ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 3, September 2013

Control of Flight Operation of a Quad rotor AR. Drone Using Depth Map from Microsoft Kinect Sensor

K. Boudjit, C. Larbes, M. Alouache

Abstract: ***In recent years, many user-interface devices appear for managing variety the physical interactions. The Microsft Kinect camera is a revolutionary and useful depth camera giving new user experience of interactive gaming on the Xbox platform through gesture or motion detection. In this paper we present an approach for control the Quadrotor AR.Drone using the Microsft Kinect sensor.***

Key Words: **AR. Drone, Quadrotor, Kinect, Gestuelle Control, UAV.**

I. INTRODUCTION

The Microsoft Kinect sensor device was released for the Microsoft Xbox 360 video game console at the end of the year 2010. The device allows a user to play video games by just moving his body and therefore allows gaming without the use of any game pad or joystick [1] [2]. In brief, the Kinect includes a color (RGB) camera, an infrared depth sensor, an accelerometer, four microphones and a motor to adjust the tilt. In addition to the commercial success of the Kinect as a gaming device, because of the numerous integrated features, its low price and off the shelf availability, the Kinect has also attracted a lot of interest from the scientific / robotic community. Soon after the release of the Kinect [1] [2], some groups have started to use the Kinect as a sensor for robotic applications. Furthermore, a module (OpenNI) for the Robotic Operating System (ROS) was made available. On the hand, the Open NI Framework provides the interface for physical devices and for middleware components; APIs enable modules to be registered in the OpenNI framework and to be used to produce sensory data. OpenNI covers communication with both low-level devices (e.g Microsoft Kinect), as well as high-level mideeleware solutions (e.g FAAST)[2]. OpenNI can interact with the Microsoft Kinect by the OpenKinect library. Body postures detected by FAAST are used by the GUI to trigger a modified version of the Keyboard-based AR.Drone client, thus implementing an effective and robust command. Chain to control the platform. Moreover, the GUI has been designed to receive information about position and orientation of the platform from an "external" system (e.g., an optical tracking system or a visual pose estimation system), thus supporting the implementation of AI (Artificial Intelligence) mechanisms to replace the user's control [3] [4]. Human–robot interaction (HRI) is a subset of HCI and can be considered as one of the most important domains of the computer vision. Although a lot of works based on gesture recognition in the domain of HRI are known in the literature recent technological advances have opened new and challenging research horizons [5] [6]. In particular, controllers and sensors used for home entertainment can be exploited also as affordable devices supporting the design and implementation of new kinds of HRI. Due to the importance of depth maps in robotics, this paper attempts to quantify the accuracy, performance and operation of the Kinect sensor, as no public information is available [7]. Furthermore, we test the Kinect sensor and its suitability for use in dynamic robotic environments by using the computed depth map to control the altitude of a flying Quad rotor AR.Drone helicopter. The project consists of a communication network made up of a quadrocopter, "parrot AR. Drone", "Kinect sensor" and a laptop. The aim of the project is to automate the procedures for navigation and control of the quadrocopter using the "Kinect sensor". This paper presents the methods and techniques used in the implementation of the Kinect sensor-quadrocopter automation system. The system is implemented entirely in JAVA, a programming language. The developers of the Microsoft Kinect sensor and the Parrot AR drone have provided required libraries in C# to aid with further development of the devices.

II. KINECT HARDWARE DETAILS

The kinect sensor connects to a PC/Xbox using a modified USB cable. The physical USB interface remains unchanged, however subsequent to the Kinect release the protocol was decoded and software to access the Kinect was enabled. The following figure shows a picture of the Microsoft Xbox 360 Kinect [1] [2].

Fig.1. Kinect Xbox 360 Microsoft

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 3, September 2013

The Kinect features two cameras, and a Micron MT9T112 640 x 480 pixels RGB camera, and 1.3 megapixel monochromes Micron MT9M001 camera filter with and IR pass filter. Accompanying the monochrome IR camera is a laser diode for illuminating the seen. Through reverse engineering it was determined the depth map has 11-bit resolution, and the video 8-bit, Despite the monochrome IR camera having higher resolution, both cameras only deliver 640 x 480 Pixel images. Both image sensors have and angular field of view of 57° horizontally and 43° vertically [1] [2]. The Kinect also features a microphone array and a motorized pivot, although neither of these features were required for visual flight control nor subsequently tested as part of this evaluation [8] [9]. The following table illustrates the main features of the Kinect.

