Autonomous Visual Navigation for Indoor Flying Drone

Omar Alkayali
Malmö, Sweden
e.omark@hotmail.com

Tahereh Morteza
Malmö, Sweden
naghmeh.morteza@gmail.com

Ömer Caner Eren
Malmö, Sweden
canereren112@gmail.com

ABSTRACT

In recent years, both remote controlled and autonomously flying drones are heavily used in different domains. Many diverse applications have been introduced to fly and control AR drones autonomously. The main objective of this paper is to autonomously fly an AR Drone 2.0 indoors using onboard cameras and sensors. The developed code will be able to fly a drone autonomously by following a green line in the floor using image processing. We aimed to create a solution that only relies on built-in cameras and sensors of the AR Drone 2.0 to reduce time, cost and effort of implementation. We use functional testing to calculate the accuracy of our solution. We created a feasible, less costly and easy implemented solution to provide autonomy and location calibration through a predefined path for a drone.

Keywords,

AR Drone 2.0, indoor, autonomous flight, built-in camera

INTRODUCTION

In recent years, drone technology became very familiar in our life due to its spreading usage in technology, education and commercial sectors. Drones known also as a quadrotor helicopter or quadcopter which is an aerial vehicle supported by four rotors [13]. They are either remote controlled or autonomously flying Miniature Aerial Vehicles (MAVs) not only used in military domain, but also in civilian environments [8]. Mostly, they are used for exploration and observation purposes in both indoor and outdoor environments.

Affordable drones appeared in the market in autumn 2010 and they are equipped with necessary sensors and user interfaces [13]. They are even available nowadays as a high-tech toys with very cheap and affordable prices. Moreover, drones can perform easy and complex tasks based on the sensor types it carries and the technology it adapts. Besides military use, drones are used in item delivery (like AMAZON), search and rescue, exploratory tasks, firefighting, security and surveillance and much more. In addition to that, they caught attention of universities and researchers, and are used widely in several researches all over the world [2,13].

Unmanned Aerial Vehicles (UAVs) is one of the vital elements used widely for automating not only drones, but all technology aspects [1,3]. Many researches conducted lately in order to improve the functionality of the UAV's and completely automate them. This means UAV functionalities such as flying, data gathering and analyzing are automated without any human interaction [3].

Any Drone, in order to navigate, relies on set of sensors. Some of these sensors measures orientations, acceleration and absolute position using GPS based navigation system [14]. A major challenge is that indoors, where there is no GPS signal available, other methods need to be used to determine the position. Fortunately, there are a wide range of sensors to achieve this goal, ranging from inaccurate ultrasonic range sensors up to very expensive high-resolution laser range scanner. Moreover, drones can rely on expensive external localization systems such as ViCons [14,15,16], or use artificial and easy distinguishable tags or landmarks [17]. One of the most common ways nowadays are optical built-in cameras as they are small, light, cheap and energy-efficient. UAV sensory data are sent to a ground station to perform the needed processing and localization algorithms [18]. Handling and processing such amount of sensors' data has proven to be a very challenging mission [8]. Furthermore, it has been reported by the authors of these researches not only issues with algorithm speed but also issues with occasional tags deficiency [14].

In order to create an affordable and autonomous flying drone, we are introducing a solution using AR Drone 2.0 where it follows a green line in the floor using image processing. The reason why we chose this solution is that we are going to rely only on built-in cameras and sensors, besides the use of a code for line detection. This will reduce time and cost for implementation. The developed solution will then be evaluated against its accuracy.

In the following chapter, we are going to summarize the state-of-the-art of the current indoor localization techniques. After that, we are going to introduce our detailed solution design and implementation. Then, the results are discussed in another section followed by evaluation. Finally, we conclude and describe possible future work.

RELATED WORK

Recently, quadrocoptors such as Drones have gained enormous interest in research in both robotics and IT especially for autonomous flight. In order for these quadrotor to localize themselves autonomously, they use built-in cameras and sensors.

One of the crucial elements for automation used widely in all technology aspects nowadays is the Unmanned Aerial Vehicles (UAVs) [3]. It has been proven that it can be part of nearly any application in various fields due to its light wait and expediency of use. Moreover, numerous researches were done to automate the drones through improving UAVs functionalities. This means that UAVs are programmed to fly, gather data, analyze it and take decision without human interaction.

Infrared localization system proposed by [20] where an infrared sensitive camera retrieves a vision data and determines the position of an object relative to the position of the camera. However, infrared can't function under natural or fluctuating light and it is too sensitive to the environment [20].

It was proposed by Engel, J [8] to fly and control AR drone autonomously using built-in sensors only in unknown environment. Visual analysis was used mainly to estimate the position of the drone as seen by the front camera, then the drone generates the required control orders to keep it on the desired route in 3-D space. The proposed solution by Engel was using Software Development Kit (SDK) which is provided by Parrot.

