Low Cost Unmanned Aircraft System for Autonomous Flight and Computer Vision Tasks

Undergraduate Students: David Hung, Coby Allred
Faculty Advisers: Dr. Michael W. Marcellin
Department of Electrical and Computer Engineering
The University of Arizona, Tucson, AZ 85721

ABSTRACT

The Arizona Autonomous Vehicles Club is participating in the 2016 AUVSI Student Unmanned Aerial Systems Competition which offers various challenges to be completed by a fully autonomous aerial vehicle. To complete various mission objectives, a low cost, rotary-wing platform was developed and deployed. The vehicle was assembled and upgraded for autonomous capability using commercially available components and open sourced software.

INTRODUCTION

Background

Held annually since 2002, the Association for Unmanned Vehicle Systems International Student Unmanned Aerial Systems (AUVSI SUAS) Competition brings university students together from around the world with the purpose of designing, integrating, and demonstrating their own unmanned aerial system (UAS). The aircraft must be capable of a variety of tasks ranging from fully autonomous flight to accurately discerning various ground targets and their characteristics. The competition aircraft can be made from either commercially available or custom components, but must still meet all competition requirements and safety regulations before being allowed to compete. While teams from the University of Arizona have previously competed in AUVSI SUAS, this is the first year the Arizona Autonomous Vehicles Club is attempting to do so.

To compete in the 2016 AUVSI SUAS competition, the University of Arizona Autonomous Vehicles Club created three sub-teams to ensure the aircraft was being developed as efficiently as possible. The hardware sub-team focused on designing and building the aircraft and were responsible for choosing low cost parts, assembly, and maintenance of the aircraft. Since a majority of the secondary tasks within the mission relied on computer vision, a second sub-team was assembled to focus specifically on completing these objectives. Finally, a software sub-team was created to focus on aircraft guidance, waypoint navigation, and integrating the data received from the aircraft to both our own ground station and the AUVSI SUAS interoperability server.

Problem Statement

Competing in the AUVSI SUAS competition requires an aircraft that is capable of a variety of different tasks; however most aircraft that are entered have been designed and built at a substantial cost, which drastically limits their widespread adoption. As the Arizona Autonomous Vehicles Club has not competed in years prior, it was determined that creating a low-cost alternative would be the best way to introduce club members to designing and competing with an aircraft that would be completed during the academic year prior to the 2016 AUVSI SUAS competition.

Objectives

During the competition, the UAS had to complete several primary tasks including fully autonomous flight, surveillance, target recognition, and localization. This year's platform was also designed to complete several additional secondary tasks given to teams including autonomous classification of targets and completing imaging of a target outside of the course boundaries. A detailed list of tasks and parameters are provided in the official AUVSI SUAS rules [1]. Finally, the UAS was designed to adhere to the strict competition guidelines for competing aircraft along with varied safety requirements.

DESIGN SOLUTIONS

Helicopter Platform

A gasoline powered helicopter was selected for this year's UAS platform. The helicopter is an attractive platform due to its vertical take-off and landing (VTOL) capabilities, dramatically increasing its robustness in rough terrain environments. Additionally, the energy density of gasoline allows for much greater operational range and airtime as compared to electrical platforms. Ultimately, a T-Rex 700N helicopter converted to gas using the Helix G700 gas conversion kit was used. After modification the final platform had a rotor diameter of 62.28 inches and weighed roughly 5.6 kilograms with a 660mL fuel tank. Under this configuration, the helicopter could achieve a flight time of 30 minutes without additional fuel tanks on the airframe at up to 80 mph cruise speed and 100 mph dash speed.

Figure 1: Model T-Rex 700N RC Helicopter. Image courtesy of AccuRC.com

Pixhawk Autopilot

A Pixhawk is used as the primary controller. The Pixhawk is an affordable, open source autopilot module that is widely used in the hobbyist community. The controller supports manual overrides, built-in failsafes, and easy integration with various peripherals. It operates on a 32 bit Cortex M4 core with 256KB of RAM, 2MB of flash memory, and a secondary 32 bit failsafe co-processor. The Pixhawk also includes a variety of sensors including a 16 bit gyroscope, a 14 bit accelerometer and magnetometer, a 3-axis accelerometer and gyroscope, and a barometer.

Manual overrides can be directly actuated from the backup 2.4 GHz radio controller, while the ground station is used to communicate mission waypoints and actions. Commands and telemetry data are transmitted between the ground station and the Pixhawk over a 915 MHz radio link. Metadata such as GPS location, altitude, and attitude are also transmitted over USB to the onboard computer (OBC) for image processing purposes. A Raspberry Pi 2 Model B was chosen as the OBC due to its low price, small form factor, and attractive technical specifications.

