CSCI 5551 Fall 2015 Term Project
UAV Applications – Image Based Tracking of Mobile User

John Erickson
eric0870@umn.edu

Computer Science and Engineering Department
The University of Minnesota at Twin Cities

I. Abstract

The intent of this project is to develop an image based tracking application for a quadrotor UAV. Much work has been done in industry and academia in the field of hobbyist level quadcopter development, resulting in open source software and hardware that are readily available and affordable. This project will leverage COTS hardware and open source software libraries in order to develop a client based control application that enables a drone to track a mobile user. The drone will autonomously maintain a desired distance and orientation from the user as it moves through space. This is not a novel application, but rather a milestone on the way to building up an autonomous flying system with marketable capabilities. This project will address a small number of trade studies, including the development platform, software development environment and software library support. The objective of these trades is early validation of the project concept. The majority of the work performed in this project will be in the domain of software development, specifically focusing on computer visualization and wireless communication. The outcome of this project will be twofold: an application potentially reusable by the development community, and a start on the development of a future robotic system.

II. Project Description

A. Definition of the Problem

The intent of this project is to develop an image based tracking application for a quadrotor UAV, more specifically, an application capable of allowing an autonomous flying system to track a mobile object using image recognition algorithms. The application must allow the user to command the drone to begin and end tracking, as well as indicate the desired distance and orientation the drone should maintain from the user. Initially a detector tag may be used for tracking; however, it is highly desirable to eventually migrate to tracking a user provided digital image.

In order to accomplish this goal, a system capable of autonomous flight control and real time feedback will be needed. At a minimum this system must possess the following: inertial measurement unit capable of reporting acceleration and attitude, altitude sensor, front facing camera, and wireless communication with client device. The following support is also highly desired: side and down facing cameras, magnetometer, user application processor resident on device, flight data and video stream recording capability. The tracking and controlling application may be hosted on either the drone itself or a client device (PC or Android based system).

B. Motivation

The motivation behind this project stems from a number of factors. One, to broaden experience working with autonomous flying vehicles from a control and communication perspective. Two, to develop an application potentially reusable by the community, academic or commercial. And three, to develop an application that may eventually be incorporated into a marketable product.

C. Related Work

Much work has been done in industry and academia in the field of hobbyist level quadcopter development, resulting in open source software and hardware that are readily available and affordable. From universities to start-ups to large corporations, time and money is being spent furthering the technology in the interest of utility, convenience, and profits. The concepts developed in this project are by no means novel; rather they are readily available in forms ranging from proprietary to open source. Companies like 3D Robotics, Parrot and EZ-Robot, among numerous others, have invested in software and hardware development, resulting in modular plug-and-play systems readily available to anyone with an interest in robotics.

D. Work Plan

The first phase of the project will be a series of trade studies to identify the development platform, software development environment and software library support. A number of development platforms will be traded, ranging from DIY to ready-to-fly systems. The selected development platform must be able to meet all of the system requirements of the project. The development environment and software library trades will assess the capabilities of available software development kits and community libraries supporting the selected development platform.

After selection of the development platform has been completed, a proof-of-concept task will be conducted. The primary goal of this task is to quickly prove out the ability to accomplish the overall project goal of mobile image tracking on the development platform. A community developed and/or commercially available application will be used for this task.

The project specific application development will follow the proof-of-concept phase. The software development will be conducted in the selected development environment, utilizing selected SDK's and software libraries. The output of this project phase will be a custom image based "Follow Me" application capable of tracking a mobile user. The validation of the final product will also be included in this phase.

Upon completion of the project, future work will be addressed and presented in the final report.

Configuration management of software developed for the project will be conducted via Git [A], and is available on GitHub at https://github.com/eric0870/CSCI_5551_Proj.

III. Work

A. Requirements

The system requirements for the project are provided in Table 1 below. These requirements were derived in the interest of completing the project within a reasonable budget and an aggressive schedule. They were used to govern the procurement of development hardware and tools, as well as drive the system and software development effort of the project.

