

Visual-Inertial Based Autonomous Navigation

Francisco de Babo Martins¹, Luis F. Teixeira², and Rui Nóbrega³

¹DEEC, FEUP, Faculdade de Engenharia, Universidade do Porto, Portugal.

²DEI, FEUP, Faculdade de Engenharia, Universidade do Porto, Portugal.

³INESC TEC, Instituto de Engenharia de Sistemas e Computadores - Tecnologia e Ciência, Porto, Portugal.

Rua Dr. Roberto Frias, s/n 4200-465, Porto, Portugal

{ee09153,luisft,ruinobrega}@fe.up.pt

<http://www.fe.up.pt>

Abstract. This paper presents an autonomous navigation and position estimation framework which enables an Unmanned Aerial Vehicle (UAV) to possess the ability to safely navigate in indoor environments. This system uses both the on-board Inertial Measurement Unit (IMU) and the front camera of a AR.Drone platform and a laptop computer were all the data is processed. The system is composed of the following modules: navigation, door detection and position estimation. For the navigation part, the system relies on the detection of the vanishing point using the Hough transform for wall detection and avoidance. The door detection part relies not only on the detection of the contours but also on the recesses of each door using the latter as the main detector and the former as an additional validation for a higher precision. For the position estimation part, the system relies on pre-coded information of the floor in which the drone is navigating, and the velocity of the drone provided by its IMU. Several flight experiments show that the drone is able to safely navigate in corridors while detecting evident doors and estimate its position. The developed navigation and door detection methods are reliable and enable an UAV to fly without the need of human intervention.

Keywords: UAV, AR. Drone, indoor autonomous navigation, vanishing point, door detection, location estimation

1 Introduction

In recent years, UAVs have become affordable and relevant in several research areas such as military applications and surveillance systems. These reliable and low-cost devices are normally equipped with high definition cameras and can be used as an autonomous image gathering device. The captured images can be processed in order to extract and obtain useful information that may be used for a variety of tasks and applications.

Being able to hover and fly laterally at low speeds, makes UAVs an ideal platform to accomplish different military and civilian tasks such as reconnaissance support in hazardous zones, visual surveillance and inspection. In addition,

some relevant industries are starting to use drones for other tasks beyond surveillance (e.g Amazon's "Prime Air"). Moreover, the most important task in order to achieve UAV autonomy is autonomous navigation. This may prove useful in a near future for tasks in indoor environments such as indoor transportation, object retrieval (e.g a missing part in an assembly line), monitoring misplaced books in a library and autonomously reporting sports events[1].

Although years of research of GPS position and data tracking have improved outdoor navigation and localization, in environments such as indoors or dense urban areas where maps are unavailable and the GPS signal is weak, an UAV will operate in high hazardous regions, running the risk of becoming lost and colliding with obstacles.

Since the scope of this project consists of enabling an UAV to autonomously navigate in an GPS impaired environment, the main challenge is using visual odometry and on-board IMU to develop navigation and position estimation algorithms to achieve an autonomous and robust navigation.

The various contributions that resulted from all the work developed on the scope of this paper are the development of a computer vision framework for autonomous navigation and position estimation in corridors and the implementation of a vanishing point and door detection methods.

2 Related Work

There has been a significant amount of work in regards to making robotic systems understand the environment they are in and reacting accordingly towards reaching a goal.

A team from the University of Texas [14] developed an assistive guide-robot that provides vision-based navigation as well as laser-based obstacle and collision avoidance to a visually-impaired person. In [13] the authors conducted experiments with a visual system able to detect obstacles as their relative size changes when the drone is approaching them.

Moreover, [10] presents a method able to autonomously steer a Micro Aerial Vehicles (MAV) towards distant features called vistas while building maps of unexplored regions. Also, in Bills et al. [9] is presented a method that classifies the types of indoor environments, such as corridors and stairs, while the MAV is traversing them, by using the perspective cues vision algorithm for estimating the desired steering direction.

Other approaches involved using stereo cameras[15] and the Microsoft Kinect RGB-D sensors [16] to solve the SLAM problem and find the best solution for a robust state estimation and control methods. These approaches perform local estimation of the vehicle position and build a 3D model of the environment (SLAM) to plan trajectories through an environment.