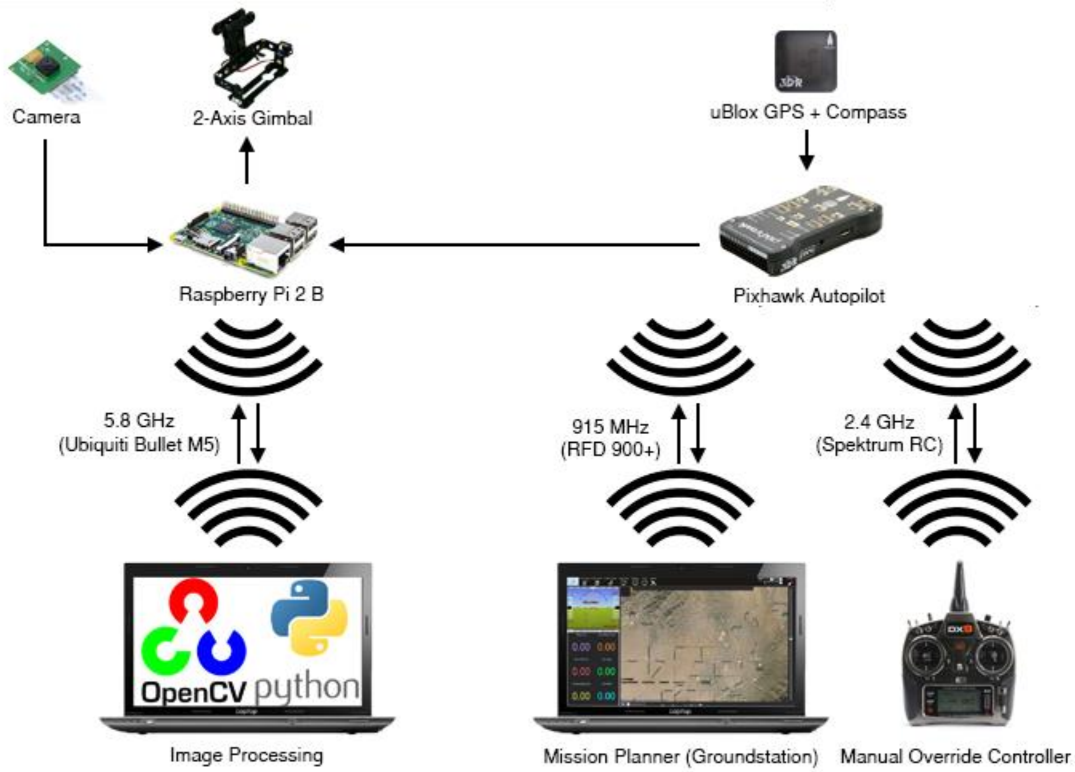

Figure 1: Avionics and ground systems overview

Mission Planner is run on a laptop PC for ground station operations. Mission Planner is an open source ground station application targeted at the Pixhawk. The software is used for initial configuration of the Pixhawk as well as for further control loop tuning. Sensor data logs generated by the Pixhawk are viewed in Mission Planner, allowing for diagnostics and trimming of the airframe. Additionally, the software includes a GUI for mission planning, with options ranging from GPS and altitude waypoint plotting to special tasks such as payload drops and image capture. Mission Planner also includes a live flight data panel, which interactively displays live telemetry data.

Imaging System

A 5-megapixel Raspberry Pi Camera is used for image capture. While the camera's poor resolution and slow shutter speed make it an unsuitable option for a final configuration, its plug-and-play compatibility with the OBC made it an acceptable choice for prototyping. In order to mitigate the effects of vehicle vibrations and dynamic vehicle attitudes, the camera is mounted on a vibration dampened 2-axis ground facing gimbal. Software settings are used to maximize shutter speed in an effort to reduce motion blur. Images are captured at 1 Hz and are synchronously tagged with metadata received from the Pixhawk controller. The tagged images are then pushed to the target recognition buffer, where images are first preprocessed by the OBC then transmitted to the ground using a 5.8 GHz radio link.

Target Recognition and Analysis

OpenCV is used to identify possible targets in images captured by the imaging system in real time. Each region of interest (ROI), or area of the image which contains a possible target, is cropped and transmitted to the ground station for further analysis. OpenCV is once again used on the ground station in order to confirm target validity and to determine the target's GPS shape, color, embedded alphanumeric color, alphanumeric letter, and alphanumeric orientation.

