An efficient algorithm for UAV indoor pose estimation using vanishing geometry

Yuxiang Wang
Department of Electrical and Computer Engineering
National University of Singapore
yuxiangwang@nus.edu.sg

***Abstract*—This paper presents an efficient vision algorithm to recover the pose of an indoor unmanned aerial vehicle with respect to a structured scene based on vanishing geometry. Specifically, the ground plane vanishing line and a vanishing point are calculated from a set of equally-spaced parallel lines. By using such geometries, an elegant closed-form expression is derived to compute the rotation matrix of the camera. This rotation matrix is then used in a constrained estimation of the camera location. Moreover, a fast line detection algorithm is proposed where one dimensional edge detections are performed to a sampled subset of image pixels. Since only minimal features are utilized, complex algorithms such as RANSAC and Levenberg-Marquadt optimization remain computationally feasible in real time. The whole processing can reach an average of 25.6 frames per second on an embedded computer: Gumstix. Lastly, The proposed algorithms are successfully verified in simulations and fly tests conducted under several conditions.**

***Index Terms*—vanishing line, vanishing point, UAV indoor navigation, track following, pose estimation, real-time system**

I. Introduction

The unavailability of satellite signal in indoor context has forced autonomous robots, such as unmanned aerial vehicles (UAVs), to seek other forms of localization methods. Due to the low cost and flexibly programmable nature of cameras, vision-based localization has become a popular choice for researchers and engineers. To perform the same function as GPS, frames captured by a camera must be processed in a way such that the location and orientation of the camera can be extracted. This is often characterized as the camera calibration problem for extrinsic parameters.

To solve such a problem, the geometry of vanishing point plays an important role. Wang and Tsai transformed vanishing line directly to roll angles in [12], but extra information was required to uniquely determine yaw and pitch. Vanishing points were also used for pose estimation in [4], [8], [2], but they required the points to be pointing at orthogonal directions. In addition, a pose estimation using single vanishing point normal to ground plane was proposed by Xu in [13]. Yet, it recovers only pitch and roll.

Although aforementioned works identified the importance of vanishing geometry, they failed to address the common situation where only one vanishing point is available in the image,e.g. road following. This is exactly the case in our situation where the UAV is to fly autonomously according to the colored track on the floor (illustrated in Figure 1). Though in a sense, vision path following is considered as a solved problem, the existing approaches do not require explicit recovery of the camera pose [10], [7], [5]. The concept of vanishing point was only used to provide heading feedback [5]. Even for the visual servoing of the UAV, near hovering assumptions (zero pitch and roll angles) are being made, such as in [11], [1], [6]. The near hovering assumption might be reasonable for low-speed coaxial rotors with self-balancing hardware, but is not valid for quad-rotor and conventional helicopter.

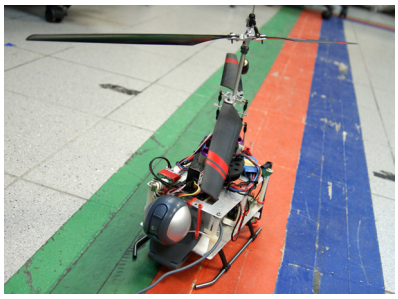


Fig. 1. Micro aerial vehicle: "FeiLion" sitting on colored track.

This paper proposes an efficient algorithm to estimate UAV pose as well as vertical and lateral displacement with respect to the track in Figure 1. An elegant analytical expression of rotation matrix using vanishing point and line is derived, which reveals the underlying connections between pose and vanishing geometry. The rotation matrix is further applied in a constrained estimation of the camera's external matrix, in which the lateral and vertical displacement of the camera relative to the track are found by solving a system of over-determined linear equations. The measurements obtained will provide essential information for accurate control of the UAV and facilitate various autonomous tasks.

