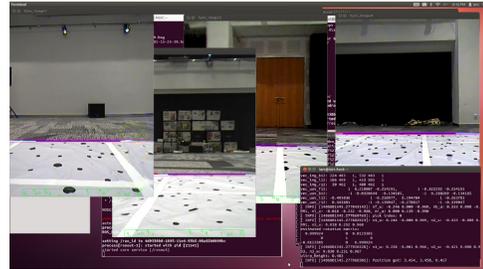
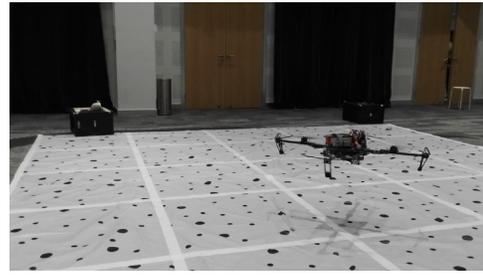


Fig. 15. The upper image shows the course of the localization experiment. The lower image is a screenshot of the host computer, which displays images wirelessly transmitted from four cameras with high-definition videos at 30Hz. The screenshot shows an instance where the field boundaries are correctly detected and highlighted by purple. For further detail on the platform and transmission, please refer to our report [21],

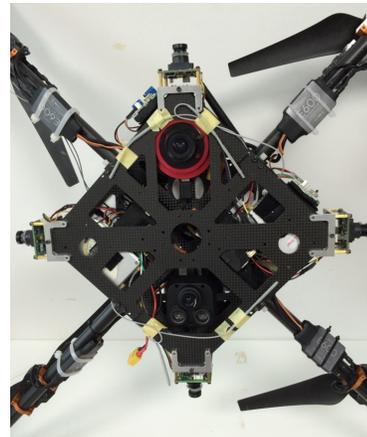


Fig. 16. The bottom view of the system. Red camera is for object tracking, Black camera & ultrasonic sensor module is the visual odometer. Four camera modules on each side can be either monocular or stereo.

- **Dynamic Bayesian Filtering:** Even with the rejection-based algorithm, noisy line observations are inevitable from time to time. They usually triggers largely translated wrong estimation of the UAV poses. To stabilize the estimation result, we introduce an EKF-based dynamic Bayesian filtering process. The output pose is filtered by taking the previous estimation results into account.

Figure 17 compares the localization result with ground-truth provided by an external motion tracking system. Without using the dynamic Bayesian programming, the maximum localization error is 0.23 meter and standard derivation is 0.12 meter. The localization results indicate the good performance in global localization for the UAV. Be reminded

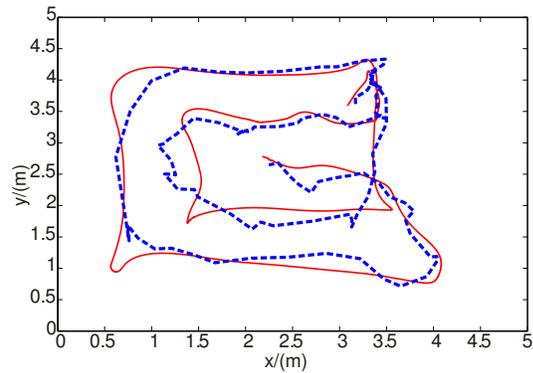


Fig. 17. Localization result in a 5m x 5m field. The red line (solid) is ground truth obtained by motion tracking system, while the blue (dashed) line is the result of our localization algorithm.

that the boundary detection results are the only extroceptive information for the localization. Since the accurate position estimation relies on an accurate attitude estimation, we omit the attitude analysis in this case due to page limit. For interested readers, please soon refer to our full length journal paper for more detail.

## VII. CONCLUSION

In this paper, in the context of IARC Mission-7, we introduced two global 6D pose estimation algorithms for UAVs. The first algorithm uses three known lines from the playground, plus an additional gravity measurement as input, leading to precise estimation in position and attitude. In the case when only a corner of the playground can be observed from the UAV, a second algorithm namely the corner-based algorithm could be adopted accordingly. We evaluated the behavior of the two algorithms under noisy observations in gravity and height estimation. The experiment result indicates the effectiveness of the algorithm. For future research, knowing that the IARC rules have changed to the case where the ground surfaces are now featured, we will fuse the proposed method with local feature detection and tracking, such as visual odometry algorithms.

