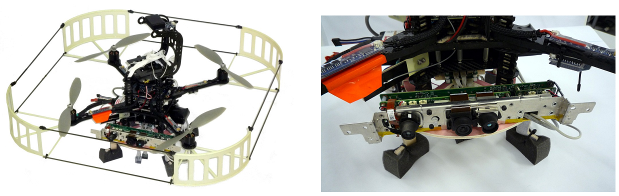
Autonomous Corridor Flight of a UAV Using an RGB-D Camera

Sven Lange Niko Sünderhauf Peer Neubert Sebastian Drews Peter Protzel



(a) (b)

Fig. 1. (a) Our modified *Pelican* quadrotor system. (b) The dismantled Kinect sensor mounted on the UAV.

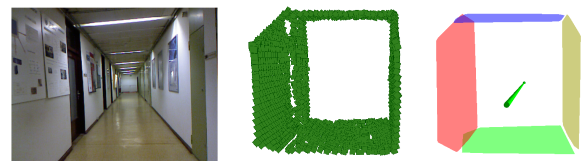


Fig. 2. *(left)* RGB image of the corridor. *(mid)* Downsampled point cloud containing about 3,000 points. *(right)* Extracted and classified planes of the walls. The green arrow shows the intended motion direction computed from the position and orientation relative to the walls.

I. Introduction

We describe the first application of the novel Kinect RGB-D sensor on a fully autonomous quadrotor UAV. We apply the UAV in an indoor corridor scenario. The position and orientation of the UAV inside the corridor is extracted from the RGB-D data. Subsequent controllers for altitude, position, velocity, and heading enable the UAV to autonomously operate in this indoor environment.

II. System Overview

The UAV we use in our project is a "Pelican" system (see Fig. 1(a)) that is manufactured by Ascending Technologies. We extended the UAV's configuration and equipped the quadrocopter with additional hardware: An SRF10 sonar sensor measures the altitude, an ADNS-3080 optical flow sensor board provides information on the current velocity. Altitude, velocity and position are controlled using cascaded PID controllers that are implemented on an ATmega644P. The Kinect RGB-D device (Fig. 1(b)) is connected to the onboard embedded PC system and interfaced using ROS.

III. Autonomous Corridor Flight Using the Kinect RGB-D Device

The Kinect driver of ROS provides a 640×480 3D point cloud that is downsampled (thinned) to approximately 3,000 points for further processing. For performance reasons, we implemented a specialized downsampling algorithm that runs five times faster compared to the function provided by the Point Cloud Library.

After downsampling, large planar sections are found in the remaining points by applying a sample consensus (MLESAC) based parameter estimation algorithm. Fig. 2 visualizes the results. The Point Cloud Library already provides convenient algorithms that extract the planes in their parameter form $ax + by + cz + d = 0$.

The authors are with the Department of Electrical Engineering and Information Technology, Chemnitz University of Technology, 09111 Chemnitz, Germany. `firstname.lastname@etit.tu-chemnitz.de`

Given these parameters, the extracted planes are assigned to one of the following wall classes: *floor, ceiling, left, right, front*. Distances from the UAV to the walls Δ_i and their orientations ϕ_i are calculated as well.

To keep the UAV aligned with the corridor and in the center of the corridor, motion commands $(dx, dy, d\phi)$ are calculated from the plane distances Δ_i and the yaw estimates ϕ_i. Weight factors w_i are used to ensure that those walls that were supported by more scan points during the plane extraction have a stronger influence on the resulting motion command.

IV. Results

A video that shows an example flight is available at our website www.tu-chemnitz.de/etit/proaut/forschung/quadrocopter.html.en. Fig. 3 shows the position estimates Δ_{left} and Δ_{right} inside the corridor while performing autonomous flight. According to these internal measurements, the maximum deviation from the corridor center was 35 cm. Mean velocity was $0.36 \frac{m}{s}$. The sourcecode for efficient point cloud downsampling and trajectory generation is available to the community as part of our ROS repository at http://www.ros.org/wiki/tuc-ros-pkg.

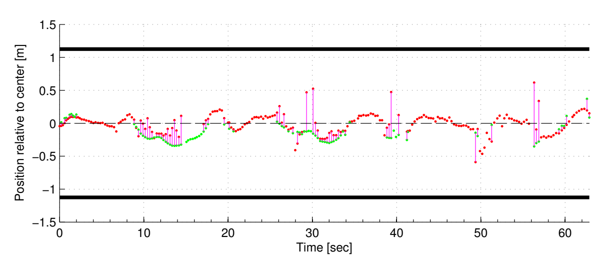


Fig. 3. Position estimates of the quadrotor within the corridor during autonomous flight, based on the Kinect measurements. The red points represent the calculated distances based on the left wall and the green points are based on the right wall. If two walls are visible, measurements are connected through a line. The corridor walls are shown as thick black horizontal lines. Maximum deviation from the corridor center was 35 cm.