An Efficient Algorithm For UAV Indoor Pose Estimation Using Vanishing Geometry

Yuxiang Wang

Department of Electrical & Computer Engineering, National University of Singapore

1. Introduction

Pose estimation and localization are essential for indoor unmanned helicopter's automatic operations. We present an algorithm that uses onboard camera as sole sensor to provide 5 degree of freedom measurements of UAV's pose and position over an RGB colored track. The measurements can be robustly delivered in real-time on a thumb-sized embedded computer (Illustrated in Figure 1).

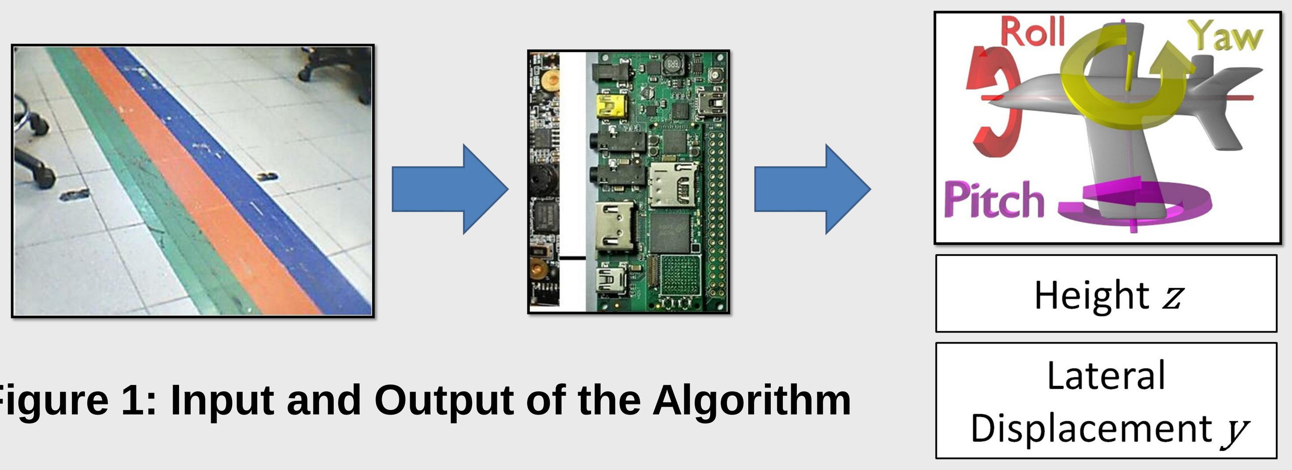

Figure 1: Input and Output of the Algorithm

The indoor and embedded environment impose great challenges on the accuracy and efficiency of the vision algorithm. The algorithm we proposed features an innovative linear-time line detection technique, an unconventional vanishing line estimation method, a constraint localization formulation and the derivation of the analytical expression of rotation matrix, hence pose, using the geometry at infinity. This algorithm works out well in simulation and fly-test, yielding an average of 25.6 Hz measurements.