## VIDEO SUPPLEMENT

The attached video comprises three parts. The first part shows the test field and the experiment using our IARC platform. The second part shows the results of localization in the test field, where the localization results are presented using the terminal in the bottom-right corner. The third part shows the line extraction results on the standard playground during the IARC 2014 competition.

## REFERENCES

- [1] E. Altug, J. P. Ostrowski, and R. Mahony, "Control of a quadrotor helicopter using visual feedback," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 1. IEEE, 2002, pp. 72–77.
- [2] M. Achtelik, M. Achtelik, Y. Brunet, M. Chli, S. Chatzichristofis, J.-D. Decotignie, K.-M. Doth, F. Fraundorfer, L. Kneip, D. Gurdan *et al.*, "Sfly: Swarm of micro flying robots," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 2649–2650.

- [3] M. Blosch, S. Weiss, D. Scaramuzza, and R. Siegwart, "Vision based mav navigation in unknown and unstructured environments," in *Robotics and automation (ICRA), 2010 IEEE international conference on*. IEEE, 2010, pp. 21–28.
- [4] S. Weiss, M. W. Achtelik, S. Lynen, M. Chli, and R. Siegwart, "Real-time onboard visual-inertial state estimation and self-calibration of mavs in unknown environments," in *Robotics and Automation (ICRA), 2012 IEEE International Conference on*. IEEE, 2012, pp. 957–964.
- [5] G. Klein and D. Murray, "Parallel tracking and mapping for small ar workspaces," in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*. IEEE, 2007, pp. 225–234.
- [6] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, vol. 1. IEEE, 2004, pp. 1–652.
- [7] D. Scaramuzza and F. Fraundorfer, "Visual odometry [tutorial]," *Robotics & Automation Magazine, IEEE*, vol. 18, no. 4, pp. 80–92, 2011.
- [8] L. Kneip, D. Scaramuzza, and R. Siegwart, "A novel parametrization of the perspective-three-point problem for a direct computation of absolute camera position and orientation," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 2969–2976.
- [9] L. Kneip, M. Chli, and R. Siegwart, "Robust real-time visual odometry with a single camera and an imu," in *Proc. of The British Machine Vision Conference (BMVC)*, Dundee, Scotland, August 2011.
- [10] M. Liu, C. Pradalier, Q. Chen, and R. Siegwart, "A bearing-only 2D / 3D-homing method under a visual servoing framework," in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Anchorage Convention District, 2010, pp. 4062–4067.
- [11] M. Liu, C. Pradalier, and R. Siegwart, "Visual homing from scale with an uncalibrated omnidirectional camera," *IEEE Transactions on Robotics*, 2013.
- [12] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 D visual servoing," *IEEE Transactions on Robotics and Automation*, vol. 15, no. 2, pp. 238–250, 1999.
- [13] N. Andreff, B. Espiau, and R. Horaud, "Visual servoing from lines," *The International Journal of Robotics Research*, vol. 21, no. 8, pp. 679–699, 2002.
- [14] X.-S. Gao, X.-R. Hou, J. Tang, and H.-F. Cheng, "Complete solution classification for the perspective-three-point problem," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 8, pp. 930–943, 2003.
- [15] L. Iocchi and D. Nardi, "Self-localization in the robocup environment," in *RoboCup-99: Robot Soccer World Cup III*. Springer, 2000, pp. 318–330.
- [16] C. F. Marques and P. U. Lima, "A localization method for a soccer robot using a vision-based omni-directional sensor," in *Robocup 2000: Robot Soccer World Cup IV*. Springer, 2001, pp. 96–107.
- [17] A. Ansar and K. Daniilidis, "Linear pose estimation from points or lines," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 5, pp. 578–589, 2003.
- [18] T. Q. Phong, R. Horaud, A. Yassine, and P. D. Tao, "Object pose from 2-d to 3-d point and line correspondences," *International Journal of Computer Vision*, vol. 15, no. 3, pp. 225–243, 1995.
- [19] D. Q. Huynh, "Metrics for 3d rotations: Comparison and analysis," *Journal of Mathematical Imaging and Vision*, vol. 35, no. 2, pp. 155–164, 2009.
- [20] M. Liu, "Personal website." [Online]. Available: <https://sites.google.com/site/mingliurobot/publication-1>
- [21] S. Yang, J. Ying, W. Liu, M. Liu, Z. Zhou, Y. Yu, M. Wang, and Z. Li, "A Comprehensive Quadrotor Research Platform For Vision-based Control and Navigation," in *submitted to the IEEE International Conference on Robotics and Automation (ICRA)*, 2015.