Software Requirements Specification
for
Warehouse Drone Collision Avoidance Project

Prepared by Tristan van Vegchel
Seacon Logistics
Approved from version 1.0 onwards

Venlo, created on November 11, 2019

Contents

List of Figures

Revision History

Revision	Date	Author(s)	Description
0.1	11-11-2019	Tristan van Vegchel	Created skeleton
0.1	12-11-2019	Tristan van Vegchel	Created shortened version
0.2	27-11-2019	Tristan van Vegchel	Added introduction
0.3	06-12-2019	Tristan van Vegchel	Added overall description
0.4	23-12-2019	Tristan van Vegchel	Added functional requirements
0.5	01-01-2020	Tristan van Vegchel	Completed first draft
1.0	10-01-2020	Tristan van Vegchel	Document reviewed and approved

List of Abbreviations

A.I. Artificial Intelligence.

Fontys Fontys University of Applied Sciences.

STG2 Graduation Project.

Definitions

Action task A task that is directly related to the movement of the drone.

Customer The person, or persons, who pay for the product and usually (but not necessarily) decide the requirements. In the context of this recommended practice the customer and the supplier may be members of the same organization.

Layer A rack in a warehouse allows for multiple pallets to be stacked on top of each other, separated by horizontal beams. A layer consists out of a single beam with the items placed directly on top of it.

Maintenance task A supportive task that allows the drone to be able to perform its original tasks without interference. An example being readjusting the distance from the beam caused by drift.

Might Features that might perform a certain task imply that said feature is generally desired by the customer, but is not listed as a requirement.

Opencv A computer vision library originally written in C.

Product The product is the name used to refer to the product of this iteration of the project, or the product of the warehouse drone collision avoidance project.

Seacon Drone Project The Seacon Drone Project is the name used to refer to the entire project to semi-automate inventory control processes using drones. As of writing this document this includes functionality for self-navigating through the warehouse, collision avoidance, performing cycle counts, and checking specific locations.

Shall Features that shall perform a certain task imply that said feature is both required and essential for the completion of the product. These features are thus considered high-priority features.

Should Features that should perform a certain task imply that said feature is required by the customer, but is given lower priority and is not considered essential.

Status task A task concerned with (partially) retrieving the status of the drone and/or its environment. An example being tracking the vertical center of the beam.

Supplier The person, or persons, who produce a product for a customer. In the context of this recommended practice, the customer and the supplier may be members of the same organization.

Chapter 1

Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description of the software product of the Drone Warehouse Collision Avoidance project. It contains the purpose and features of the software, the interfaces, and the constraints under which it must operate. This document is primarily intended for the department of Engineering & IT of Seacon Logistics, as this department will contain/contains the future developers of the next iteration of this product. As this project is developed as a bachelor thesis project, the by Fontys University of Applied Sciences appointed tutors, lecturers, and experts are also considered an intended audience.

1.2 Document Conventions & Standards

This document was created based on the IEEE Recommended Practice for Software Requirement Specifications standard 830-1998.

1.3 Scope

The Warehouse Drone Collision Avoidance system (which is in this document generally referred to as "product") will primarily be responsible for providing a drone with a prototype solution to move along all layers of a single side of a rack in a warehouse, while avoiding collisions.

1.4 Overview

The remainder of this document will feature an overall description of the product and the specific requirements. The former includes the perspective, information regarding a set of used interfaces, the product functions, and information regarding the product its users. The latter contains the specific requirements listed with unique identifiers, and a list of assumptions made in order for the product to function.

Chapter 2

Overall Description

2.1 Product Perspective

The product will be developed as a first step of a larger project, which ultimately has the goal to semi-automate the cycle count processes of the inventory control & management department of the customer's warehouses. The concept was initially developed in order to combat the increasing scarceness of warehouse cycle count employees. An overview of the larger project can be found in figure 2.1.

System Interfaces As the product for which this document is written will be the first component of the Seacon Drone project, the interfaces of the Seacon Drone project which the product will communicate with have yet to be defined.

Hardware Interfaces The product will support Dji Tello drones (Ryze Robotics, 2018). It will developed to be used with a Wi-Fi enabled laptop. The product will provide a set of drone control command keys, which will only work as intended on an QWERTY or AZERTY keyboard layout.