Finally, a team from University of Texas [8] developed a method aimed at navigating autonomously in indoor environments (corridors) and industrial environments (production lines) and detecting and avoiding obstacles (people). The navigation was accomplished by using the vanishing point algorithm, the Hough

transform for wall detection and avoidance and HOG (histogram of oriented gradients) descriptors using SVM (support vector machine) classifiers for detecting pedestrians.

However, when there is no GPS coverage, some visual navigation autonomous vehicle approaches struggle to estimate with high confidence the location of the drone indoors while enabling the drone to maintaining a stable wall-avoidance trajectory. The approach described in this paper focuses on the autonomous aerial navigation using the frontal camera of a drone to detect the vanishing point of each frame and thus keeping the drone aligned with it and the door detection for estimating the position of the drone.

3 System Architecture

The proposed method consists of a modular algorithm designed using image processing methods. The result of the algorithm is a generated map in which the position of the drone and the doors of a corridor are estimated and displayed.

As most indoor environments satisfy the Manhattan World assumption [2], i.e., most planes lie in one of three mutually orthogonal orientations, all the work developed made use of this assumption.

3.1 Framework

A detailed framework was created to tackle the challenge of autonomous navigation and position estimation using a monocular camera. A graphical representation of the algorithm is shown in Figure 1.

This framework is devised into two parts:

- **Hardware Side:** Image and sensory data from the vehicle.
- **Software Side:** All the image processing methods.

In order to reduce the on-board payload data processing, the drone sends the acquired data to a laptop running a linux distribution with an i7-4790K (4.40 GHz) processor.

The hardware side is comprised of the drone mechanism and stability, the camera input and the medium through which the video feed is transmitted. The computer side processes a series of algorithms and methods that process the video feed and output the drone commands.

3.2 Setup

The live video feed comes from a 720p camera on-board the AR. Drone. The video is recorded at 30 frames per second while the drone navigates in a corridor.

The AR. Drone has limited computational power which may prove a problem as image processing using computer vision algorithms require a lot of resources. Therefore, the analysis of the video feed of the drone is performed remotely in real-time using the OpenCV library for C++ in a computer that is connected to

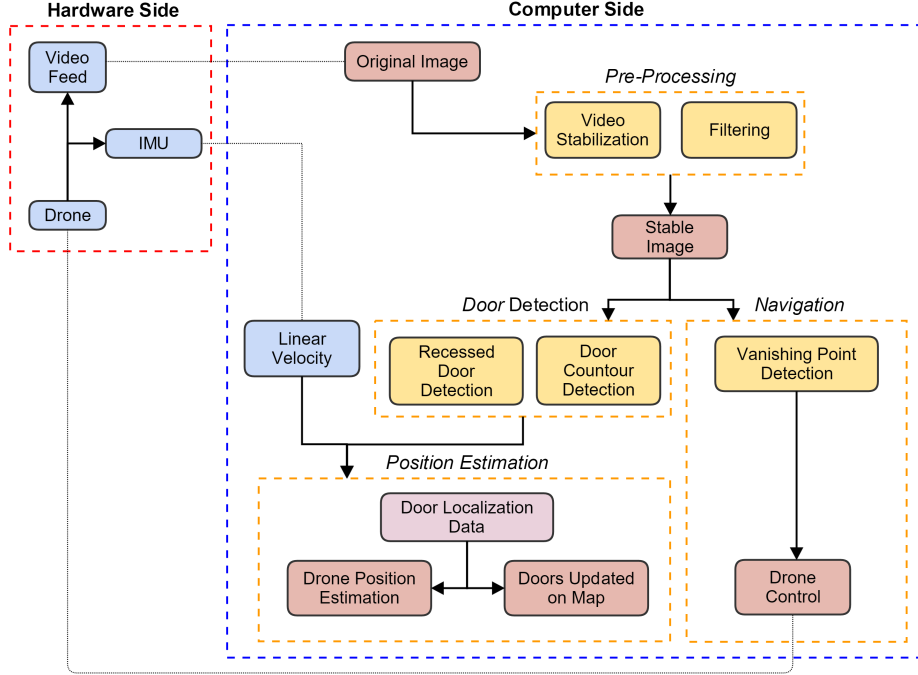


Fig. 1. The developed framework which bridges the gap between the hardware of the drone and the external processing unit using Wifi.

the AR. Drone via wireless. The communication is established using the Robot Operation System (ROS) software framework running on the computer and using a custom made third-party driver. The reason for using this driver rather than a built-in driver is because the development team behind ROS has yet to build one. Therefore, the Ardrone_autonomy¹ library was used.