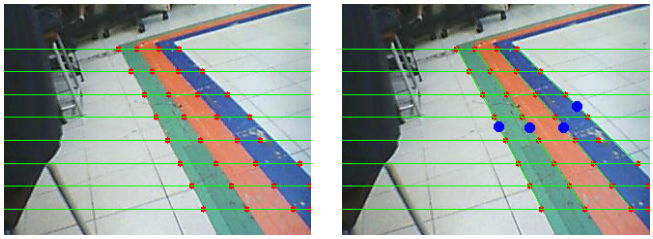
The organization of the paper is as follows. Section II presents the image processing to efficiently extract the line features on the track. Section III describes the pose estimation approach for the UAV using the vanishing point and line. Section IV shows the results and analysis of the simulation and actual fly-tests. Finally, we draw conclusions and present future work in Section V.

II. Image Processing and Feature Extraction

A. Line Extraction

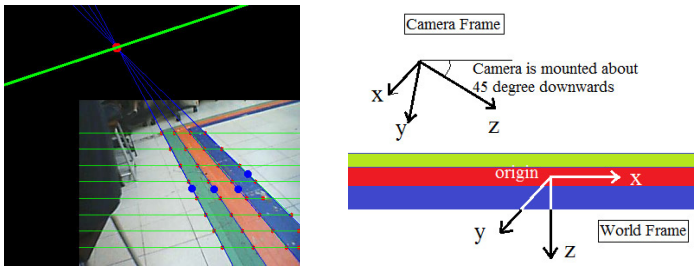
The high complexity of Canny edge detection and Hough Transform makes it unsuitable for onboard processing. In this particular case where a track of colored bands is available, much more efficient algorithms can be expected. Inspired by the idea proposed in [6], we come up with a more reliable algorithm. The 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;



(a) Edge detection (b) RANSAC line fitting

Fig. 2. Illustration of line segmentation.



(a) Detected vanishing point and line. (b) The coordinate systems.

Fig. 3. An illustration of processed image and coordinate system.

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 outlier excluded.

There are several implications on efficiency within this new algorithm. First of all, sampling the image downsizes the image from 320×240 to $320 \times k$, where k is the number of samples. This reduces the two-dimensional problem to a multiple of one-dimensional ones. Secondly, the process of undistortion can now be performed only to the points of interest. Thirdly, RANSAC is less taxing with small k. It is even feasible to try out all combinations. Lastly, compared to Hough Transform, this method saves the trouble in filtering and reordering of detected lines. In overall, this line detection algorithm retains a complexity of $O(n)$ when k is small and is thus an ideal method in our application.

B. Finding the Vanishing Point and Vanishing Line

The next step is to calculate the vanishing point and vanishing line from the lines we obtained.

In theory, parallel lines should intersect at one single vanishing point on image. However, due to the noisy real image, this is often not the case. By the standard method in [3], we calculate the least square solution and then refined it with Levenberg-Marquadt algorithm.

According to [9], vanishing line can be expressed analytically with three coplanar equally-spaced parallel lines. With four such lines on ground planes in our case, the problem can be formulated into an over-determined system of linear equations. Nonetheless, such formulation is not explicitly disclosed in that paper. Hence, our formulation will be presented here as an example of applying Schaffalitzky's findings [9].

A group of parallel lines can be expressed in standard form:

$$\boldsymbol{L}_\lambda : ax + by + \lambda = 0$$

or its normal vector $\boldsymbol{L}_\lambda$ in matrix form:

$$\boldsymbol{L}_\lambda = \begin{bmatrix} 0 & a \\ 0 & b \\ 1 & 0 \end{bmatrix} \begin{pmatrix} \lambda \\ 1 \end{pmatrix} \tag{1}$$

which defines a projection from 1D projective space $\boldsymbol{P}^1$ to 2D projective space $\boldsymbol{P}^2$. By the definition of projective space, this relation is invariant to projective transformation (homography). Note that when λ equals to consecutive integers, the line group has equal spacing. Define $\boldsymbol{M}^{-\mathrm{T}}$ to be the line transformation matrix between world plane and image plane. By multiplying $\boldsymbol{M}^{-\mathrm{T}}$ on both side of (1), we obtain a projection matrix from points in $\boldsymbol{P}^1$ to images of $\boldsymbol{L}_\lambda$.