Image	Resolution	Channels	Accuracy
Depth	640*480 pixels	1	0-2047
RGB	640*480 pixels	3 (R, G, B)	0-255
Infrared	640*480 pixels	1	0-255

Table.1. Characteristics of the Kinect

The information that applies the most for indoor navigation is the field of view and data steams shown above. The field of view will help in calculating the distance as an approximate of an abject that is ahead of the Kinect sensor. The data streams will help finding out the speed of processing the relayed raw data sent by the Kinect sensor to be processed by a computer unit.

III. QUADROTOR HELICOPTER EXPERIMENTAL PLATFORM

The AR.Drone (Figure 2) is a remote-controlled consumer quadrotor helicopter developed by Parrot. The body is made of a carbon fiber tube structure and high resistance plastic. A protection hull is made of expanded Polypropylene (EPP) farm, which is durable, light in weight and recyclable. The hull provides protection during indoor flights. The propellers are powered by four brushless motors (35,000 rpm, power 15 W). Energy is provided by a Lithium Polymer Battery with a capacity of 1000 mAh, which allows a flight of approximately 10 minutes [3]. The AR. drone carries an internal computer with a 468MHz ARM9-Processor and 128 MB of RAM, running a custom Linux operating system. A mini-USB connector is included for software flashing purposes and to attach add-ous (e.g., GPS sensor). An integrated 802.11g wireless card provides network connectivity with an external device that controls the vehicle [3]. A remote control is not included instead, a regular Wi-Fi enabled device was initially designed for Apple platform (e.g., iPhone, iPad, and iPod) and because available on other platform during 2001. it is also possible to control the AR.Drone farm a Linux or Windows PC with the software designed for application developers.

Fig.2. Quadrotor AR.Drone

IV. AUTOMATION AND CONTROL

The ability to recognize gestures is important for an interface developed to understand user's intentions. Interfaces for robot control that use gesture recognition techniques have been studied in depth, as using gestures represents a formidable challenge. In fact, several issues arise from environments with complex backgrounds, from dynamic lighting conditions, from shapes to be recognized (in general, hands and the other parts of the human body can be considered as deformable objects), from real-time execution constraints, and so on [9]. To realize this project several steps must be taken to correctly recognize a movement of the user and recreate moving the quad-rotor. We must begin by collecting data from different movements using the Kinect, and then get several ways to do the same command with natural gestures. We must then analyze the data and run tests on it to find the best features to be extracted for optimal recognition [10]. Once these parameters are determined, we must train a neural network or in the analyzes above, with a large number of examples of movements, and if possible, different people. Once trained, the network must be exported to a file. Then it is imported into the control program for the quad-rotor that we can recognize the movements live and translate directly into understandable commands to the drone [1]. Microsoft Kinect, which is connected to a PC (control station) via USB follows the body of the user, the gestures are translated into commands to be sent to the AR.Drone via Wi-Fi. The objective is to allow the user to fully control the movements of quad-rotor using the body as a kind of natural controller [1] [2]. Moreover, the GUI (Graphic User Interface) allows the user to remotely monitor the parameters of the platform, the flight altitude, navigation data (telemetry) and streaming video from onboard cameras. Figure 3 shows the principle of connection between the UAV and Kinect [2]. The computer acts as an interface between the AR. Drone and the Kinect sensor. An automation program implemented in Java and

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 3, September 2013

runs on the computer when it is running the quad-copter and the Kinect sensor will be connected.

Fig.3. Principle of connection between the AR.Drone and the Kinect

The purpose of use of the sensor is KINECTS extract the position of the at least four user information for use to control movement of the UAV. The control must be intuitive, while limiting the risk of loss of control. For this, the skeleton is extracted from the three-dimensional position of the user and his hands are used. Figure 4 shows the skeletons in three dimensional.

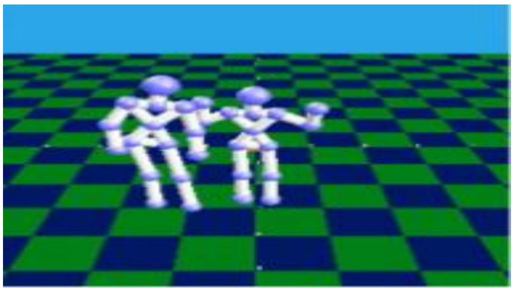

Fig.4. Skeletons in three-dimensional

To get to off the AR.Drone one must go through four stages, each stage corresponds to a specific gesture. Figure 5 shows the list of possible movements, after running the main GUI class; we obtain the following GUI figure 6.

