Master's Thesis Engineering Technology

2022-2023

Autonomous navigation of a Tello Ryze drone based on AprilTags

Tibo Poncelet

Master of Electronics and ICT Engineering Technology

Introduction

Autonomously navigating drones are becoming increasingly more common in applications such as infrastructure inspection, agriculture, and package delivery [1]. They can replace manual labor and improve efficiency of tasks in certain cases [1].

Problem statement and goals

The goal of this thesis is to develop **software** that enables a Tello drone to navigate autonomously based on **visual input**. In order to achieve this, software that can control and receive sensory data from a Tello Ryze drone must be developed.

The drone must be able to autonomously navigate a predetermined path based on artificial **landmarks** spread across the path. It is important that the delay between input and reaction of the drone is as low as possible. In addition, the software must be designed in a way that facilitates swapping and integrating new navigational algorithms.

Method and materials

The Drone

The drone used for this application is the **Tello Ryze drone** developed by Ryze Tech. This is a budget friendly drone costing less than €100 with limited functionality and a low-resolution camera. Figure 1 depicts this drone.

Figure 1: The Tello Ryze drone

The camera of the drone can capture images of 5 megapixels and stream videos at 720p [2]. The lithium polymer battery of the Tello Ryze allows for approximately 13 min of flight time [2]. The maximum flight attitude angle is 25° and it can achieve a maximum speed of 28,8 km/h [2].

AprilTags

An AprilTag is an **artificial landmark** designed to be easily detected by cameras used in robotic applications which usually produce lower quality images than common cameras [3]. The tags are similar to QR-codes but the difference is in their payload. QR-codes contain larger amounts of data but require higher resolution cameras to be scanned [3]. AprilTags have a smaller payload but are easier to detect with lower resolution cameras [3]. An example of an AprilTag is given in figure 2.

Figure 2: An AprilTag [3]

The Software

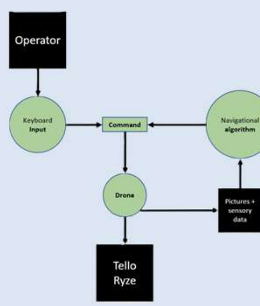

Figure 3: A diagram of the software structure

The software is developed using: Python, C++ and the **Robotic Operating System** (ROS). This system uses individual **nodes** which can be interchanged. This allows developers to easily swap and test new navigational algorithms. Input is processed by the input node and converted to drone commands. The algorithm node generates commands in the same format. These commands are then published to the command topic where they are accessed by the drone node and executed. This node can also save timestamped pictures and other sensory data to their corresponding folders. The algorithm node can access these folders and use their data to compute the next command. Figure 3 depicts a diagram of this process.

The Evaluation

Autonomous navigation based on AprilTags is tested by instructing the drone to follow a predetermined path marked by AprilTags. These tags contain instructions for the drone on how to navigate to the next tag. The software was validated through a series of tests to ensure its reliability and effectiveness in varying lighting conditions and environments. Figure 4 depicts the system detecting the edges and center of an AprilTag.

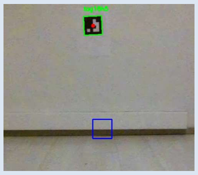

Figure 4: Detection of an AprilTag

Results and conclusion

Figure 5 depicts the detected average size of the edges of the AprilTag in function of the distance between the drone and tag. The measurements start at 30 cm, the size drops significantly until 250 cm. After that point it stagnates. This makes it difficult to estimate the distance to the tag.

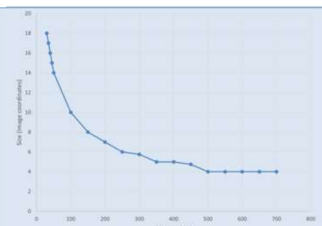

Figure 5: Tag size in image relative to distance from tag to camera

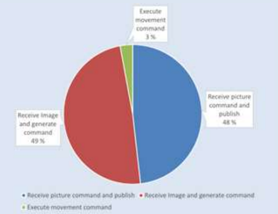

Figure 6: Pie chart depicting time usage during the algorithm

Figure 6 depicts the time necessary for the system to take a picture, detect tags, generate an instruction and execute that instruction. The total time required to do this is **80,49 ms**. 48% of that time is spent taking, encoding and publishing the picture. 49% is spent decoding and generating an instruction. The remaining 3% is used to execute the instruction

Supervisors / Co-supervisors / Advisors: Prof. dr. Nick MICHIELS
Prof. dr. Joni-Kristian Kämäräinen
Lauri Suomela

[1] Y. Yasuda, L. Martins, and F. Cappabianco, "Autonomous visual navigation for mobile robots: A systematic literature review," ACM computing surveys, vol. 53, no. 1, pp. 1–34, 2020

[2] Ryze Tech, "Tello User Manual v1.4". [Online]. Available: https://dl-cdn.ryzerobotics.com/downloads/Tello/Tello%20User%20Manual%20v1.4.pdf. [Accessed: 8 May2023].

[3] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in 2011 IEEE International Conference on Robotics and Automation, pp. 3400–3407, IEEE, 2011.

Tampere University

De opleiding industrieel ingenieur is een gezamenlijke opleiding van UHasselt en KU Leuven

UHASSELT

KU LEUVEN