$$\boldsymbol{A} = \boldsymbol{M}^{-\mathrm{T}} \begin{bmatrix} 0 & a \\ 0 & b \\ 1 & 0 \end{bmatrix}$$

Also note that the first column of the projection matrix in (1) is the vanishing line in $\boldsymbol{P}^2$, then the first column of $\boldsymbol{A}$ is the vanishing line in the image. Let $\boldsymbol{A} = [\boldsymbol{a}_1^{\mathrm{T}},\ \boldsymbol{a}_2^{\mathrm{T}},\ \boldsymbol{a}_3^{\mathrm{T}}]^{\mathrm{T}}$, and the i-th image line is given by $\boldsymbol{L}_i = [x_i,\ y_i,\ 1]^{\mathrm{T}}$. Corresponding i-th object point $\boldsymbol{X}_i = [i,\ 1]^{\mathrm{T}}$. For each correspondence, the two linear equations are:

$$\begin{bmatrix} 0 & -\boldsymbol{X}_i^{\mathrm{T}} & y_i \boldsymbol{X}_i^{\mathrm{T}} \\ \boldsymbol{X}_i^{\mathrm{T}} & 0 & -x_i \boldsymbol{X}_i^{\mathrm{T}} \end{bmatrix} \begin{pmatrix} \boldsymbol{a}_1 \\ \boldsymbol{a}_2 \\ \boldsymbol{a}_3 \end{pmatrix} = \boldsymbol{0} \tag{2}$$

A total of four equally-spaced lines give 8 independent equations, which is more than enough to determine the matrix $\boldsymbol{A}$ and hence the vanishing line. An illustration of the detected vanishing line and vanishing point is shown in Fig. 3(a).

III. Pose Estimation and Constrained Localization

Without losing any generality, we choose the world coordinate frame with respect to the track. Specifically, the forward middle line of the track is x direction and the orthogonal direction to the left is y direction. By convention this is a right-hand coordinate, thus z-axis points vertically upwards (see Fig. 3(b)). The estimated camera pose and location in the following parts will be in term of this frame.

A. Pose Estimation

Given the vanishing point on image $\boldsymbol{X}_\infty = [x,\ y,\ 1]^{\mathrm{T}}$, we have

$$\boldsymbol{X}_\infty \propto \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}] \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} = \boldsymbol{K}[\boldsymbol{r}_1\ \boldsymbol{r}_2\ \boldsymbol{r}_3\ \boldsymbol{t}] \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

where $\boldsymbol{R} = [\boldsymbol{r}_1\ \boldsymbol{r}_2\ \boldsymbol{r}_3]$. Hence, we obtain

$$\boldsymbol{r}_1 = \frac{\boldsymbol{K}^{-1}\boldsymbol{X}_\infty}{||\boldsymbol{K}^{-1}\boldsymbol{X}_\infty||}$$

Let the ground plane vanishing line $\boldsymbol{L}_\infty = [p,\ q,\ 1]^{\mathrm{T}}$. Since all points on ground plane has a form $\boldsymbol{x} = [x,\ y,\ 0,\ w]^{\mathrm{T}}$, the projection matrix reduces to a homography: $\boldsymbol{H} = \boldsymbol{K}[\boldsymbol{r}_1\ \boldsymbol{r}_2\ \boldsymbol{t}]$. Thus the transformation equation is:

$$\boldsymbol{L}_\infty \propto \boldsymbol{H}^{-\mathrm{T}}\boldsymbol{l}_\infty = \boldsymbol{H}^{-\mathrm{T}} \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Inverting the homography to another side and expand the term, we obtain

$$\begin{pmatrix} \boldsymbol{r}_1^{\mathrm{T}} \\ \boldsymbol{r}_2^{\mathrm{T}} \\ \boldsymbol{t}^{\mathrm{T}} \end{pmatrix} \boldsymbol{K}^{\mathrm{T}}\boldsymbol{L}_\infty \propto \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