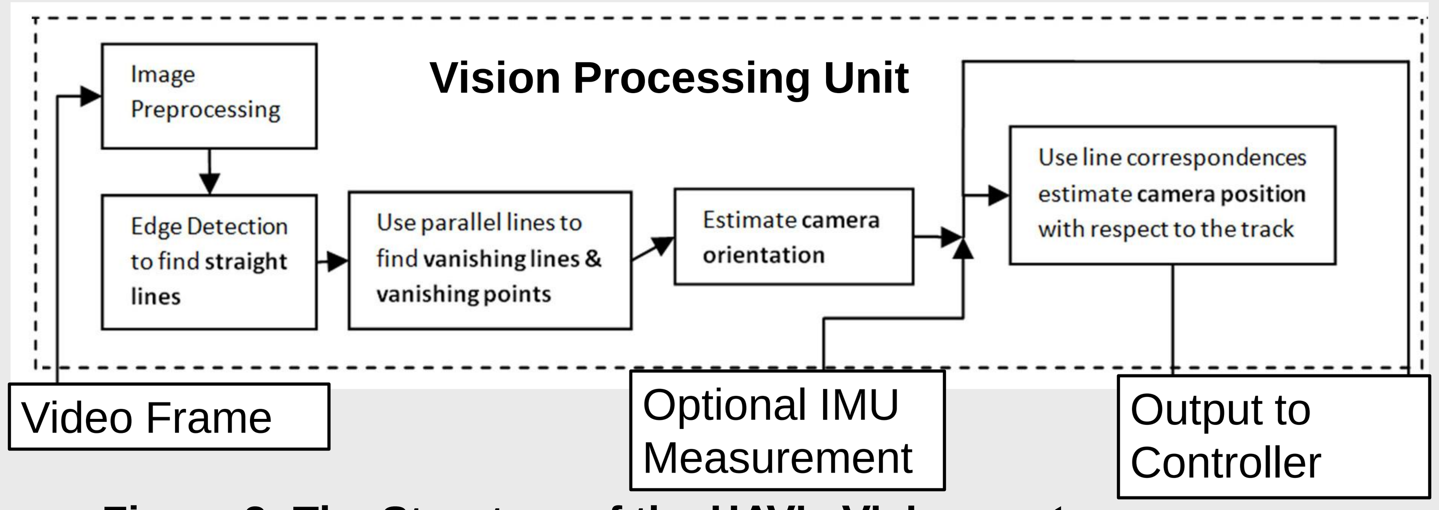

Figure 2: The Structure of the UAV's Vision system

As a side remark, the system has been successfully applied to a mini-UAV, codenamed "MerLion" in the Singapore Amazing Flying Machines Competition 2011. The UAV was able to autonomously follow the track and complete a series of tasks on its own, thus is awarded the overall championship. (see Figure 3)

Figure 3: "Merlion" Wins Championship

2. Image Processing and Feature Extraction

As is shown in Figure 2, image processing and feature extraction are the very first step in the algorithm. The efficiency of low level processing defines the complexity of the algorithm. In this case, it includes the detection of straight lines and the identification of vanishing point and line. As we will show in the next section, geometry at infinity is crucial in pose estimation.

2.1 Linear-time Line Extraction

Canny Edge Detection and Hough Transform are good, generic line detection algorithm. Yet, their large complexity makes it infeasible in our application. Considering the special pattern of interest, we proposed a much faster algorithm: "Sample and Fit". Key steps are summarized below:

1. Binary search for the top-most horizontal pixel sequence that contains [Green Red Blue] pattern
2. Take samples evenly in the lower half of image;
3. Preprocess each sample, perform 1D edge detections and then search for the segment groups that contain [Green Red Blue] pattern;
4. Rectify radial distortion;
5. Fit line using least square methods;
6. If the error is above a threshold, RANSAC is triggered. The line is then fit again with outliers excluded.

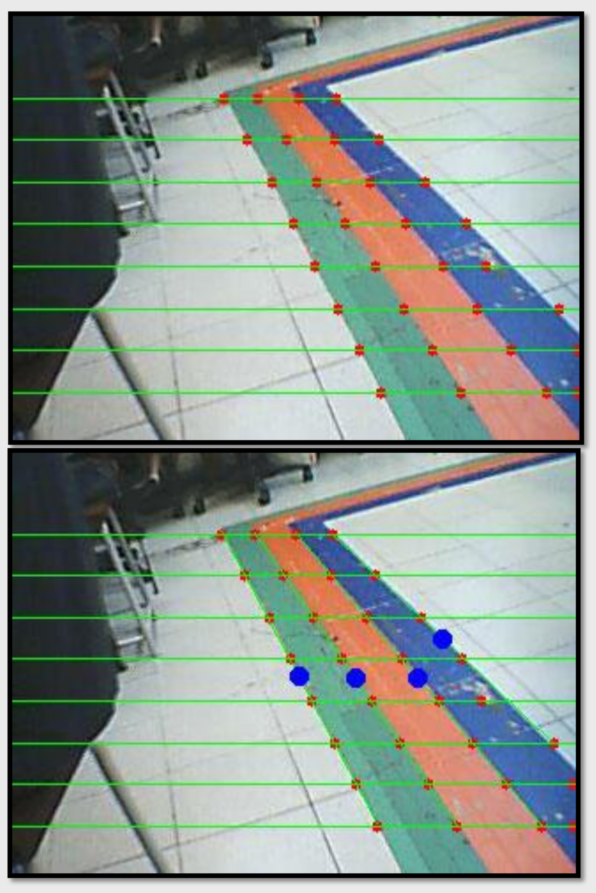

Figure 4: Sample and Fit

2.2 Vanishing Point and Vanishing Line

The next step is to calculate the vanishing point and vanishing line on ground plane(the horizon) from the lines we obtained. These hidden geometric entities in indoor environment play a crucial role in pose estimation.