3.3 Pre-Processing

In order to extract useful information from the live video feed, some methods were implemented having in mind that the AR. Drone is susceptible to oscillations caused by either external and uncontrollable factors, like airflows, as well as internal factors such as its own dynamic.

a) Video Stabilization

The aim of the image stabilization method developed for this project was providing useful stabilization for a forward-moving and panning video stream.

¹ https://github.com/AutonomyLab/ardrone_autonomy



Fig. 2. Drone navigating inside a corridor.

The first step is to find the transformation from the previous to the current frame consisting in a rigid Euclidean transform, achieved by using optical flow for all frames. In order to detect corner feature points for optical flow tracking, the OpenCV implementation of *goodFeaturesToTrack* was used which is based on the Shi-Tomasi corner detector algorithm[3]. Retrieved feature points consist of 2D pixel locations that are matched in consecutive frames and afterwards are tracked using the OpenCV Lucas-Kanade Optical Flow implementation².

The second step is getting the trajectory for x, y and the angle at each frame by accumulating the frame-to-frame transformations and the third step consists of smoothing the trajectory by using a sliding average window.

The final step consists of creating a new transformation and applying it to the current frame from the video.

As seen in Figure 3, the developed stabilization method sets out to perform as intended as it compensates undesirable motion when a strong oscillation occurs. The first picture represents the original frame and the second one the stabilized frame.



Fig. 3. Result of the Stabilization method when a strong oscillation occurs.

b) Filtering

After video stabilization and before releasing the frame to the other modules, some relevant image operation needs to take place. In order to reduce some noise in each frame, some image smoothing (blurring) was applied. This was done by applying the *opening* morphology operation and a Gaussian filter to remove noise and unimportant edges.

² http://docs.opencv.org/modules/video/doc/motion_analysis_and_object_tracking.html

3.4 Vanishing Point Detection

Any set of parallel lines on the plane define a vanishing point and when traversing a corridor or a hallway, one can easily observe four lines drawn to the ends of that corridor or hall. Therefore, an image from a corridor may have more than one vanishing point. Following the vanishing point will enable the drone to avoid collisions with the walls.

The process of detecting the vanishing point in a corridor consists of the following steps:

- **Step 1:** Detecting the lines in the frame using the Hough Transform[5].
- **Step 2:** Filtering out horizontal and vertical lines: only lines which respect the condition $170^\circ < \theta < 10^\circ \cap 180^\circ < \theta < 100^\circ$ are diagonal;
- **Step 3:** Calculating the intersection of all the diagonal lines.
- **Step 4:** Obtaining the vanishing point which is the point with the highest number of intersections.
- **Step 5:** Removing outliers and using a Kalman filter to maintain the vanishing point stable throughout the whole video stream.

Figure 4 shows the result of this module. The drone will adjust its yaw to the left or to the right depending on the location of the vanishing point.

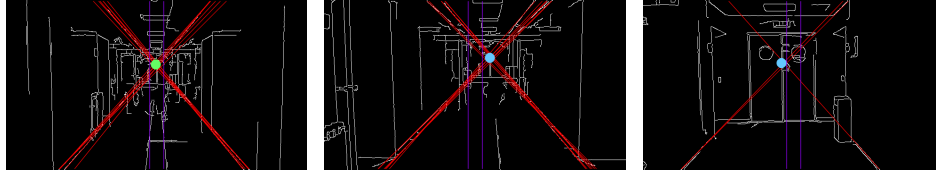


Fig. 4. Result of the vanishing point detection method.