Software Interfaces The product will be developed in Ubuntu 18.04. This will thus be the only operating system and version which will have guaranteed support. Furthermore, the device running the product is required to have a python interpreter version 3.6.x with pip, as this will be the language and package management system for development, respectively. The python-pip packages the software will make use of are:

- opencv-python (version 4.1.1.26) (OpenCV Team, 2019)
- av (version 6.2.0) (Boers, 2019)
- tellopy (version 0.6.0) (Hanyazou, 2018)
- simple_pid (version 0.2.4) (Lundberg, 2019)
- pynput (version 1.4.5) (moses palmer, 2019)

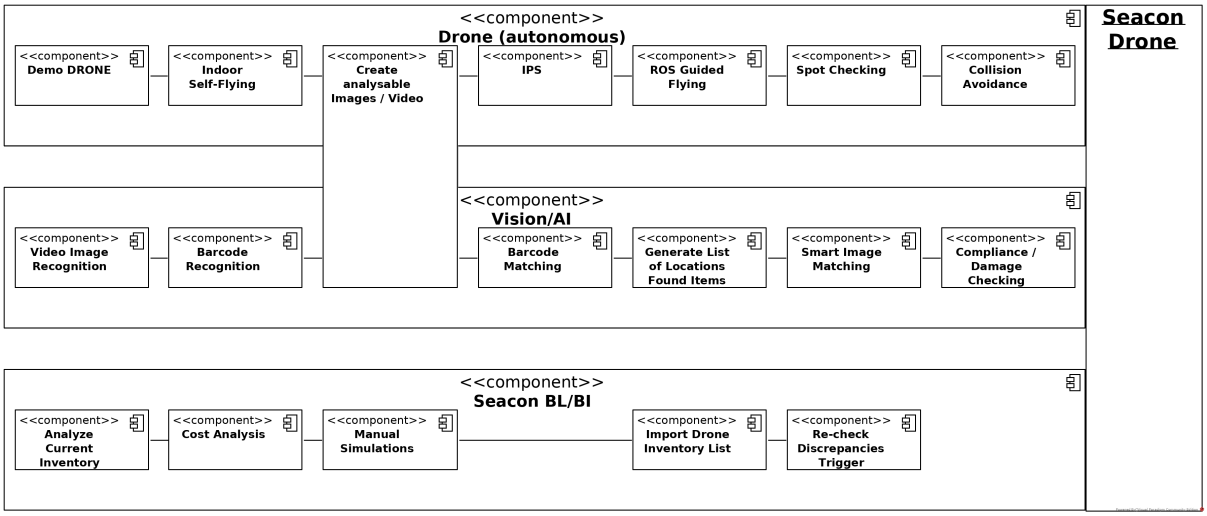

Figure 2.1: Roadmap of the Seacon Drone Project

2.2 Product Functions

The product can be divided into 3 parts: x-axis, y-axis, z-axis of the drone. Each axis can then be subdivided further into a video processing part, drone movement part, and a collision avoidance part. Figure 2.2 visualizes those 3 axes.

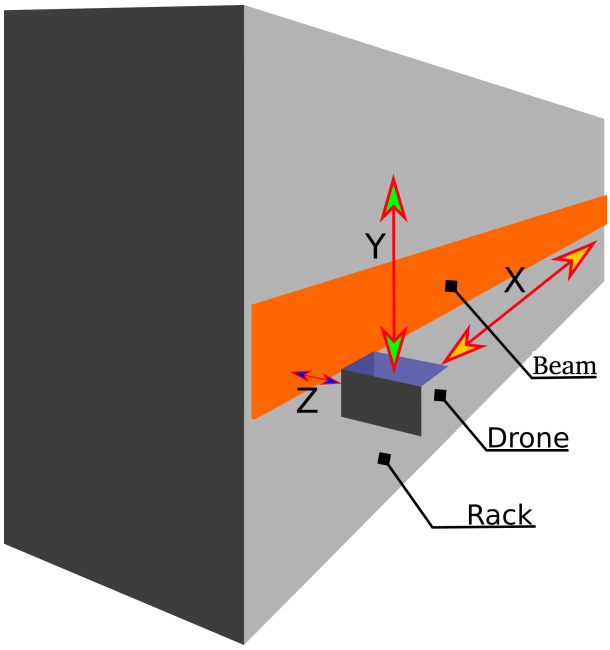

Figure 2.2: Simple drawing visualizing the intended product.

- X-axis:
  - Move the drone sideways
  - Stop the drone at the end of a bar
  - Detect objects in the drone's trajectory
  - Avoid collisions with objects in the drone's trajectory
- Y-axis:
  - Keep the bar in the center of the drone's camera
  - Change the drone's altitude to switch layers
  - Keep track of the current layer and the highest layer
- Z-axis:
  - Estimate the pixel size of the bar
  - Keep the drone to maintain distance from the bar

When referring to "objects", one of the following things is meant:

- Pallets
- Racks
- Forklifts
- People
- Drones

2.3 User Classes and Characteristics

As the product of this iteration will not yet include inventory control functionality, the typical users do currently not include a warehouse employee class. Thus the sole user class of this product will be developers who aim to improve and/or extend the current product. Developers are required to have an extensive knowledge of Python and Opencv.