- **Vanishing Point**: Obtained using Gold Standard Method (Levenberg Marquadt Optimization). (Illustrated in Figure 5)

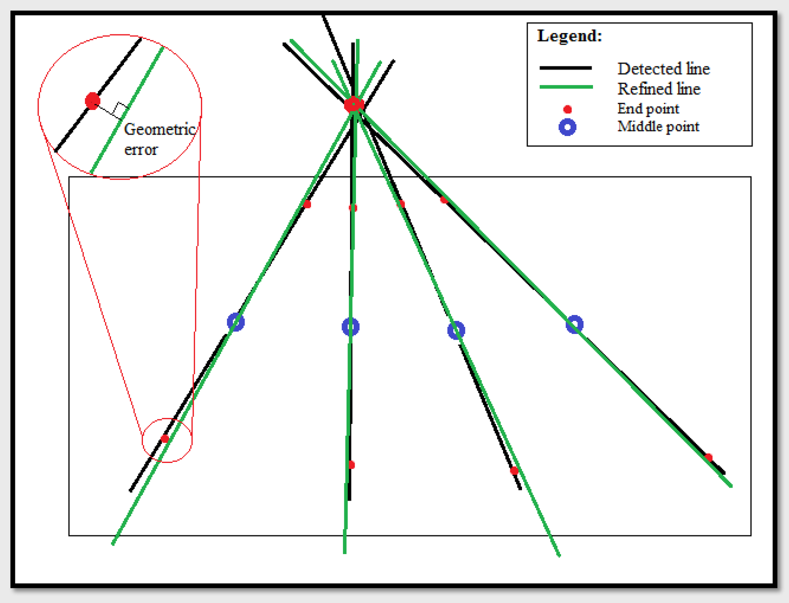

Figure 5: Minimizing Geometric Error

- **Vanishing Line**: Assume equal spacing between the coplanar parallel lines and apply Schaffalitzky's line grouping method, we may obtain the vanishing line as the first column of the projection matrix *A* in (3). This matrix *A* projects a group of standard parallel lines L_λ (2) onto the lines on image, hence can be determined with given data.

$$L_\lambda : ax + by + \lambda = 0 \qquad (1)$$

In Matrix form:

$$L_\lambda = \begin{bmatrix} 0 & a \\ 0 & b \\ 1 & 0 \end{bmatrix} \begin{pmatrix} \lambda \\ 1 \end{pmatrix} \qquad (2)$$

Apply homography M:

$$A = M^{-\top} \begin{bmatrix} 0 & a \\ 0 & b \\ 1 & 0 \end{bmatrix} \qquad (3)$$

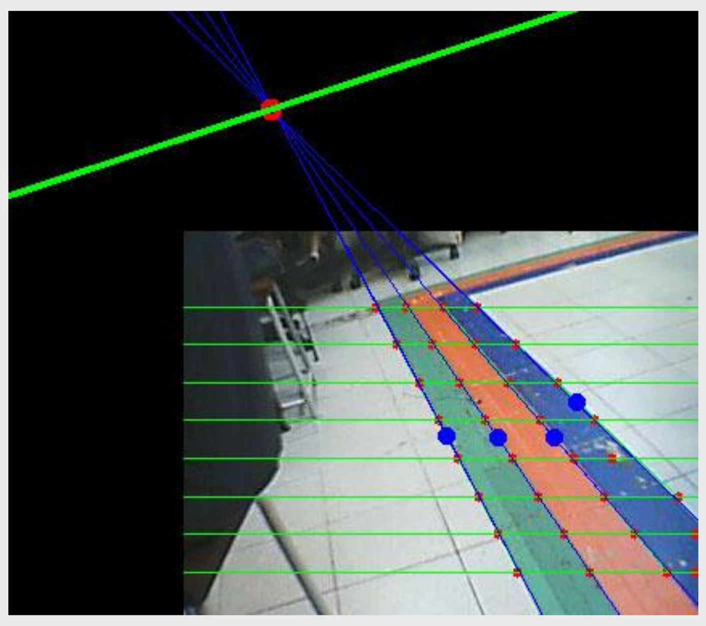

Figure 6: Detected Vanishing Point and Vanishing Line

3. Pose Estimation and Localization

Given the features obtained above, now we can compute the pose and position of the camera/UAV.