3.5 Door Detection

As a means to achieve localization estimation, the approach taken was that of detecting structural elements of the corridors (like doors) in which the drone would be navigating. A single detection method was perceived as not being enough due to the reason that while traversing the corridor, the front facing camera of the drone does not see the doors, but instead sees recessed doors.

a) Recessed Door Detection

This method aims at detecting the recesses in the corridor which would mean that a door was detected. This method was divided into two parts:

a).1 Floor Segmentation using the watershed segmentation algorithm which performs a non-parametric marker-based image segmentation. This is an interactive image segmentation in which different labels are assigned to the known objects in the frame. Since the floor is the foreground and the rest of the frame is background, they are labelled accordingly.

a).2 Vertex Detection which is achieved by applying a mask to isolate an extract the ground in each frame and using the Hough transform[5] to detect horizontal and vertical lines. The vertexes in the far left and far right side of the frame, which correspond to a recess in the left and the right side of the corridor, are then acquired. Each new vertex detection is cross-validated with the previous one and if successive vertex detections occur during a certain period of time, a door is detected. If the coordinates of the newly detected vertex are within a certain range of the previously detect vertex, it is considered a valid door detection.

As seen in Figure 5, which shows the results of the developed method, the vertexes of the doors are successfully detected.



Fig. 5. The various steps of the recessed door detection method.

b) Door Contour Detection

This method aims at detecting the contours of doors in corridors, filtering out the unnecessary contours in order to detect those who correspond to doors. To achieve this, the following sequence of operations is used:

- **Step 1:** Apply a threshold to the current frame in order to obtain a binary image.
- **Step 2:** Detect the edges in the current frame using the Canny edge detector.
- **Step 3:** Retrieve the contours from the binary image using [6].
- **Step 4:** Approximate the contours to a rectangle, which will make the contour sides be a lot more regular, using the Douglas-Peucker algorithm.
- **Step 5:** Segment the rectangular contours using geometry. All the contours whose height is not bigger than its width are discarded.

Resembling the recessed door detection method, this method also uses temporal information in order to avoid unreliable detections. This is achieved by simply keeping a counter for both left and right door contour detections. When

that counter reaches a specific value, a door is detected and the corresponding counter is reset.

In the end, the contours of the doors in the video feed are detected properly as seen in figure 6.

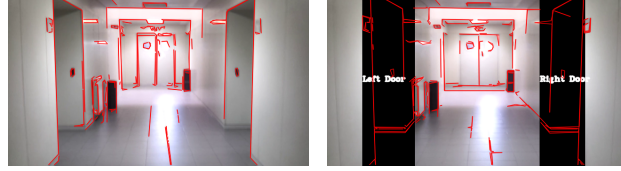


Fig. 6. Result of the door contour detection method.

3.6 Position Estimation

The final module of the developed framework is responsible for the position estimation of the vehicle traversing the corridor. In order to estimate this information, the following data is used:

- **The velocity of the drone:** either a real-time value provided by the drone, or, assuming that the vehicle is moving at a steady pace, a constant value.
- **Precoded Map Information:** a text file containing relevant metric data from the corridor in which the drone will be traversing such as the distance from the starting point of each door. Before each flight the framework loads the corresponding floor map data containing the information shown in Table 1 (all the measurements are in centimetres)
- **The door detection information:** obtainable from the door detection module.

Table 1. Example of corridor data: details, door numbers (I00X) and distances.

storey	length	start	rooms	left	right	left	right	left	right
0	2300	north	6	I006 500	I011 500	I007 1700	I010 1700	I008 2300	I009 2300

After reading the floor information provided by the data file, a visual representation of the floor map is created. The map itself never changes, however, two types of information are updated/drawn on the map:

- **The position of the drone:** updated depending on its current velocity and the initial point of departure.
- **The location of the detected doors:** drawn on the map when the current position of the drone is near the location of a door and the door detection module recognizes it.

The result of the this module, which uses all the previously developed modules, can be seen in Figure 7.

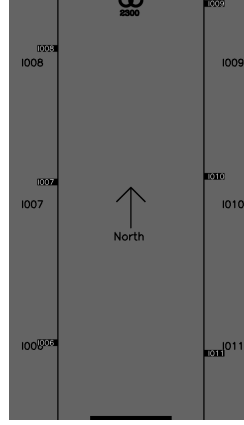


Fig. 7. Example of a successful door detection and position estimation.

4 Experiments

After testing simple take-off, hover and navigation commands on the Parrot AR. Drone, it was noticeable that it could not maintain a stable trajectory on narrower corridors (width lower than 1,5 meters). The reason for the loss of stability was due to different airflows:

- Originating from the various rooms behind the doors in the corridor.
- Caused by the rotors of the drone. Since the corridor is so narrow, the air-flow caused by the drone does not disperse uniformly causing the drone to destabilize.

