Framework to evaluate the control and navigation systems for UAVs using the kinect sensor

Paulo R. S. Custódio\textsuperscript{a,1}, Bruno M. Pera\textsuperscript{a}, Gustavo C. Silva\textsuperscript{b}, Helosman V. Figueiredo\textsuperscript{a c} and Osamu Saotome\textsuperscript{c}

\textsuperscript{a}Laboratório de Robótica e Veículos Autônomos, Universidade do Vale do Paraíba, São José dos Campos, SP, Brazil
\textsuperscript{b}PROBES - Pesquisa, Inovação e Desenvolvimento LTDA, São José dos Campos, SP, Brazil
\textsuperscript{c}Instituto Tecnológico de Aeronáutica, São José dos Campos, SP, Brazil

Received on September 30, 2016 / accepted on October 19, 2016

Abstract

To make possible the evaluation of control system of Unmanned Aerial Vehicle (UAV) in indoor environment, it was necessary the tracking of UAV to get its position in the space (axis X, Y and Z) and attitude (Pitch, Roll and Yaw). These data has been acquired in outdoor environment through GPS (Global Positioning System), in indoor environment, it is impossible to use GPS because to get the position it is necessary to receive data from a minimum number of satellites and due to obstacles such as walls, rooftops and others, this minimum number is not reached. There is alternatives being developed to three-dimensional tracking in indoor environments, some of them are already commercial such as Vicon system, that have a precision of millimeters, however there is high cost on it. This work proposes a solution using a popular sensor used in video games, the Microsoft Kinect V1 and the software FLOSS (Free/Libre and Open Source Software). The image processing track the marks made on the UAV getting data that allows the evaluation of UAV, verifying the performance of control system embedded. The proposed system shows a precision of centimeters in the axis X, Y and Z, with this result we can confirm that the proposed system has a good performance to track the UAVs in indoor environment besides to use a low cost hardware and an open source software that will be available for other users on github platform.

\textbf{Keywords}: UAV, Kinect, Image Processing, OpenSource.

\textsuperscript{1}E-mail Corresponding Author: paulo55866@gmail.com
1. Introduction

The usage of unmanned aerial vehicles (UAV’s) has grown up in the last years, especially for the ability to fly over hostile locations where a man cannot reach, it can be used in many areas such as military, academic research and commercial systems. In all applications it is required autonomy and self location capacity in the environment [1].

The UAVs in outdoor environments uses a Global Positioning System (GPS) for navigation, therefore in an indoor environment the use of this system is impossible due the attenuation of GPS signals caused by physical barriers resulting in positioning errors that cannot be ignored. Basically the GPS works using the propagation of signals between satellites and the receiver module, to get the position the module performs the calculation based on the velocity and the travel time of signals from each satellite, to obtain a good precision the receiver must receive signals from 7 satellites, in the indoor environments this number is not reached [2].

At the moment, UAVs tracking systems in indoor environments are in development, the most common solutions are: cameras network, radio based localization and pseudolites that replicate satellites signals. In the cameras network system, the cameras are strategically positioned to have a general view of the desired environment. Among the available systems that use cameras to track objects, Vicon system stands out as a motion capture using markers in the object that are tracked by infra-red (IR) cameras, the only problem of use this system is the high cost of an individual camera [3].

Microsoft Kinect is an efficient sensor to get three-dimensional images, it was developed as motion sensor control for XBox 360 video game, therefore it is being used in several areas of research where the use of three-dimensional images is required [4].

This paper evaluate a system capable to track UAVs in an indoor environment using a commercial off-the-shelf (COTS) hardware and open source software, making the use of tracking systems accessible for everyone.

2. Related work

The use of kinect sensor with the UAVs is not new. There is some papers that describe it, some are used to control and others to map the environment. A JAVA code based was used to control the quad rotor using gestures that the kinect already recognizes and send the signal via Wi-Fi [5]. In another study, the kinect sensor was instructed with aircraft marshalling gestures commands in real time to control the UAV [6]. In the use of sensor to map area, the sensor is carried by the UAV, some applications use it to measure
distance from floor, it measure the Y axis of the UAV and use a Proportional Integral Derivative (PID) system to keep it in an altitude \([7, 8]\). The most of works using kinect in this area are focused on control by gesture and self mapping environment, the present paper is focused on use the kinect sensor to track the UAV in indoor environment.