The first two equations imply that the vector $\boldsymbol{K}^{\mathrm{T}}\boldsymbol{L}_\infty$ is orthogonal to both $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$. This is exactly the characteristic of $\boldsymbol{r}_3$. Thus, up to direction ambiguity, we have derived:

$$\boldsymbol{r}_3 = \pm\frac{\boldsymbol{K}^{\mathrm{T}}\boldsymbol{L}_\infty}{||\boldsymbol{K}^{\mathrm{T}}\boldsymbol{L}_\infty||}$$

The other vector $\boldsymbol{r}_2$ is simply $\boldsymbol{r}_3 \times \boldsymbol{r}_1$ by right hand rule. Taking into consideration the physical meaning of $\boldsymbol{r}_3$: the unit vector in z direction of the world frame represented in camera frame, we can easily resolve the ambiguity by looking at camera's principal ray direction.

To sum up the derivation, rotation matrix can be expressed elegantly with one vanishing point and vanishing line:

$$\boldsymbol{R} = \left[\frac{\boldsymbol{K}^{-1}\boldsymbol{X}_\infty}{||\boldsymbol{K}^{-1}\boldsymbol{X}_\infty||} \pm \frac{(\boldsymbol{K}^{\mathrm{T}}\boldsymbol{L}_\infty)\times(\boldsymbol{K}^{-1}\boldsymbol{X}_\infty)}{||\boldsymbol{K}^{\mathrm{T}}\boldsymbol{L}_\infty||\boldsymbol{K}^{-1}\boldsymbol{X}_\infty||} \pm \frac{\boldsymbol{K}^{\mathrm{T}}\boldsymbol{L}_\infty}{||\boldsymbol{K}^{\mathrm{T}}\boldsymbol{L}_\infty||}\right]$$

B. Constrained Localization

We should notice that no information is given along the track's longitudinal direction, thus it is infeasible to recover x coordinate of camera center from the images. As a result, a shifting world coordinate system is adopted such that x is always zero.

Based on this assumption and the rotation matrix in Part A, the task of self-localization reduces to a constrained estimation of only two values, i.e., the lateral displacement and height. Given metric information of the track, the solution can be obtained using correspondences. Let the camera center $\tilde{\boldsymbol{C}} = [0,\ y,\ z]^{\mathrm{T}}$, we have

$$\boldsymbol{t} = -\boldsymbol{R}\tilde{\boldsymbol{C}} = -y\boldsymbol{r}_2 - z\boldsymbol{r}_3 \tag{3}$$

Also let $\boldsymbol{l}_i$ be Line i in on ground plane and let $\boldsymbol{L}_i$ be the image of $\boldsymbol{l}_i$ in image plane. Since the lines are all parallel to x direction, we have a reduced form of $\boldsymbol{l}_i = [0,\ 1,\ w_i]^{\mathrm{T}}$. Then the homography between ground plane and image plane relating the line correspondences can be written as:

$$\boldsymbol{l}_i \propto \begin{pmatrix} \boldsymbol{r}_1^{\mathrm{T}} \\ \boldsymbol{r}_2^{\mathrm{T}} \\ -y\boldsymbol{r}_2^{\mathrm{T}} - z\boldsymbol{r}_3^{\mathrm{T}} \end{pmatrix} \boldsymbol{K}^{\mathrm{T}}\boldsymbol{L}_i$$