Therefore, the narrower corridor was used to test the behaviour of the developed framework with a previously recorded video instead of a live video from the Parrot AR. Drone.

The corridor used for the flight tests with the drone has a length of 23 meters and 6 rooms: 3 on the left and 3 on the right (all with recessed doors).

Two different trajectories were used as test scenarios:

- **Experiment #1:** Northern entrance as the starting point.
- **Experiment #2:** Southern entrance as the starting point.

The narrow corridor that was used has a length of 25 meters and 12 rooms: 8 on the left and 4 on the right (only the latter were recessed doors).

All the experiments were conducted in optimal conditions, meaning that:

- There were no people walking along the corridor.
- All the doors were completely shut in order to reduce airflow interference.

Figure 8 shows the typical output of the framework during an experiment.

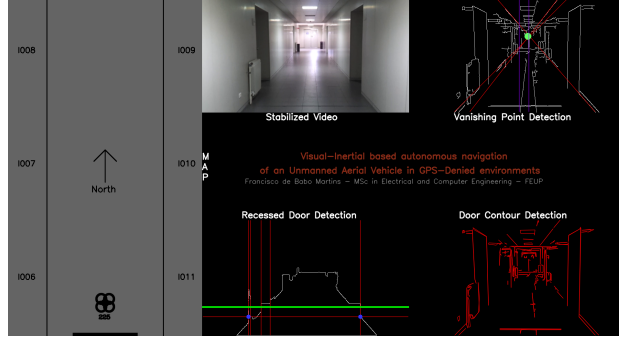


Fig. 8. Example of the control window provided by the framework.

The developed modules proved to work accordingly and as desired. Table 2 provides a summary of the 3 experiments conducted.

Table 2. Summary of the experiments.

	Experiment #1	Experiment #2	Experiment #3
Vanishing Point Standard Deviation	[11.47 ; 10.88]	[25.23 ; 10.85]	[18.60 ; 16.91]
Door Detection Success Rate	83.33%	75%	53.33%
Number of Doors	6	4	12
Arrival Success Rate	80%	80%	N/A

The majority of the doors were successfully detected in all the experiments. Each experiment had 10 trials, the average frame rate was around 26 frames per second, and the mean time required by the drone to traverse the corridors was 1 minute and 20 seconds (for experiments #1 and #2). However, the door contour detection proved to be not so robust, leading to a door detection that was almost totally dependent on the recess detection module. The *arrival success rate* corresponds to the number of trials in which the drone was able to travel from one end of the corridor to the other without colliding with the walls. Since experiment #3 was conducted without a flying drone, the *arrival success rate* is not relevant.

The experiment with the best overall results was the first one due to its lower *vanishing point standard deviation*, and higher *door detection success rate*. This results demonstrate that the drone was able to keep aligned with the center of the corridor without colliding with the walls, reach the end of such corridor and

detect most of the its doors. Some simulations and flight experiments can be seen on youtube: <https://youtu.be/xjIpChc90io>

5 Conclusions

This project proved that it is possible to enable an UAV to possess a system capable of providing a collision-free navigation, a door detection system and a robust position estimation of both a vehicle and doors in corridors.

Using only the vanishing point, the drone was able to fly with a wall collision-free navigation and doors with different formats were successfully identified thanks to the use of two different detection methods: *recessed door* and *door contour detection*. Also the position of the drone and the position of the doors in the floor map were estimated with minimal errors and were quite satisfactory.

The developed framework is constructed in such a way that the inclusion of additional modules would be a simple and accessible task. Since each module is independent of one another, sharing just one video feed among each other, it would be as simple as switching a specific module on or off. This feature enables the framework to have a considerable degree of flexibility as it is not bound to a specific type of robot, being versatile to the point of working properly on either aerial or ground vehicles. A ground vehicle would provide a higher stability and lower external interferences like airflows.

Concluding, the proposed method has proven to be flexible and versatile enough in order to be considered a positive asset in applications such as: autopilot systems; mobile surveillance; traffic management; domestic, industrial and military applications; and search and rescue missions.