Though, drone camera visual information are in 2-D and it is difficult to use them to navigate in the real 3-D world. Unfortunately, due to the fact that algorithms used such as SLAM [10], use reliable odometry which is not available in aerial robots and the extra computational power needed to build a 3-D model for the environment [11]. Moreover, textureless surfaces like walls are difficult to be predicted in the constructed 3-D models and mostly have halls.

Another localization technique was used by [19], by using Ar Drone along with UbiSense UWB Localization and fixed 'landmarks' such as QR codes. The author mentioned a resulting sensor fusion of IMU, UWB and extended kalmann filtering besides a need for more sensors such as lasers for accuracy improvement.

Another proposed solution by Yue, S. [9] using external low-resolution camera sensor named Kinect to locate the position of the drone approximately during indoor flight. The proposed model was simulated in MATLAB and C++. The main controller used in this model is the PID and has been improved with different filters designs like Kalman filter. However, we are only relying on built-in cameras and sensors.

RESEARCH GOALS

The main objective of this paper is to autonomously fly an AR Drone 2.0 indoors using onboard cameras and sensors. The developed code will be able to fly a drone autonomously by following a green line in the floor using image processing. This goal gives rise to the following research question:

RQ: How to autonomously fly an AR Drone 2.0 to follow a line on the ground using image processing?

RESEARCH METHOD

Design and creation method is chosen particularly as it is the most suitable choice when developing new IT products, e.g. artefacts. The design and creation activities used in this research are based on well-established principles of systems development [12]. It is an iterative process which includes five steps: Awareness, Suggestion, Development, Evaluation and Conclusion. A functional testing well be done to evaluate the accuracy of the autonomously flying drone.

SOLUTION DESIGN

We tried to implement various different techniques to find a proper solution to provide AR. Drone 2.0 these attributes: autonomous fly and path calibration. First of all we focused on to find a way to make a drone fly autonomously. Parrot Company introduced a Software Development Kit (SDK) to provide developers an environment for building applications specifically for AR Drone 2.0[7]. We did a literature review related to application implemented by Parrot SDK, and Zhout et al. stated that compiling the Parrot SDK is hard and fixing the compiler problems takes a lot of time [6]. This problem forced them to use a ROS software to reach their goals. Then we decided that it will take too much effort to learn both Parrot SDK API and ROS Software. At the end we came up with a much more efficient solution called "Node AR Drone" library [22] that is designed for node.js [21] applications. We only needed to check "Node AR Drone" library api which was very well documented and required less effort comparing to Parrot SDK API.

Secondly we found 6 different solutions to calibrate the drone through its path. First we thought to implement infrared cameras, but this solution was an external device and an expensive solution and dependent on environmental factors such as light density [20]. Then we decided to place 4 ultrasound sensors in each corner of an indoor environment to locate the drone's position but required extensive instrumentation [20] and also a costly solution. There was another solution called image tagging [19] to calibrate the drone by processing the tagged images on its path. This solution was not costly but we needed to learn image tagging process which required time to learn. At the end we came up with a solution that requires very few effort, time and is not costly. We planned to paint a green path on the ground and use the Drone's bottom camera for capturing the images. We had a project previously to distinguish the green color from an image stream. This project used to notify the drone periodically if it is on the green path or not.

The project contains elements listed as below:

1. **Parrot AR Drone 2.0:** This device used as a UAV and performs autonomous flight through a green path.

   Technical Specifications:

   - **Processor**: ARM Cortex A8 1 GHz 32-bit processor with DSP video 800 MHz TMS320DMC64x
   - **OS**: Linux 2.6.32
   - **RAM**: DDR2 1 GB at 200 MHz
   - **USB**: High-speed USB 2.0 for extensions
   - **Wi-Fi** b g n
   - **Gyroscope**: 3 axles, accuracy of 2,000°/second
   - **Accelerometer**: 3 axles, accuracy of +/- 50 mg
   - **Magnetometer**: 3 axles, accuracy of 6°
   - **Pressure sensor**: Accuracy of +/- 10 Pa
   - **Altitude ultrasound sensor**:Measures altitude
   - **Vertical camera**: QVGA 60 FPS to measure the ground speed

2. **Client Computer:** This computer contain a client application that controls the drone automatically. The computer will establish a connection by using AR Drone's Wifi. This connection will be used to send autonomous flight direction commands and also image stream.
   Specifications:
   - Windows 7 Professional
   - Node.js v6.10.2 LTS
   - Wireless hardware
   - Node Ar Drone library for Node.js: https://github.com/felixge/node-ar-drone

