Kinect Hand Gesture Drone Flight Controller

Darren Jody van Roodt

Thesis presented in fulfilment
of the requirements for the degree of
Honours of Science
at the University of the Western Cape

Supervisor: Mr Mehrdad Ghaziasgar
Co-supervisor: Mr Reg Dodds

April 2016

Declaration

I, Darren Jody van Roodt, declare that this thesis "*Kinect Hand Gesture Drone Flight Controller*" is my own work, that it has not been submitted before for any degree or assessment at any other university, and that all the sources I have used or quoted have been indicated and acknowledged by means of complete references.

Signature: ..... Date: .....

Darren Jody van Roodt.

Abstract

Controlling a drone using hand gesture is done by capturing the user's hands an arms and recognising the command that is being gestured. This project will attempt to develop a computerized hand gesture control system to control a drone during flight. The system will observe the user's hand gestures through a sequence of frames and convert the recognised gestures into signals to a drone, commanding it to move. The project will regard the hands and arms as the regions of interest. The user will be asked to make certain hand gestures which will be correctly identified by the system.

Key words

HDI

OpenCV

Acknowledgment

I would like to thank my supervisors Mr. M. Ghaziasgar and Mr. R. Dodds for their support and encouragement during my Honours year.

Contents

List of Tables

List of Figures

Glossary

HDI Human Drone Interaction.

OpenCV Open Source Library for computer vision

Chapter 1

Background

This project is an attempt to control a drone with hand gestures. It entails developing an application that uses a Microsoft Kinect, which will be pointed at a user, to recognize hand gestures from the user. This will involve recognizing hand gestures first. Later it can be extended to more complex gestures. The application will then take the recognized hand gestures and converts it to signals to command the drone.

Chapter 2

User Requirements Document

2.1 Introduction

This section describes the problem from the user's point of view. It states the problem domain and the functionality of the program.

2.2 User's view of the problem

Controlling a drone is normally done through controllers, smart phones and tablets. The user wants a drone control system which uses natural movement as a means of controlling a drone.

2.3 Domain of the problem:

Personal drones are becoming popular in everyday environments. Therefore the domain of the problem is Human and Drone Interaction(HDI).

2.4 Description of the problem:

The user requires a system that uses natural movement to control a drone. The system will capture natural gestures and recognise. Through recognising the natural gestures the system can send signal commands to the drone, which will control it.

2.5 What is the solution for the problem?

The system that the user wants is a natural gestures drone controller. The natural gestures which the system will recognise is hand gestures. The solution for the suggested system will use a Microsoft Kinect to capture the hand gestures of the system user. Then system will have to recognise the hand gesture, every hand gesture is a control command for the drone. Therefore after recognising the hand gesture a command signal will be send to the drone to control it. The system will have a predefined hand gestures.

2.6 What is expected from the Software?

The Software is expected to detect the user's hands and arms using a Microsoft Kinect 360. Then the system will recognise hand gestures that the user makes to fly the drone. After recognising the hand gestures, a signal is sent to the drone which will make the drone move according to the gesture. The system will have a user friendly interface.

2.7 What is not expected from the Software?

The software will not track multiple users and it will not monitor users who are not directly facing the Kinect. The system is not expected to turn on the drone, the users need to turn it on using the button.

2.8 Conclusion

The above section describes what the user requires from the system that will allow them to control a drone with hand gestures. It also describes the expectations of the system to define the scope of the project. Chapter 3 provides the developers design of the system based on the users requirements.

Chapter 3

Requirements Analysis Document

3.1 Introduction

In chapter 2, the user specified the need for a hand gesture controlled drone system. This chapter provide the design of the system based on the user requirements of the previous chapter. This will be done by breaking down the problem into high level parts and identifying the main parts to solve the problem.