5.1 Future Work

By refining the door and vanishing point detection modules, the door detection success rate would increase for corridors without recessed doors and the vanishing point detection would not be susceptible to errors in corridors with poster and boards hanged on the walls. Also, using a drone with a faster data exchange rate would speed up the video input stream processing and therefore make and provide quicker navigation decisions.

Furthermore, adding a collision avoidance and person detector module would provide a safer navigation for coexisting drones and people. Furthermore, processing frames and performing image operations with the GPU instead of the CPU would lead to an improved performance.

Using additional hardware (external or internal to the drone), such as range-finders, would provide additional data regarding possible obstacles such as pillars, windows and foreign objects.

References

1. Ferreira, Filipe Oliveira Ramos Trocado. "Video Analysis in Indoor Soccer with a Quadcopter." Master Thesis, Faculdade de Engenharia da Universidade do Porto. (2014).
2. Coughlan, James M., and Alan L. Yuille. "The Manhattan world assumption: Regularities in scene statistics which enable Bayesian inference." In NIPS, pp. 845-851. (2000).
3. Stavens, David. "The OpenCV library: computing optical flow." (2007).
4. Van Den Heuvel, Frank A. "Vanishing point detection for architectural photogrammetry." *International archives of photogrammetry and remote sensing* 32 (1998): 652-659.
5. Duda, Richard O., and Peter E. Hart. "Use of the Hough transformation to detect lines and curves in pictures." *Communications of the ACM* 15, no. 1 (1972): 11-15.
6. Suzuki, Satoshi. "Topological structural analysis of digitized binary images by border following." *Computer Vision, Graphics, and Image Processing* 30, no. 1 (1985): 32-46.
7. Welch, Greg, and Gary Bishop. "An introduction to the kalman filter. 2006." University of North Carolina: Chapel Hill, North Carolina, US (2006).
8. Lioulemes, Alexandros, Georgios Galatas, Vangelis Metsis, Gian Luca Mariottini, and Fillia Makedon. "Safety challenges in using AR. Drone to collaborate with humans in indoor environments." In *Proceedings of the 7th International Conference on Pervasive Technologies Related to Assistive Environments*, p. 33. ACM, (2014).
9. Bills, Cooper, Joyce Chen, and Ashutosh Saxena. "Autonomous MAV flight in indoor environments using single image perspective cues." In *Robotics and automation (ICRA)*, (2011) IEEE international conference on, pp. 5776-5783. IEEE, 2011.
10. Saska, Martin, Tomáš Krajník, Jan Faigl, Vojtěch Vonásek, and Libor Přeučil. "Low cost mav platform ar-drone in experimental verifications of methods for vision based autonomous navigation." In *Intelligent Robots and Systems (IROS)*, 2012 IEEE/RSJ International Conference on, pp. 4808-4809. IEEE, (2012).
11. Engel, Jakob, Jürgen Sturm, and Daniel Cremers. "Camera-based navigation of a low-cost quadrocopter." In *Intelligent Robots and Systems (IROS)*, 2012 IEEE/RSJ International Conference on, pp. 2815-2821. IEEE, (2012).
12. Bills, Cooper, Joyce Chen, and Ashutosh Saxena. "Autonomous MAV flight in indoor environments using single image perspective cues." In *Robotics and automation (ICRA)*, 2011 IEEE international conference on, pp. 5776-5783. IEEE, (2011).
13. Mori, Takayoshi, and Stefan Scherer. "First results in detecting and avoiding frontal obstacles from a monocular camera for micro unmanned aerial vehicles." In *Robotics and Automation (ICRA)*, 2013 IEEE International Conference on, pp. 1750-1757. IEEE, (2013).
14. Galatas, Georgios, Christopher McMurrough, Gian Luca Mariottini, and Fillia Makedon. "eyeDog: an assistive-guide robot for the visually impaired." In *Proceedings of the 4th International Conference on Pervasive Technologies Related to Assistive Environments*, p. 58. ACM, (2011).
15. Achtelik, Markus, et al. "Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments." *SPIE Defense, Security, and Sensing. International Society for Optics and Photonics*, (2009).
16. Huang, Albert S., et al. "Visual odometry and mapping for autonomous flight using an RGB-D camera." *International Symposium on Robotics Research (ISRR)*. (2011).