By the direct linear transform formulation:

$$\begin{bmatrix} 0 & -w_i(\boldsymbol{K}^{\mathrm{T}}\boldsymbol{L}_i)^{\mathrm{T}} & (\boldsymbol{K}^{\mathrm{T}}\boldsymbol{L}_i)^{\mathrm{T}} \\ w_i(\boldsymbol{K}^{\mathrm{T}}\boldsymbol{L}_i)^{\mathrm{T}} & 0 & 0 \end{bmatrix} \begin{pmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \\ -y\boldsymbol{r}_2 - z\boldsymbol{r}_3 \end{pmatrix} = \boldsymbol{0}$$

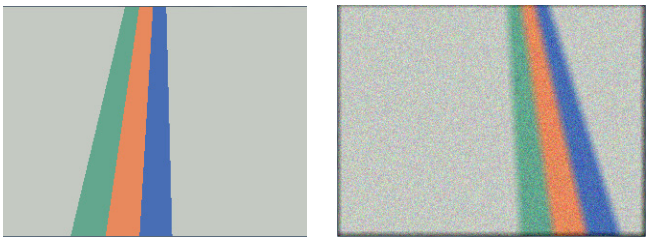
Apparently, only the first equation is relevant, so for each i, we have:

$$[(\boldsymbol{K}^{\mathrm{T}}\boldsymbol{L}_i)^{\mathrm{T}}\boldsymbol{r}_2 \quad (\boldsymbol{K}^{\mathrm{T}}\boldsymbol{L}_i)^{\mathrm{T}}\boldsymbol{r}_3] \begin{pmatrix} y \\ z \end{pmatrix} = -w_i(\boldsymbol{K}^{\mathrm{T}}\boldsymbol{L}_i)^{\mathrm{T}}\boldsymbol{r}_2$$

Piling up the equations and applying singular value decomposition (SVD), the least square solution of y and z is obtained. Note that there is only one useful equation from each line correspondence and only two line correspondences are linearly independent. Thus, we have just enough equations for the two unknowns.

Likewise, counting the overall number of equations used and degree of freedom is interesting too. All in all, four linearly independent equations coming from a pencil of lines and one extra constraint of equal spacing add up to a total of five independent constraints. Correspondingly, there are exactly five degrees of freedom to be determined.

In a nutshell, by considering the geometric meaning of vanishing features, this algorithm nicely by-passes the non-linearity within the quadratic equations that define rotation matrix, and hence makes the process of calibrating the camera's extrinsic parameters more efficient.



a. Ideal image b. Blurry and noisy image

Fig. 4. Illustration of simulation conditions

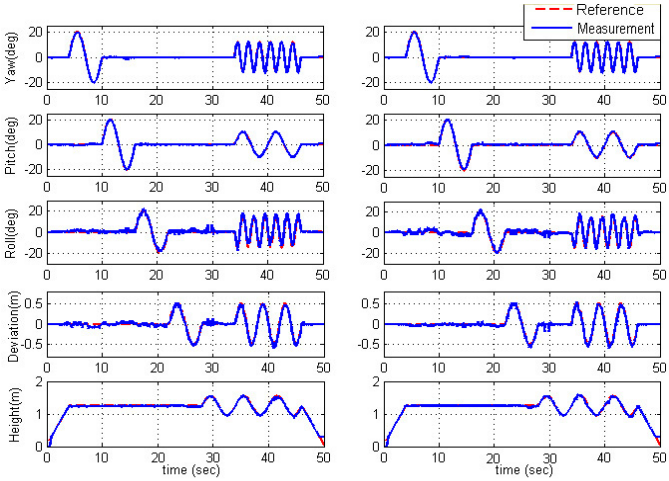


Fig. 5. Simulation Results.
Left: Ideal Image; Right: Blurry AND noisy image.

IV. Experiments

In order to verify the integrity of the algorithm, a series of simulation and fly-tests were carried out.

A. Simulation

1) Methodology: In the simulation, a virtual UAV flies according to a predefined trajectory above the track. Video frames generated by the virtual onboard camera are then fed into the vision algorithm realized using OpenCV.