3. Materials

In development of this paper, it was used the Kinect sensor V1, its basic architecture is composed of a RGB camera capable to record images of 640x480 pixels in a rate of 30 frames per second, an IR projector and an IR camera that detect the light mirrored by projector \([9]\). In the Figure 1 it is observed how the main components are distributed in the sensor.

![Figure 1 - Kinect components.](image)

From kinect components, the most important in this project are the two cameras and IR projector, the RGB camera used to track a specific coloured point and the combination of IR camera and IR projector to determine the depth of image.

It was choosen the utilization of a computer having a Linux operational system (OS), with distribution Ubuntu 15.10. The software libraries used were the OpenCV \([?]\), responsible by image processing and the OpenKinect, responsible by communication with the sensor. Both libraries are supported to work with Python programming language, that was used in the development.

4. Object detection

The first stage of development was write the code to obtain data from sensor, the library OpenKinect provides functions that make possible the sensor control and data acquisition. In the project it was necessary the
acquisition of data from RGB camera and from IR camera, as said, the RGB camera provides an image with 640x480 pixels and the IR camera provides a 640x480 matrix with values from 0 to 2047 (11 bits values) [11].

With the data obtained from sensor, it was necessary the development of an algorithm to detect a specific object in the image, once that was attached a solid color sphere on the UAV to shows its position. The process of sphere detection was made from the acquisition of RGB image, the algorithm developed can be visualized in the flowchart in the Figure 2, where it is made the conversion of color format RGB to HSV (Hue, Saturation and Value), because in the HSV domain the colors components are separated from intensity, making easier the detection of objects [12]. From HSV image, were extracted 3 matrix, each one of them represents a component from original image H, S and V, with that it was made a selection of values from each matrix in an interval of the color to be detected, generating a black and white image for each component, for example: assuming that the interval V is between 50 and 70, in the image generated the pixels that are not in this interval are black and the pixels in this range are white. From these three images corresponding to each component (H, S and V), it was applied the logical operation \textit{AND} between those images resulting in one image with corresponding area from sphere in the white color. Next, it was applied the filters \textit{dilate} and \textit{gaussianBlur} with the objective to reduce the noise on image. In the resulting figure it was applied the detection of circles that define the points X and Y from center of detected circle, as the only circle resulting in the image will be the correspondent to a sphere, in this case this points X and Y represents the UAV mark on the image. The functions used to convert the image, filters and detection of circles was provide by OpenCV library.
5. Distance detection

To get the distance between a point and the kinect sensor, it was necessary to use the data obtained from IR camera, as said previously these data are a matrix 640x480 with values from 0 to 2047. The values of distance are inversely proportional to sensor depth, the documentation of library OpenKinect describes a way to convert depth data to values in centimetres, as shown in the Equation 1, this consider information related to IR camera focus. Where the \(\text{distCM}\) is the value of distance in centimetres and \(\text{valProx}\) is the value of depth (11 bits number) [13].

\[
distCM = 12.36 \times \tan\left(\frac{\text{valProx}}{2842.5} + 1.1863\right) \tag{1}
\]

With the tests using the kinect sensor, it was realized that there is a displacement from images obtained from two cameras, when a pixel in the RGB is not the same physical point in the depth image due the difference of position of cameras on the sensor. There is an algorithm, called registration, that do the alignment of the images to have the physical point in the same pixels, however in the case of this specific application, it was observed that is not necessary the implementation of this algorithm, once that the tracked object keep the proportion in both images, using the radius of sphere is possible to determine with precision the pixel that corresponds the sphere in the depth image, this is doing by the sum of radius in the X position
of pixel and subtracting the value of radius in the Y position of pixel, the Figure 3 shows the displacement and the correction made by the code.