Chapter 3

Specific Requirements

3.1 Functional Requirements

The following section describes the functions the product shall, should, and might include. For the definitions of these 3 terms refer to the glossary. Each of these 3 terms form a section, which are then further segmented into status, maintenance, and action tasks, for which the definitions can also be found in the glossary. Each functional requirement will have the following format:

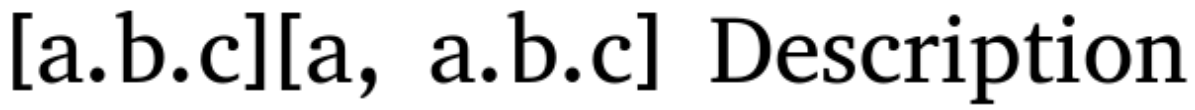

Figure 3.1: Format for a functional requirement.

Whereas:

- **a** is the priority group (shall, should, might)
- **b** is the type of task
- **c** is the unique number of that requirement within its group.

Each requirement is prefixed by at least one set of brackets containing its unique ID, but may have a second set containing one or more IDs of prerequisite tasks or groups, each separated by comma. Note that in figure 3.1 the prerequisites bracket contains a single **"a"**, which indicates that all requirements in that priority group are a prerequisite to that task.

[1] The product shall make the drone:

- [**1.1**] Status tasks:
  - [**1.1.1**] Track the vertical center of the beam
  - [**1.1.2**] Estimate the distance from the beam
  - [**1.1.3**] Distinguish objects in the foreground from the background
  - [**1.1.4**][1.1.3, 1.2.1, 1.2.2] Check for objects in its trajectory

- [**1.2**] Maintenance tasks:
  - [**1.2.1**][1.1.1] Maintain its altitude to keep the beam vertically centered
  - [**1.2.2**][1.1.2] Maintain a distance between 30 - 80 centimeters from the beam
- [**1.3**] Action tasks:
  - [**1.3.1**][1.2] Move along the beam from left to right
  - [**1.3.2**][1.1.4] Avoid collisions by landing

[2] The product should make the drone:

- [**2.1**] Status tasks:
  - [**2.1.1**] Keep track of the remaining layers needing to be scanned
  - [**2.1.2**] Determine the end of the beam
- [**2.3**] Action tasks:
  - [**2.3.1**][1.1.2, 2.1] Move to other layers of a rack
  - [**2.3.2**][1.2] Move along the beam from right to left
  - [**2.3.3**][2.1.2] Stop at the end of the beam

[3] The product might make the drone:

- [**3.1**] Status tasks:
  - [**3.1.1**][1.1.4] Determine if an object in its trajectory can be avoided
  - [**3.1.2**] Determine pixel size:distance ratio of the beam
  - [**3.1.3**][1.2.1] Detect bar-codes on the beam
- [**3.2**] Maintenance tasks:
  - [**3.2.1**][3.1.1] Readjust trajectory to avoid collision
- [**3.3**] Action tasks:
  - [**3.3.1**][3.2.1] Move to the opposite rack of the same hallway

3.2 Assumptions

In order for the product to function accordingly, the following assumptions have been made:

- The user is required to fly the drone to the correct starting position. This position is for the drone to face the left end of the lowest beam, at a distance between 30-80 centimeters.
- There are no things sticking out more than 10 centimeters from the rack.
- The drone will have enough battery capacity to complete one side of a rack.
- The warehouse interior lights are turned on and provide enough lighting.
- The drone is equipped with an RGB-camera.
- There is at least 3 meters of free space after the end of a beam.
- The drone remains connected to the device controlling it.

References

Boers, M. (2019), 'Pyav'.
URL: *github.com/mikeboers/PyAV*

Hanyazou (2018), 'Tellopy'.
URL: *github.com/hanyazou/TelloPy*

Lundberg, M. (2019), 'Simple-pid'.
URL: *github.com/m-lundberg/simple-pid*

moses palmer (2019), 'Pynput'.
URL: *github.com/moses-palmer/pynput*

OpenCV Team (2019), 'About opencv'.
URL: *https://opencv.org/about/*

Ryze Robotics (2018), 'Tello user manual'.
URL: *dl-cdn.ryzerobotics.com/downloads/Tello/20180212/Tello+User+Manual+v1.0_EN_2.12.pdf*