For comparison, two fly conditions are assumed: (a) ideally stable quasi-static flight with normal lighting condition; (b) UAV flight with vibration and high-ISO camera configuration (simulated using blurring and Gaussian noise). A sample of these images is given in Fig. 4.

2) Simulation Results: The results of both conditions are shown in Fig. 5. As we may observe, the measurement (solid blue line) follows the reference (dashed red line) closely. There is no significant difference between ideal case and deteriorated case. This implies that the algorithm is robust against harsh conditions.

B. Fly Test

1) Platform: The fly test is conducted on a coaxial rotor-craft named "FeiLion" [11], where a newly developed vision system is installed. The onboard vision system consists of a Logitech C500 webcam and an embedded computer Gumstix Overo Fire (see Fig. 6). The algorithm is implemented on Gumstix to measure the position and orientation of FeiLion in real time.

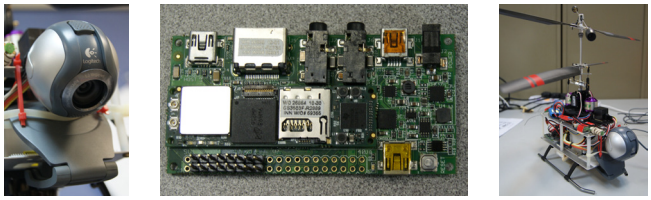


Fig. 6. Illustration of hardware.
Left: Logitech Webcam; Middle: Gumstix with expansion board; Right: FeiLion

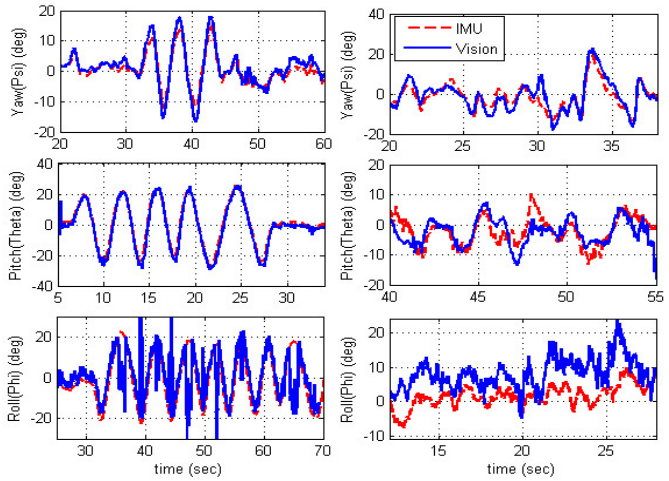


Fig. 7. FlyTest Results. Left: Hand-hold Motion; Right: R/C Motion.

2) Methodology: Similar to simulation, two experiments are conducted: (a) Hand-held UAV motion: for easy maneuver of decoupled testing of different channels; (b) R/C UAV flight: actual application scenario subjected to severe vibration and noise. In both experiments, measurements are compared to the IMU reading of yaw/pitch/roll angles.

3) Fly Test Results: The angle measurements of vision and IMU are plotted in comparison in Fig. 7. In both experiments, vision provides sensitive and drift-free yaw and pitch readings. The performance is comparable to, if not better than that of IMU. Roll angle, however, is noisier than the IMU readings and exhibits fallacious behavior when the angle gets large. The position measurements of the first experiment is also plotted (see Fig. 8), motion on both directions during the experiment are reflected as it is on the plot. The average frame rate in the two experiments is 25.6 FPS. Such real-time performance is sufficient for UAV's indoor navigation.

V. Conclusion

In this paper, we provide critical insight into how hidden geometric entities such as a vanishing line (horizon) in indoor environment can provide useful information. Specifically, a new pose estimation and self-locating algorithm for an indoor UAV is proposed and discussed with great details. The algorithm features an innovative linear-time line detection technique, an unconventional vanishing line estimation method, a constrained localization formulation and the derivation of the analytical expression of rotation matrix using geometry at infinity. In addition, the vision measurements are verified in simulation and fly-tests.