![RGB IMAGE](image1.png) ![DEPTH IMAGE](image2.png)

**Figure 3** - Displacement and correction.

With the determination of the sphere detected, it was possible the conversion of pixels position X and Y to real measure in centimetres, considering the center of image in the points(0,0) it was used the Equations 2 and 3 that convert the values to Cartesian coordinates in centimetres. Where $x$ is the distance in X axis in centimetres, $y$ is the distance in Y axis in centimetres, $z$ is the distance of pixel in centimetres from sensor, $i$ and $j$ are the pixel points, $w$ is the width of image and $h$ is the height of image [14].

$$x = (i - \frac{w}{2}) \times (z - 10) \times 0.0021$$  \hspace{1cm} (2)

$$y = (j - \frac{h}{2}) \times (z - 10) \times 0.0021$$  \hspace{1cm} (3)
6. Results and Discussion

In this section is presented and discussed the results obtained from project development. It was made an analysis about how the RGB-D sensor Microsoft Kinect behaviours to track a UAV in an indoor environment.

By the timing tests made, we get the following acquisition times for RGB camera, depth camera and the processing time of object detection, the Table 1 shown the results.

<table>
<thead>
<tr>
<th>Time (s)</th>
<th>RGB</th>
<th>Depth</th>
<th>Processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>fps</td>
<td>30</td>
<td>29</td>
<td>*</td>
</tr>
</tbody>
</table>

First was made a test to evaluate the sensor precision. Was created an environment with defined distances and compared the data that the sensor return to check the reliability level of it.

After the tests to determine the sensor precision we observe that the axis Z varies according to real distance, in the interval of 0.5 m and 2.5 m the measure error is millimetric, as the axis X and Y depends of axis Z they also show a millimetric precision in the same distance. With this test we can determine the region of depth sensor, to be able to detect is necessary the minimum distance of 50 centimetres and maximum distance of 3.6 meters, as you can see in the Figure 5, to be possible to track the UAV.

![Figure 4 - Detection region.](image)

With the determination of detection region, it was made tracking UAV tests, we attached a red sphere in the aerial vehicle, the developed code was capable to determine the spacial position of the sphere as shown in the
Figures 6 and 7. It was drawn a green circle around of sphere and the side of it are described its position in the axis X, Y and Z in centimetres, the blue cross show the center of image where the values of X and Y are both 0.

**Figure 5 and 6 - UAV Tracking.**

The software register the position of aerial vehicle for its whole route. So, that way, we could analyse the behaviour of UAV during the fly, also we could generate a chart of route as is displaying in the Figure 8. The region with more dots means that the UAV stayed there for a while more than other place.

**Figure 7 - UAV Route.**
7. Conclusion

With the result obtained from project development we can say that the
kinect shows a good yield to track UAV in indoors environment with low
density of obstacles, returning a millimetric precision of the position in the
most part of its detection field.

To conclude the usage of this system to track the UAV in indoor en-
vironment, it will change according to the precision that is waited for the
application where the UAV is inserted. Being highly recommended to track
stables UAVs, low speed and in the environment with low density of objects
and standard illumination.

Acknowledgments. We thank the Laboratory of Robotics and Au-
tonomous Vehicles(LRVA) of Universidade do Vale do Paraiba for the sup-
port and the use of its installations.

References

[1] Stegagno, P. Indoor Localization of UAVs, Max-Planck-Gesellschaft,
2015.

[2] Mui, M., Chintalapally, A. The Rising Demand for Indoor Localization

recorded from a Vicon system and a Microsoft Kinect sensor, Proceed-
ings of the ACM Symposium on Applied Perception, August 03-04,
2012, Los Angeles, California.
doi:10.1145/2338676.2338703

19, no. 2, pp. 4-10, Apr. 2012.
doi:10.1109/MMUL.2012.24

Quad rotor AR. Drone Using Depth Map from Microsoft Kinect Sen-
sor, International Journal of Engineering and Innovative Technology

[6] Asiimwe, A., Anvar, A. Automation of the Maritime UAV Com-
mand, Control, Navigation Operations, Simulated in Real-Time Using