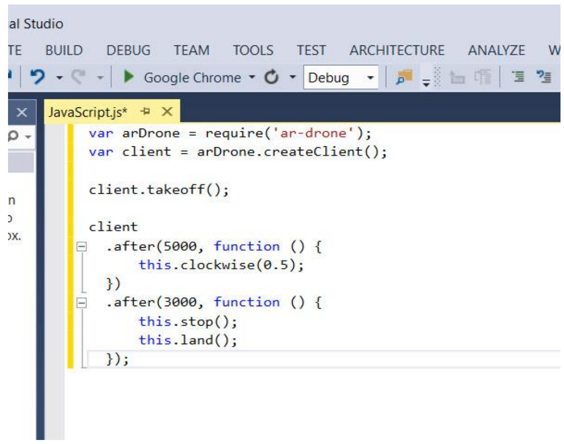
3. **Green Path :** There is green path drawn on the floor and used to help the drone to calibrate its direction. This solution requires almost no effort and also easy to create a green path on the floor by using an ordinary paint.

IMPLEMENTATION

1. Autonomous Drone Controller Application(ADCA)

This application programmed by using java script, node.js and "Node AR Drone" node.js library. Node.js is a platform built on Chrome's JavaScript runtime for easily building fast and scalable network applications [22]. "Node AR Drone" library is specifically developed for Parrot AR Drone 2 and provides:

- **Movement**: left, right, front and backward
- **Rotation**: clockwise, anti-clockwise
- Landing, taking off
- Sensor data information
- Camera streaming



This is a sample autonomous code that make drone take off and after 5 seconds drone turns half tour in clockwise direction. 3 seconds later drone lands on the ground.

ADCA is responsible for both autonomous drone control algorithm and image streaming using drone's built-in camera. **ADCA** serves the captured image stream through a basic http local port. Green Path Recognizer Application that runs simultaneously in the client program provides feedback to **ADCA** if the drone is following the green path with rest web service interface. **ADCA** periodically listens the web service to get more accurate information to calibrate its path through the green line.

2. Green Path Recognizer Application (GPRA)

GPRA is a Java application that notifies the **ADCA** application if the drone is in correct path or not. This application works as follows

- Fetches the image streams that are provided by **ADCA** from the localhost port.
- Analyzes the images by using OpenCV library.
- Provides Rest Web Service API for **ADCA.**
- If the image contains a green color, the rest web service api will return true otherwise false.

GPRA has as JavaFX UI as below:

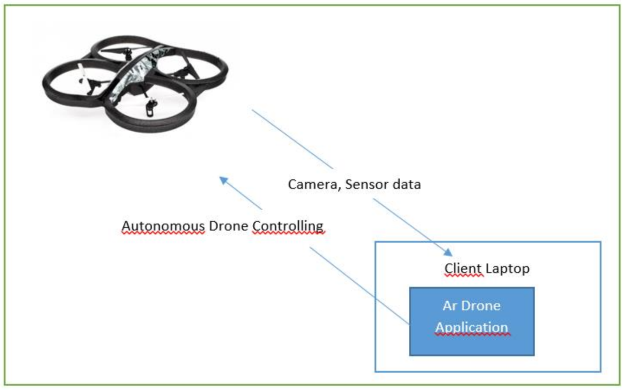
- The JavaFX Scene Builder is used to create visual layout for HSV (Hue, Saturation, and Value) controllers (sliders) and to display the image stream from camera, its mask and morph views.

- The live stream from camera is shown in the layout with HSV controllers and these controllers are used to select the color range of our specific green path.

- The image stream obtained from camera is pre-processed using OpenCV methods to obtain binary image.

ARCHITECTURE

This part represent the architecture for an indoor AR Drone system and Ar Drone Appliation contains two applications called **ADCA** and **GPRA** applications. The **ADCA** is responsible to communicate and control the Drone. Camera stream received by **ADCA** and checked by **GPRA.** According to calibration result **ADCA** autonomously controls the Drone by using Node Ar Drone library api.



RESULTS

We were able to fly drone autonomously by using **ADCA** application. First drone took off approximately one meter from the ground and moved forward for 5 seconds. Then drone landed on the ground again. This whole process controlled by **ADCA** application by using “Node Ar Drone” library.

Secondly we tried to provide path calibration for the AR Drone 2.0. **GPRA** application is responsible to provide the feedback by using image processing algorithms for **ADCA** application. But we were not able to have a successful image stream from AR Drone 2.0 to **ADCA** and from **ADCA** to **GPRA.** We are aiming to provide both autonomy and path calibration for the drone after we successfully perform the images stream flow.

EVALUATION

To be written after the implementation is done.

