Autonomous Landing of AR.Drone

Roman Barták„ Andrej Hraško„ David Obdržálek

Autonomous Landing of AR.Drone

Roman Barták, Andrej Hraško, and David Obdržálek

Charles University in Prague, Faculty of Mathematics and Physics, Praha, Czech Republic
{roman.bartak, david.obdrzalek}@mff.cuni.cz
andrej@hrasko.eu

Abstract. Inexpensive robotic toys are becoming widely available and though they are called toys, they provide gradually better sensing features and rising computational power. Adding new autonomous functions, such as autonomous landing for flying drones, helps the users with routine maneuvers and so allows using them with lower risk of breaks caused by loss of control in critical operational phases. In this paper we present software for autonomous landing of an AR.Drone – an affordable robotic quadricopter. This software uses an easy-to-customize landing pattern and exploits only the sensors available on the drone. The whole landing process is simplified to a "push-button" approach for the end user, allowing for safer overall operation.

1 Introduction

AR.Drone by Parrot Inc. is a high-tech yet affordable flying toy – a quadricopter with several sensors including two cameras, accelerometer, gyroscope, sonar, and a built-in controller for basic stabilization. Pre-processed information from sensors can also be received via Wi-Fi so the computer can serve as a more powerful external brain for the drone. For controlling the drone from a computer and for visualizing information from sensors, several software applications exist. We use *Control Tower* [6] as the basic computer interface to AR.Drone and we extended its functionally with autonomous landing to a user-customizable pattern. The idea is that the user flies the drone above the pattern and by pushing a button in the software GUI the drone autonomously lands on that pattern. Such functionality is great for beginner users who can control the drone in free space but are not able to precisely land or are afraid of controlling it low above the ground where mistakes in control or turbulences caused by airflow could easily cause crashing and then damaging of the drone.

2 Landing description

For autonomous landing, the drone must reliably detect the landing point. There exist methods for visual navigation (e.g. using an H-shaped landing pattern [5] or a specific circular pattern [3]), however they require better visual sensors (cameras) and they usually do not support horizontal orientation.

We decided for the landing pattern consisting of two colored discs which makes it very easy to be prepared by an inexperienced user. The user can print any two colored discs; it is only important that the colors are different enough from each other and from the typical colors in the landing zone. This makes it much easier to use in real life (instead of being forced to use specially crafted and unalterable unique pattern). In addition to this easy customization of the landing pattern, the other advantage is that such pattern also defines the horizontal orientation.

To preprocess the picture, we straightforwardly use HSV filtering, image erosion, and image dilatation (these basic image pre-processing algorithms are readily available and described in the OpenCV library [7]). Then, to identify each disc in the picture, blob detection algorithm (described in [4]) is used and the center of the disc pair is calculated to set the target landing point.

The drone is controlled using four independent standard *proportional-integral-derivative (PID) controllers* [2] fed by the distance to the target. The input for each PID controller is the coordinate part of the target distance in its specific direction and the output is one control value for the AD.Drone low-level interface functions (the pitch, roll, yaw, and vertical speed). The distance calculation is shown on Figure 1.

More details about the algorithms and their implementation can be found in [1].

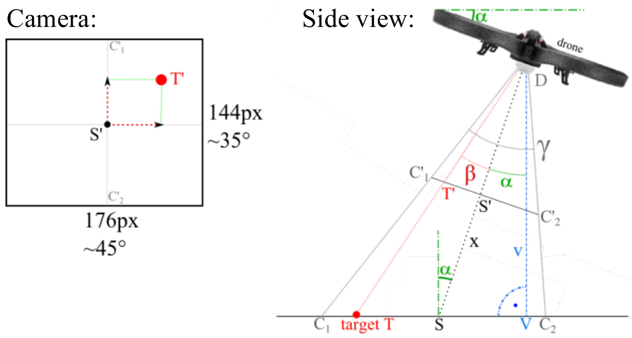

Fig. 1. Distance to the target calculation: $dist(T,V) = v * \tan(\alpha+\beta)$, where $\beta = \arctan(\tan(\gamma/2) * pixel\_dist(T',S') / pixel\_dist(C'_1,S'))$