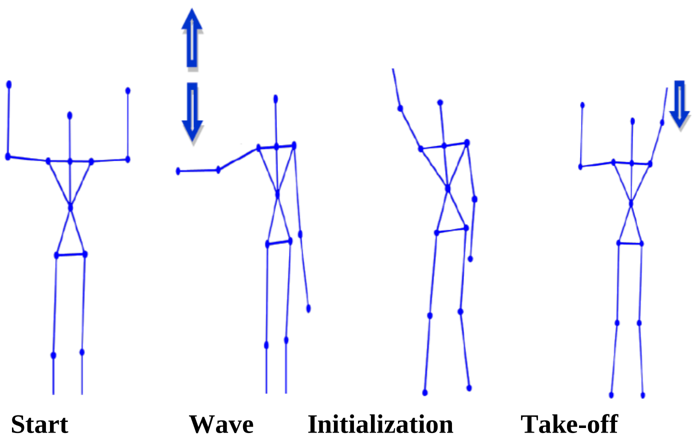

Fig.5. Movement Control

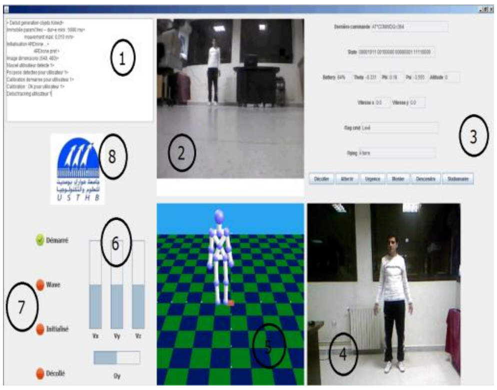

Fig.6. GUI for the execution of the Control tasks

1- This is an area of commentary, which shows: the connection with the drone, the detection of the user steps off, the position of both hands .... etc..
2- Camera AR.Drone
3- A window that displays the parameters of the drone: battery level, flight status, Vx, Vy ... etc..
4- Camera RGB Kinect.
5- Extraction du squelette de l'utilisateur en 3D en utilisant le capteur de profondeur de la Kinect.
6- Les trois barres verticales montrent la commande du drone en vitesse de translation sur les trois directions X, Y et Z, l'autre barre horizontale
7- The four switches that define the state of the system.
8- Logo of Our University

V. CONTROL RESULTS

In this part of testing, we will show some tests that have been made through our application. For this part, we used the SDK to retrieve ARDrone curves. The figure 7 illustrates the principle of data transfer between the kinect and the drone. And programming the various parts

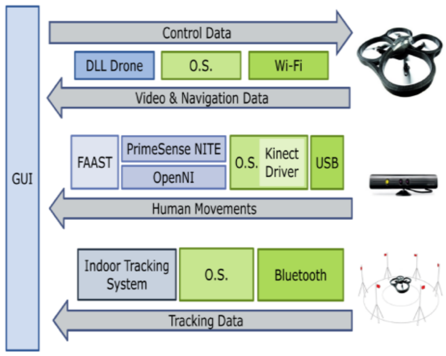

Fig.7. The principle of data transfer between the Kinect and the drone

A. Test for pitch

In this part we will send a command to the quadrotor using hand gestures for a flight in pitch. Figure 8 and 9 shows the results obtained during a test pitch.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 3, September 2013

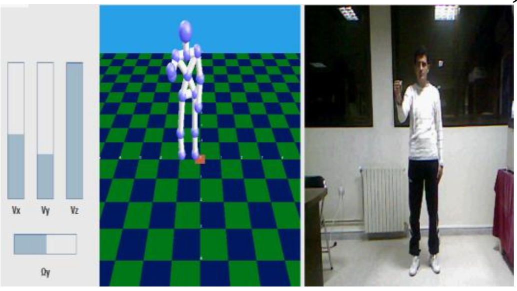

Fig.8. Test for using Kinect (Pitch).

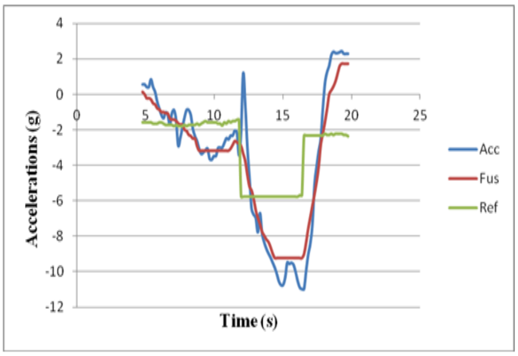

Fig.9. Test for pitch with disturbance

Figure 8 shows that when used in the gesture for a flight in the pitch quadrotor executes the instruction in real time. Figure 9 shows that when applied to a set (in green), the quadrotor perfectly follows the reference (in blue)

B. Test for Roll

In this section we test the roll, sending instructions to the quad rotor using kinect through appropriate actions. Figure 10 and 11 shows the results obtained during a test Roll.