ID	Requirement	Threshold	Objective
1	Cost of development Hardware	< $500	< $300
2	Cost of development Software	< $100	Free
3	Availability of development Hardware/Software	Now	
4	Project development schedule	2 months	
5	Development platform form factor	Quadcopter UAV	
6	Development toolchain support	Mature community/ commercial support	
7	Development platform sensor suite a. Inertial Measurement Unit b. Altitude sensor	 a. 6 DOF (accel, gyro) b. Ultrasound telemeter	 a. 9 DOF (accel, gyro, mag) b. Pressure sensor
8	Development platform video a. Camera(s) b. Video stream frame rate	 a. 1 (front facing) b. 15 FPS	 a. 2 (front, down facing) b. 30 FPS
9	Development platform support for user application software	Onboard	Remote
10	Development platform real-time communication with user application a. Communication type b. Data: user app to drone c. Data: drone to user app d. Rate	 a. Wired or WIFI b. Command / control c. Nav data, video d. 1 Hz updates	 a. -- b. -- c. -- d. 100 Hz updates
11	The system shall provide an application capable of image based tracking of a mobile user *Goal: the application should be agnostic to both the development platform and host device*		
12	The system shall provide a driver capable of routing images to the application, and control commands to the drone		

Table 1

B. Trade Studies

1. Development Platform Trade

A number of development platforms were traded, ranging from DIY to ready-to-fly systems. The outcome of the study was the selection of the AR Drone 2.0 by Parrot. The AR Drone is equipped with an onboard sensor suite and proprietary set of control algorithms that provide assistance to the user for all supported maneuvers [8]. Ergo, less time will be spent fine tuning vehicle stability, leaving more time to accomplish project specific goals. The AR Drone is also widely used by the community, resulting in numerous user applications available for flight demos, proof-of-concept and open source examples. Table 2 provides a summary of each of the development platforms included in the trade study.

Platform	Cost	Pros	Cons
Ardupilot, by 3D Robotics [1][2]	$500	Open SW/HW	Does not meet project development schedule requirement
Phantom 3, by DJI [3]	$700	High quality camera and sensors	Does not meet project cost requirement
AR Drone 2, by Parrot [4]	$300	Meets project requirements, Widely used by community	Closed onboard processor (but has well defined client interface)
Bebop, by Parrot [5]	$500	High quality camera and sensors, Open onboard processor	Does not meet project development toolchain requirement
Spiri, by Pleiades [6]	$1200	High quality camera and sensors, Open onboard processor	Does not meet project cost and availability requirements
Hummingbird, by AscTec [7]	$5000	High quality camera and sensors, Open onboard processor	Does not meet project cost requirement

Table 2

2. Development Tools Trade

The development environment and software library trades were combined to ensure compatibility. The primary objective of this study was to select a development environment supporting the integration of ROS, Open CV and the AR Drone SDK. Secondary objectives included community support, quick start and ease of use. Tables Table *3* andTable *4* below provide a summary of the development tools traded. The outcome of the study was the selection of the Qt Creator development environment. Due to lack of ROS support in Windows, the software development for the project will be done in Linux. A number of software libraries and development kits were included in the trade, the majority of these will be leveraged during the development of the Follow Me application.

IDE [9]	OS	Pros	Cons
Qt Creator [B]	Linux, Windows	Cross platform support, Multi-language support, Native support for CMake projects	
Visual Studio [C]	Windows	Well supported, mature toolchain	No Linux support, Limited capability in free version
EZ-Builder [D]	Windows	High level IDE for quick start, Well supported in community	No Linux support

Table 3

Library / SDK	OS	Pros	Cons
ROS [E]	Linux	Rich set of robotics libraries, Well supported in community	
Open CV [F]	Linux, Windows	Rich set of CV libraries, Multi-language support, Well supported in community	
AR.Drone SDK [G]	Linux, Windows	High and low level support of AR Drone specific features	
EZ-SDK [H]	Windows	High level IDE for quick start, Well supported in community	No Linux support
EZ-SDK Mono [I]	Linux	High level support for quick start	

Table 4

C. Proof of Concept

The primary goal of this task was to quickly prove out the ability to accomplish the overall project goal of mobile image tracking on the development platform. The Windows based EZ-Builder [C] application was leveraged for this task due to its inherent support for the AR Drone quadcopter. EZ-Builder is a high level development environment created by EZ-Robot [10]. The application is targeted to developers interested in programming and interacting with the EZ-Robot products, however it also contains libraries for third party robots and even supports low level libraries for DIY development. EZ-Robot has made a number of tutorials available to the developer who wishes to come up to speed quickly. One of such tutorials [11] discusses using the application to control the AR Drone to track an object based on its color. To conduct the proof-of-concept, the suggestions provided in this tutorial were used.