3.1 Pose Estimation

- **Closed-form Expression of Camera Pose**: Camera pose is encapsulated in a rotation matrix R. We have derived that given vanishing point X_∞ and vanishing line L_∞, the rotation matrix can be elegantly expressed using (4). Plus and minus sign can be determined by verifying the objects are in front of the camera. *K* is the intrinsic matrix of camera.

$$R = \left[\frac{K^{-1}X_\infty}{||K^{-1}X_\infty||} \quad \pm \frac{(K^\top L_\infty) \times (K^{-1}X_\infty)}{||K^\top L_\infty|| \, ||K^{-1}X_\infty||} \quad \pm \frac{K^\top L_\infty}{||K^\top L_\infty||} \right] \qquad (4)$$

- **UAV Pose**: Camera pose is then transformed to UAV pose with a pre-calibrated rotation matrix. Yaw, pitch, roll angles (*ψ, θ, φ*) are determined through the direct cosine representation of the rotation matrix.

3.2 Constrained Localization

UAV location *y* and *z* with respect to the track is calculated by forming a system of linear equations. For each line correspondence, we have one equation (5) .

$$\begin{bmatrix} 0 & -w_i(K^\top L_i)^\top & (K^\top L_i)^\top \\ w_i(K^\top L_i)^\top & 0 & 0 \end{bmatrix} \begin{pmatrix} r_1 \\ r_2 \\ -yr_2 - zr_3 \end{pmatrix} = 0 \qquad (5)$$

4. Simulation and Fly-test

- In simulation, measurements are compared to pre-defined trajectory.

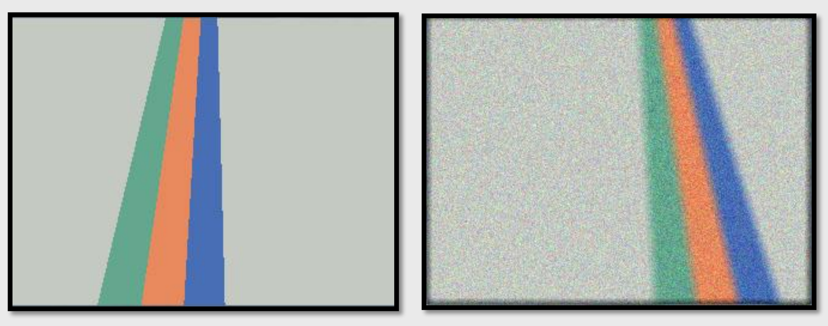

Figure 7: Simulation conditions:
Left: ideal; Right: Noisy and Blurry

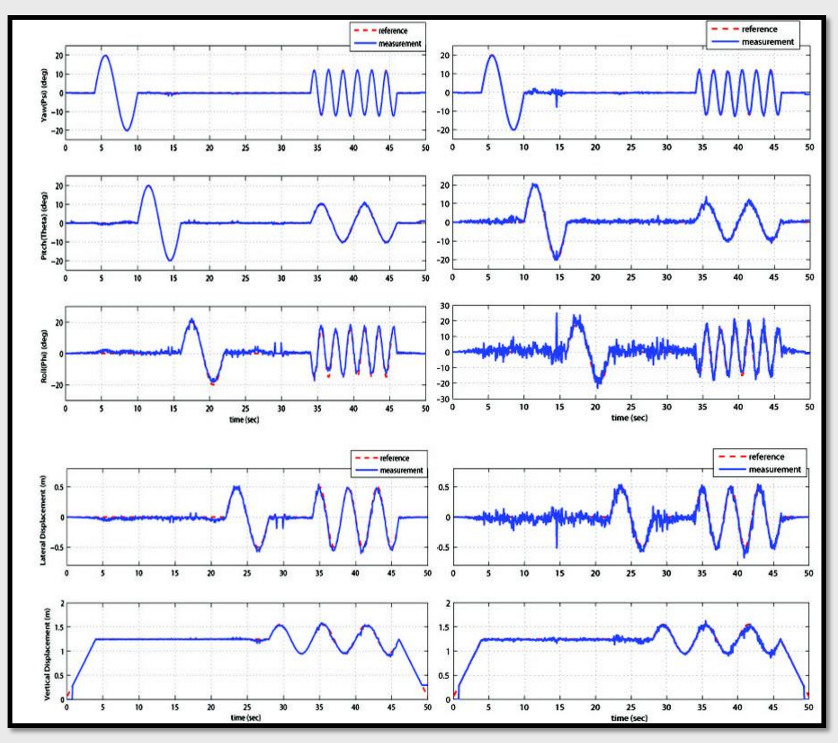

Figure 8: Simulation results:
Left: Ideal Simulation;
Right: Blur + Gaussian Noise,
From top to bottom: *ψ, θ, φ, y, -z*

- In fly-test, measurements are compared to IMU and sonar readings.