CONCLUSION

Drone have penetrated the world of technology and many businesses are looking forward to utilize the capacities of these machines to the best effects. Among these studies we have estimated and proposed a solution for indoor location technology to calibrate the drone through its path. Our solution requires a user to paint a green on the ground, and then the drone autonomously flies according to that path. This solution does not require any additional devices or image tags unlike the other solutions. Eventually, we were able to create a solution that mainly focused on feasibility, low cost and easy implementation aspects to provide autonomy and path calibration properties for a drone.

The proposed solution has many avenues for future improvements and refinement. A further study on the latest localization technologies will be a possible future work. The use of tags and even external supported sensors might be another future study. Finally, an important future work could be to use visual navigation and camera calibration without tags or land marks.

ACKNOWLEDGMENTS

Our utmost gratitude goes to Shahram Jalaliniya and Thomas Pederson, who have been guiding us even before the beginning of this research. Your support and advice have been extremely crucial throughout this work.

REFERENCES

[1] Mallick, Tuton Chandra, Mohammad Ariful Islam Bhuyan, and Mohammed Saifuddin Munna. "Design & implementation of an UAV (Drone) with flight data record." Innovations in Science, Engineering and Technology (ICISET), International Conference on. IEEE, 2016.

[2] Bai, Guanghua, et al. "Research on obstacles avoidance technology For UAV based on improved PTAM algorithm." Progress in Informatics and Computing (PIC), 2015 IEEE International Conference on. IEEE, 2015.

[3] Al Habsi, Shaima, et al. "Integration of a Vicon camera system for indoor flight of a Parrot AR Drone." Mechatronics and its Applications (ISMA), 2015 10th International Symposium on. IEEE, 2015.

[4] Landry, Benoit. "Planning and control for quadrotor flight through cluttered environments." PhD diss., Massachusetts Institute of Technology, 2015.

[5] Eisenbeiss, Henri. "A mini unmanned aerial vehicle (UAV): system overview and image acquisition." International Archives of Photogrammetry. Remote Sensing and Spatial Information Sciences 36.5/W1 (2004): 1-7

[6] Zhao, Tianqu, and Hong Jiang. "Landing system for AR. Drone 2.0 using onboard camera and ROS." Guidance, Navigation and Control Conference (CGNCC), 2016 IEEE Chinese. IEEE, 2016.

[7] M. Parrot for Developers. Online available 2nd April 2017 http://developer.parrot.com/ar-drone.html

[8] Jakob Julian Engel, “Jakob Julian Engel , Engel, J. "Autonomous camera-based navigation of a quadrocopter." TUM University Thesis (2011).

[9] Sun, Yue. "Modeling, identification and control of a quad-rotor drone using low-resolution sensing." (2012).

[10] K. Celik, S.-J. Chung, M. Clausman, and A. K. Somani, “Monocular vision slam for indoor aerial vehicles,” in IROS, 2009.

[11] M. Goesele, N. Snavely, B. Curless, S. M. Seitz, and H. Hoppe, “Multiview stereo for community photo collections,” in ICCV, 2007

[12] Oates, B. J. (2005). Researching information systems and computing. Sage.

[13] Krajník, Tomáš, et al. "AR-drone as a platform for robotic research and education." International Conference on Research and Education in Robotics. Springer Berlin Heidelberg, 2011.

[14] Krajník, Tomáš, et al. "A simple visual navigation system for an UAV." Systems, Signals and Devices (SSD), 2012 9th International Multi-Conference on. IEEE, 2012.

[15] D. Mellinger, N. Michael, and V. Kumar, “Trajectory generation and control for precise aggressive maneuvers with quadrotors,” in International Symposium on Experimental Robotics, Delhi, India, Dec 2010.

[16] N. Michael, J. Fink, and V. Kumar, “Cooperative manipulation and transportation with aerial robots,” Autonomous Robots, Jan 2011.

[17] S. Lange, N. Sunderhauf, and P. Protzel, “A vision based onboard approach for landing and position control of an autonomous multirotor UAV in GPS-denied environments,” Electrical Engineering, 2009.

[18] M. Achtelik, A. Bachrach, R. He, S. Prentice, and N. Roy, “Stereo vision and laser odometry for autonomous helicopters in GPS-denied indoor environments,” in SPIE Unmanned Systems Technology XI, 2009.

[19] Benini, Alessandro, Adriano Mancini, and Sauro Longhi. "An imu/uwb/vision-based extended kalman filter for mini-uav localization in indoor environment using 802.15. 4a wireless sensor network." Journal of Intelligent & Robotic Systems (2013): 1-16.

[20] Lowe, Karsten. Distance estimation between transceivers over short distances. Diss. Clemson University, 2007.

[21] Node.js api available on 20th April 2017 https://nodejs.org/dist/latest-v6.x/docs/api/

[22] Node Ar Drone Librarty on 20th April 2017 https://github.com/felixge/node-ar-drone