An EZ-Builder project was created that included an AR Drone Movement Panel (third party add on) and a Camera controller. The Camera controller was configured to command the drone to track objects with the color red. During the test, the EZ-Builder application communicated with the drone over the PCs WIFI network connection. The drone was commanded to Take-Off and then commanded to track the red object. At a high level, the tracking was controlled by the application as follows: The drone telemetered camera image and vehicle state information to the EZ-Builder application over its WIFI network. The application used the camera image as input into its control algorithms to generate acceleration commands that would allow the drone to maintain a desired orientation with respect to the red object. These acceleration commands were then passed back to the drone via the WIFI network.

The proof-of-concept was successfully accomplished in a relatively short timeframe. The AR Drone quadcopter demonstrated the ability to track a mobile object based on image processing performed real time on a remote device. A video of this test is available at the Github URL provided in section II.D above.

D. Follow Me Application Development

E. Future Work

IV. Results and Conclusions

V. Terminology

This section is included to provide the reader with a brief explanation of the terms used in this paper.

Term	Description
Client	Reference to remote application communicating with drone
COTS	Commercial Off-the-Shelf
DIY	Do it yourself
DOF	Degrees of Freedom
Drone	Reference to UAV
Host	Reference to system containing client software (may be used in place of client)
Open CV	Open source Computer Visualization software library
PC	Personal Computer
Quadcopter	Quadrotor UAV
ROS	Robotics Operating System
SDK	Software Development Kit
Target	Mobile user tracked by drone
UAV	Unmanned aerial vehicle

VI. References

ID	Reference
1	3D Robotics. "ArduPilot." Internet: http://ardupilot.com, [10/10/2015].
2	C. Anderson, J. Munoz. "3D Robotics, Inc." Internet: https://store.3drobotics.com, [10/10/2015].
3	DJI. "DJI Phantom 3 Series." Internet: http://www.dji.com/products/phantom-3-series, [10/10/2015].
4	Parrot. "AR Drone 2.0." Internet: http://ardrone2.parrot.com, [10/10/2015].
5	Parrot. "Bebop Drone." Internet: http://www.parrot.com/products/bebop-drone, [10/10/2015].
6	Pleiades. "Meet Spiri." Internet: http://pleiades.ca, [10/10/2015].
7	Ascending Technologies. "AscTec Hummingbird." Internet: http://www.asctec.de/en/uav-uas-drones-rpas-roav/asctec-hummingbird, [10/10/2015].
8	S. Piskorski, N. Brulez, P. Eline, F. D'Haeyer. *AR.Drone Developer Guide*. Dec. 2012.
9	Open Source Robotics Foundation, "ROS IDEs." Internet: http://wiki.ros.org/IDEs, [11/1/2015].
10	D.J. Sures. "EZ-Robot." Internet: http://www.ez-robot.com, [10/10/2015].
11	D.J. Sures. "Color Following AR Parrot Drone." Internet: http://www.instructables.com/id/Color-Following-AR-Parrot-Drone/?ALLSTEPS, [10/10/2015].
12	O.H. Boyers. "An Evaluation Of Detection and Recognition Algorithms To Implement Autonomous Target Tracking With A Quadrotor." B.S. Thesis, Rhodes University, South Africa, 2013.
13	S. Singh, B. Grocholsky, S. Nuske, M. Dille. "UAV / UGV Air-Ground Collaboration." Internet: http://www.frc.ri.cmu.edu/projects/agc/Overview.html, [10/31/2015].
14	D. Eggert. (2014, January). "Communicating with the Quadcopter." *objc.io*. [On-line]. 1(8). Available: https://www.objc.io/issues/8-quadcopter/communicating-with-the-quadcopter/ [Oct. 7, 2015].
15	
16	
17	
18	
19	
20	

VII. Resources

ID	Resource
A	Git, Linus Torvalds, https://git-scm.com.
B	Qt Creator, The Qt Company, http://www.qt.io/ide/.
C	Visual Studio, Microsoft, https://www.visualstudio.com.
D	EZ-Builder, EZ-Robot, https://www.ez-robot.com/EZ-Builder/windows.
E	ROS, Open Source Robotics Foundation, http://www.ros.org.
F	Open CV, itseez, http://opencv.org.
G	AR.Drone SDK, Parrot, http://developer.parrot.com/ardrone.html.
H	EZ-SDK, EZ-Robot, https://www.ez-robot.com/EZ-Builder/sdk.
I	EZ-SDK Mono, EZ-Robot, https://www.ez-robot.com/EZ-Builder/mono.
J	AR.FreeFlight, Parrot, https://play.google.com/store/apps/details?id=com.parrot.freeflight&hl=en.