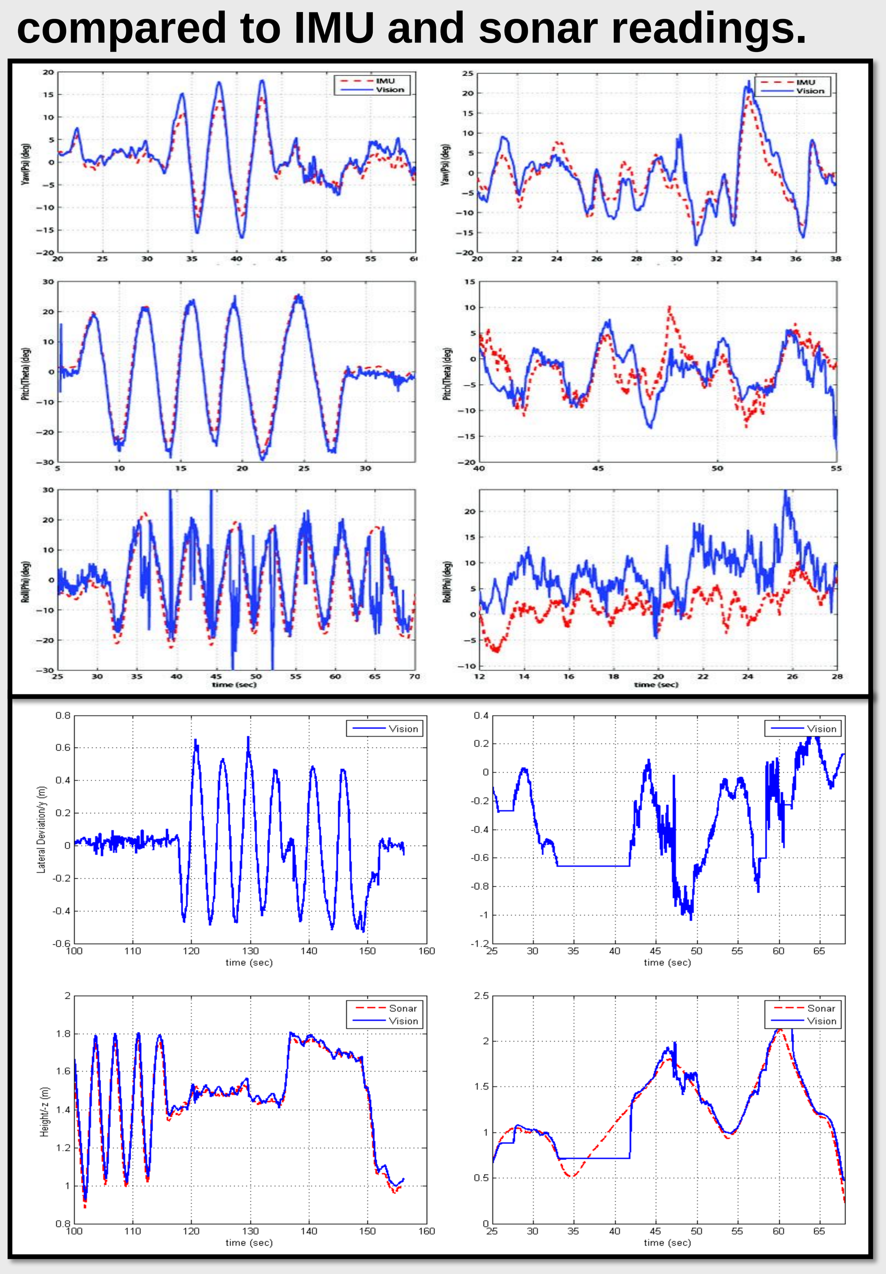

Figure 9: Fly-test results:
Left: Hand-held Oscillation;
Right: R/C Test Flight,
From top to bottom: *ψ, θ, φ, y, z*

Contact person: Yuxiang Wang, Dept. of Electrical & Computer Engineering; email address: yuxiangwang@nus.edu.sg