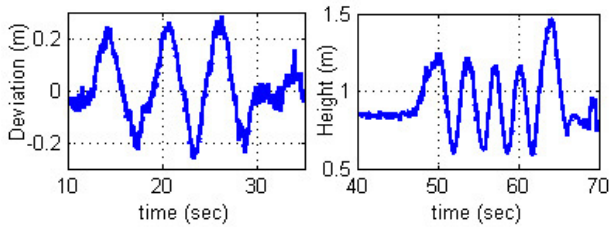


Fig. 8. Hand-held Motion Position Plot. Left: Deviation; Right: Height.

This work is not complete yet. In order to realize fully autonomous actions, dynamic models and control schemes must be implemented as well. Also, the vision measurement can be further improved in several ways, for instance, by filtering of the results. The temporal-spatial information between frames might be useful too. Lastly, fusing vision measurements and the pose angle readings from IMU will probably produce even better measurements.

Acknowledgment

Author of the paper would like to express his sincere gratitude to Dr. F. Lin, Mr. S.K. Phang and Mr. F. Wang for their invaluable help and encouragement.

References

[1] E. Frew, T. McGee, , Z. Kim, "Vision-based road-following using a small autonomous aircraft," *Proceedings of Aerospace Conference 2004*, pp. 3006-3015, 2004.

[2] L. Grammatikopoulos, G. Karras and E. Petsa, "An automatic approach for camera calibration from vanishing points," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 62, pp. 64–76, 2007.

[3] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*, 2nd edn., The Edinburgh Building, UK: Cambridge University Press, 2003.

[4] Z. Kim, *Geometry of Vanishing Points and its Application to External Calibration and Realtime Pose Estimation*, UC Berkeley: Institute of Transportation Studies, 2006. Retrieved from: http://escholarship.org/uc/item/1m71z3t3

[5] S.-P. Liou and R. C. Jain, "Road following using vanishing points," *Computer Vision, Graphics, and Image Processing,*, vol. 39, pp. 116–130, 1987.

[6] S. K. Phang, J.J. Ong, R.T.C. Yeo, B.M. Chen and T.H. Lee, "Autonomous Mini-UAV for indoor flight with embedded on-board vision processing as navigation system," *Proceedings of the IEEE R8 International Conference on Computational Technologies in Electrical and Electronics Engineering*, Listvyanka, Irkutsk, Russia, pp. 722–727, 2010.

[7] Q. Fang and C. Xie, "A study on intelligent path following and control for vision-based automated guided vehicle," *Proceedings of the 5th World Congress on Intelligent Control and Automation*, vol. 6, pp. 4811–4815, 2004.
Journal

[8] C. Rother, "A new approach to vanishing point detection in architectural environments," *Image and Vision Computing*, vol. 20, pp. 647–655, 2002.

[9] F. Schaffalitzky, A. Zisserman, "Planar grouping for automatic detection of vanishing lines and points," *Image and Vision Computing*, vol. 18, pp. 647–658, 2000.

[10] C. Thorpe; M.H. Hebert,T. Kanade,S.A. Shafer, "Vision and navigation for the Carnegie-Mellon Navlab," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 10, pp. 362–373, 1988.

[11] F. Wang, T. Wang, B.M. Chen and T.H. Lee, "An indoor unmanned coaxial rotorcraft system with vision positioning," *Proceedings of 8th IEEE International Conference on Control and Automation* , pp. 291–296, 2010.

[12] L.L. Wang, W.H. Tsai, "Camera calibration by vanishing lines for 3-D computer vision," *Pattern Analysis and Machine Intelligence, IEEE Transactions*, vol. 13, pp. 370–376, 1991.

[13] W. Xu, P. Li, B. Han, "An attitude estimation method for MAV based on the detection of vanishing point," *Proceedings of the 8th World Congress on Intelligent Control and Automation*, Jinan, China, pp. 6158–6162, 2010.