The typical process to use autonomous landing is as follows. The user sets the HSV parameters of the two color blobs in the picture and optionally selects the horizontal angle (yaw) of the drone with respect to these two blobs. Any time later the user can invoke the "*land on target*" action to let the drone automatically land or the "*lock on target*" action to hover above the landing point at the current altitude. When the landing pattern disappears from the camera view, the system can optionally attempt to find it by flying in the direction where the pattern was lost ("*estimate target*").

3 Evaluation and Conclusion

During the tests it has proven the AR.Drone is able to land safely even when the environment conditions are too advantageous for manual control (e.g. air flow at ground level). Figure 2 presents the graphs showing the landing time, precision, and reliability of the landing from different original altitudes and orientations with or without the wind generated by two fans close to ground at the landing area.

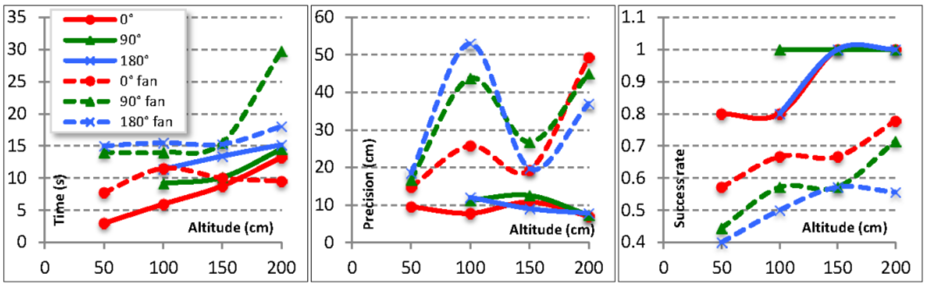

Fig. 2. Dependence of landing time (top left), precision (top right), and reliability (bottom) on the initial altitude (50, 100, 150, 200 cm) and orientation relative to the target (0°, 90°, 180°).

In this work we have shown how standard control software of a low-cost quadricopter AR.Drone can be extended by an automatic landing procedure. That lowers the bad chance to break the drone during landing which belongs to the most dangerous phases of the flight; especially the windy conditions are difficult for manual landing and often cause crashes and damages. This is important for users who exploit the drone as a toy, for playing and entertainment. Our preliminary experimental results show this extension is reliable under normal, non-laboratory conditions.

Acknowledgements. This research was supported by the Czech Science Foundation projects P103/10/1287 and P202/12/G061.

References

1. R. Barták, A. Hraško, D. Obdržálek, On Autonomous Landing of AR.Drone: Hands-on Experience. Proceedings of FLAIRS 28 (2014)
2. M. King. Process Control: A Practical Approach. Chichester, UK: John Wiley & Sons Ltd. (2010)
3. S. Lange, N. Sünderhauf, P. Protzel, A Vision Based Onboard Approach for Landing and Position Control of an Autonomous Multirotor UAV in GPS-Denied Environments. International Conference on Advanced Robotics, pp. 1-6 (2009)
4. R. Szeliski. Computer Vision: Algorithms and Applications. Springer, Heidelberg (2010)
5. S. Yang, S.A. Scherer, A. Zell, An Onboard Monocular Vision System for Autonomous Takeoff, Hovering and Landing of a Micro Aerial Vehicle. Journal of Intelligent & Robotic Systems, Volume 69, Issue 1-4, pp. 499-515 (2013)
6. ControlTower software. Accessed May 14, 2014. https://code.google.com/p/javadrone/wiki/ControlTower
7. OpenCV software library. Accessed May 14, 2014. http://opencv.org/