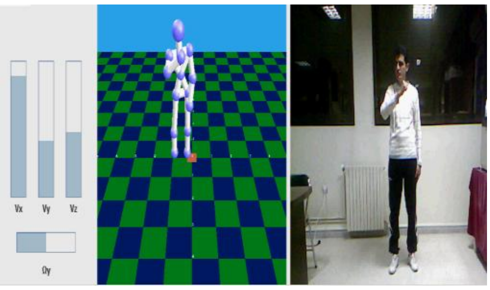

Fig.10. Test for using Kinect (Roll)

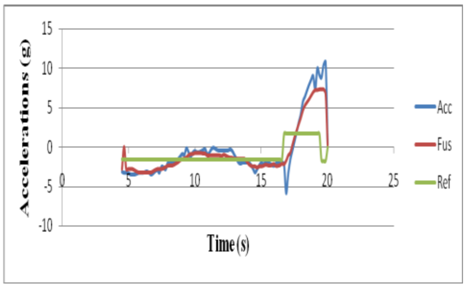

Fig.11. Test for pitch with disturbance

Figure 10 shows the action taken for a flight in Roll through the graphical interface. The quadrotor performs a flying roll when it receives instructions. Figure 11 illustrates the results obtained during the application of a disturbance (in green), the quadrotor perfectly follows the reference signal and the acceleration (in blue) shows that the response and immediate and in real time. The results obtained during the testing operation with Roll and Pitch disturbance, we noticed:

-Engines AR. Drone reacted immediately given maneuver (Real-time response).

-The stability of the motors and it confirms the proper functioning of the gesture control.

- The change of three vertical bars by changing the position of the left hand showing the drone control travel speed on the three directions X, Y and Z.

VI. CONCLUSION

The Microsft Kinect camera is a revolutionary and useful depth camera giving new user experience of interactive gaming on the Xbox platform through gesture or motion detection. In recent years, many user-interface devices appear for managing variety the physical interactions. In this paper we present an approach for control the quadrotor ar.drone using the Microsft Kinect sensor. The successful control of quadrotor using the Kinect depth map demonstrates that the sensor is capable of operation in dynamic environments. Its low cost, high frame rate and absolute depth accuracy over a useful range make it suitable for use on robotic platforms. Further work will involve integrating the Kinect into the navigation layer of the quadrotor system.

REFERENCES

[1] R. Asiimwe, A. Anvar, “Automation of the Maritime UAV Command, Control, Navigation Operations, Simulated in Real-Time using Kinect Sensor: A Feasibility Study”, World Academy of Science, Engineering and technology 72, 2012.

[2] A. Sana, F. Lamberti, G. Paravati, and F. manuri, « A Kinect-Based Natural Interface for Quadrotor Control », 2013 International federation for Information Processing Published by Elsevier.

[3] Ar.Drone. web site <http:\\ardrone.parrot.com>.

[4] F. Lamberti, A. Sana, G. Paravati, « Improving Robustness of Infrared Target Tracking Algorithms Based on Template Matching », IEEE Transactions on Aerospace and Electronic System 47 (2) 2012 pp. 1467-1480.

[5] D. Tsetserukou, J. Sugiyama, J. Miura, « Belt Tactile Interface for Communication with Mobile Robot Allowing Intelligent Obstacle Detection », in : Proc, IEEE World Haptics Conference (WHC 2011), Istanbul, Turkey 2011, pp. 113-118.

[6] J. Stowers, M. Hayes, B-S. Andrew, “Altitude Control of a Quadrotor Helicopter Using Depth Map from Microsoft Kinect Sensor”. Proceeding of the 2011 IEEE international

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 3, Issue 3, September 2013

Conference on Mechatronics, April 13-15, 2011, Istnabul, Turkey, pp. 978-985.

[7] S.C. Diamantas, A. Oikonomidis, and R.M. Crowder, “Depth Estimation for Autonomous Robot Navigation: A Comparative Approach”, in Imaging Systems and techniques (IST), 2010 IEEE International Conference on, July 2010, pp. 426-430.

[8] J. Stowers, M. Hayes and A. Bainbridge-Smith, “ Quadrotor helicopters for Visual Flight Control”, in Proceeding of Electronics New Zealand Conference 2010, Hamilton, New Zealand, 2010, pp. 21-26.

[9] J. Charles, M. Everingham,”Learning Shape Models for Monocular Human Pose Estimation from the Microsoft Xbox Kinect”, 2011 IEEE International Conference on Computer Vision Workshop pp. 1202-1208.

[10] J. Tong, J. Zhou, L. Liu and All, “Scanning 3D Full Human Bodies Using Kinects”, IEEE Transaction on Visualization and Computer Graphics, vol. 18, N° 4. April 2012. Published by the IEEE Computer Society, pp. 643-650.