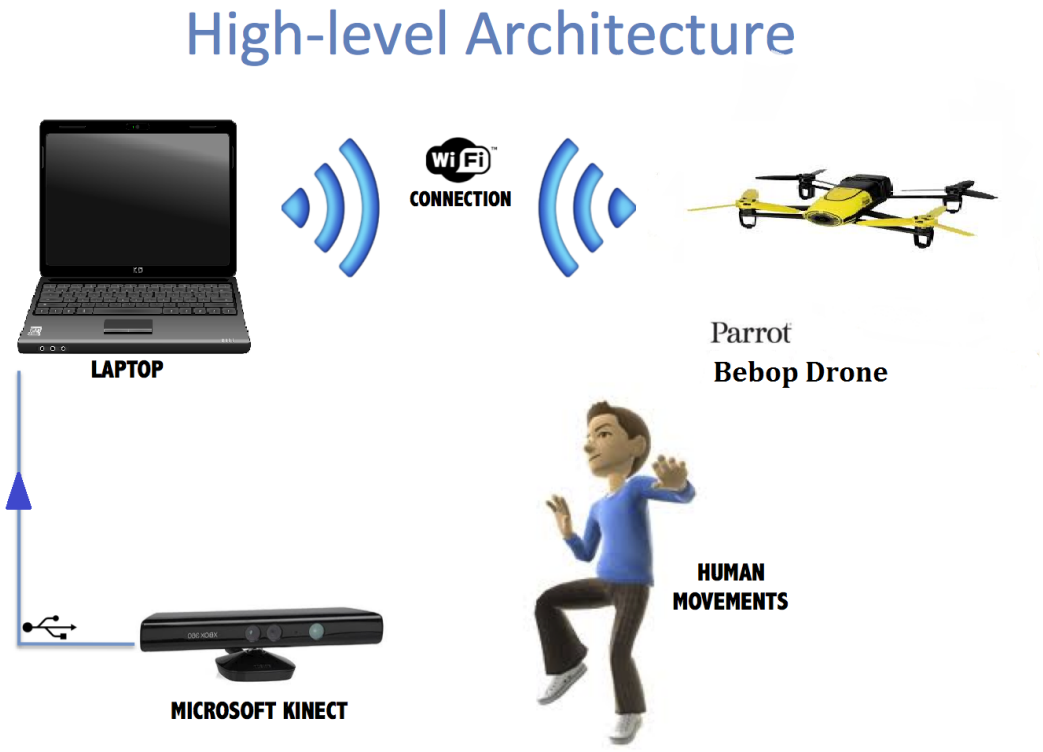

Figure 3.1: High Level Architecture

3.2 High level breakdown of problems

The application captures live videos of the subject, who is going to control the drone. The subject has to stand directly in front of the Kinect camera and facing it. When the subject is directly facing the Kinect then the subject has to perform the hand gestures which have been predefined to control the drone.

Identifying hand gestures need to be accurately done. The video taken when the subject does the gesturing will be have to be track the hand and process the hand gestures. The processing of the video will be done with OpenCV. OpenCV is a computer vision and machine learning software library which provides efficient implementation of many computer vision algorithms (Bradski and Kaehler, 2008). In order to send control commands to the drone the application has to transfer UDP(User Datagram Protocol) packets over WiFi.

3.3 Deep analysis of the problem

3.3.1 Predefined hand gestures

A set of hand gestures is needed to control the drone. In this section a set of predefined hand gestures will be giving to control the drone in a very basic manure.

The hand gesture to make the drone take off and land is showing your palms to the kinect.

The next hand gesture will command the drone to do an emergency landing, the hand gesture is to cross your arm.

In order to make the drone fly forwards and backwards the user needs to clinch a fist with both hands, which creayes a virtual yoke, then to move your hand forward and backwords. Keep the same yoke form and turn your hand left to turn left and the same goes for turning right.
These hand gestures are basic for drone control, in a later stage more complex movements can be added.

3.3.2 Hands gestures and tracking

The application will accept video input from the Kinect. The region of interest is the hands. The application needs to be able to track the hands and also detect the gestures made by the user of the application. OpenCV will be used to track the hands and to determine which hand gestures were made.

3.3.3 Drone control

In order to control the drone the user will perform a predefined hand gesture. If the gesture is to rotate right then the drone needs to rotate right. In order to control the drone we need to understand the packets that are sent to the drone from the standard drone controller. Drones use WiFi to communicate with the controller. The packets that are sent over the WiFi of the drone uses UDP. Therefore if the user perform a gesture, the drone will get the commands via network packets.

3.4 Existing solutions

? has created a AR. Drone control of Flight Operator using the Depth Map of Microsoft Kinect. Using the depth map will render a skeleton from the user who is standing in front of Kinect. Then the user has to perform a set of gestures to control the drone. The 3D coordinates , (x,y,z) are used to control the drone. The system is accurate only over a useful range. Sanna et al. created a drone controller which uses FAAST(Flexible Action and Articulate Skeleton Toolkit), in which they use the skeleton render of the user to determine which gesture is being made. The latency of system is 2 seconds.

3.5 Best solution

The best solution is be to detect hand gestures and recognise the commands for the drone. Then send the command to the drone via WiFi as an UDP packet.

This solution can be extended to us human movement to control drones in real-time. Tools need to implement the solution is a Microsoft Kinect and a computer will OpenCV. In order to track the arms system will use the skeleton approach to locate the arm and determine the position.

3.6 Testing the solution

The solution will be tested by calculating the accuracy of hand gestures of controlling the drone. Also, testing to determine if the application can send control commands to the drone in real time and if the system can determine the hand gesture in real time will be carried out.

3.7 Conclusion

This chapter discussed the designer interpretation of the problem and provides a framework to start developing with. The next chapter displays the project plan.

Chapter 4

Project Plan

Term 1

Research

- Learn how to use OpenCV.
- Learn how to send commands to the drone

Term 2

Prototype

- Accurately locate region of interest.
- Send commands to drone over WiFi.
- Recognise a predefined set of hand gestures.
- Use the recognised gestures to control the drone.

Term 3

Implementation

- Recognise additional set of hand gestures to control the drone.
- Improve on hand gesture recognition.

Term 4

Testing

- Test the accuracy of the system.
- Test the accuracy/speed of the gesture detector.
- Test the accuracy/speed of the sending a signal to a drone.

Conclusion

This chapter states the projects plan. The next chapter will describe how the system will be displayed to the user, in terms of GUI(Graphical User Interface).

Bibliography

Boudjit, K., Larbes, C., and Alouache, M. (2013). Control of flight operation of a quad rotor ar. drone using depth map from microsoft kinect sensor. *International Journal of Engineering and Innovative Technology*, 3(3):15–19.

Bradski, G. and Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library.* " O'Reilly Media, Inc.".

Ikeuchi, K., Otsuka, T., Yoshii, A., Sakamoto, M., and Nakajima, T. (2014). Kinecdrone: enhancing somatic sensation to fly in the sky with kinect and ar. drone. In *Proceedings of the 5th Augmented Human International Conference*, page 53. ACM.

Sanna, A., Lamberti, F., Paravati, G., and Manuri, F. (2013). A kinect-based natural interface for quadrotor control. *Entertainment Computing*, 4(3):179–186.