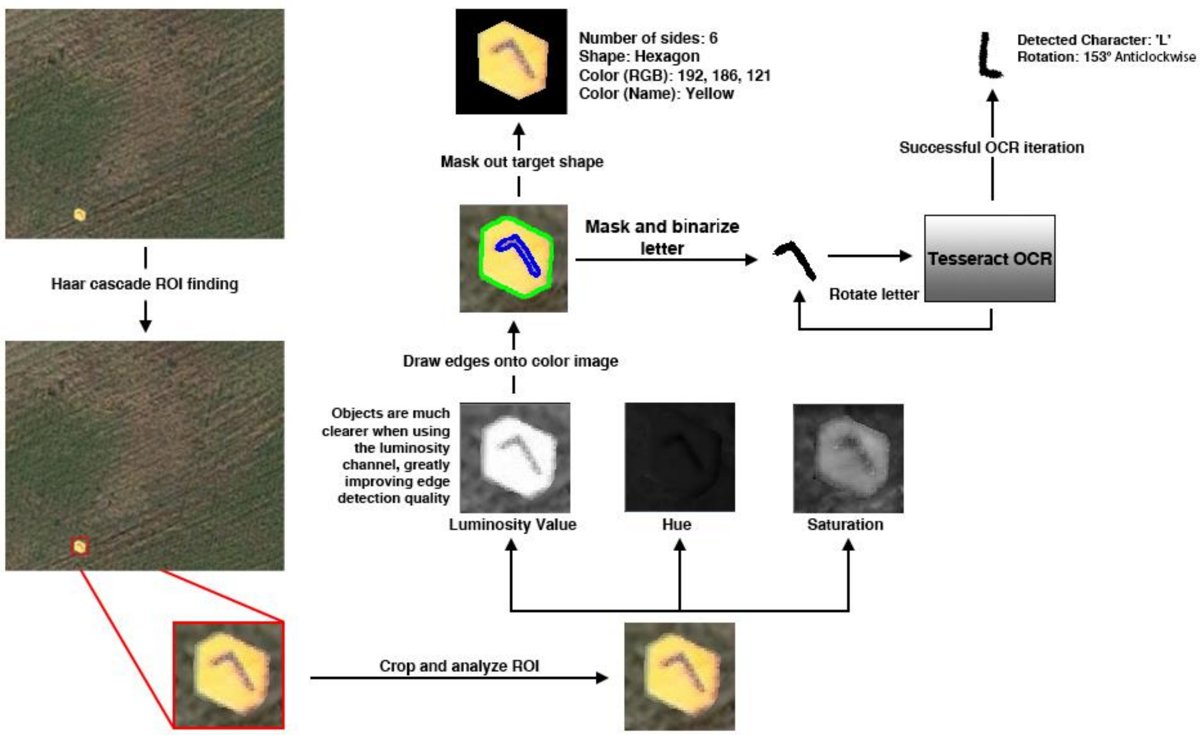

Figure 2: Image processing system overview. All processes shown above are implemented using OpenCV

The OpenCV target identification is performed onboard using a Haar-like feature cascade classifier. The technique, first used in real-time face recognition in 2001 by Paul Viola and Michael Jones [2], is an ROI detection algorithm which searches for the presence of objects of interest in an image. The approach uses a sliding window approach to scan for key features which indicate object presence. The scanning process is broken down into cascading stages, dramatically reducing the algorithm's computational expense. Detected ROIs are cropped and further geotagged using the uncropped image's geotag, UAS altitude, and ROI image coordinates All other metadata is inherited from the uncropped image. Cropped images are transmitted to the ground station via FTP. Duplicate records of each target are aggregated based on relative GPS distance.

Further image processing is performed using a combination of standard and in-house OpenCV algorithms. Extracted ROIs are transformed into their Hue-Saturation-Value (HSV) representation. The HSV representation is split by channel, and the luminosity (value) channel is processed using the Canny edge detection algorithm [3]. In this implementation of the algorithm, a Gaussian blur is applied to the image, the intensity gradient is found using the Sobel operator [4], and the edges are refined by determining “strong” edges and identifying connected edges. It was found that the algorithm still resulted in fragmented edges, thus a custom algorithm was written to combine edges based on their relative distance, reducing the fragmented edges to a larger outline and a smaller embedded alphanumeric outline. The two outlines are extracted and processed in parallel to determine the target’s shape, color, and embedded alphanumeric character.

In order to determine the target’s shape and color, the target outline is approximated to a polygon using the Douglas-Peucker algorithm [5]. The number of edges in the resulting polygon is calculated and is compared to a table of polygon edge counts and names. If no matches are found, the target is labeled as an *n-gon,* where *n* is the number of sides. The approximated polygon is then used to mask the original image, reducing the image to the isolated target. The RGB values of the masked image are averaged in order to determine the target’s average RGB color. The resulting color tuple is compared to a table of common colors, and the closest match within acceptable error range is used to label the target. If no suitable matches are found, then the target is labeled using the color tuple instead.

The embedded alphanumeric character outline is used to determine three characteristics: letter orientation, alphanumeric color, and alphanumeric character. Color is determined using the same method outlined above for target color identification. Optical Character Recognition (OCR) is used to determine the alphanumeric character embedded in the target. Tesseract, an open source OCR engine [6], was chosen for this task due to its availability, high accuracy, and ease of integration. However, images must be carefully preprocessed in order to fully take advantage of Tesseract’s features. Namely, Tesseract performs best with binary images and little skew. In order to meet these requirements, the extracted character outline is filled and redrawn on a blank background, yielding a binary image. Plausible character orientations are determined by exploiting anticipated character aspect ratios and are used to deskew characters. The deskewed characters are inputted into Tesseract, and each target is labeled based on the letter output by Tesseract. In the event that the plausible character orientations fail to produce a confident letter output, the character is rotated in increments of 10 degrees until a letter is recognized. Upon successful character recognition, the degree rotation and image metadata are used to determine the alphanumeric cardinal orientation.

Communications Systems

The primary communications link is the 915 MHz telemetry link. The telemetry link uses the RFD 900+, a small and lightweight modem which can easily achieve ranges of more than 20 miles, far exceeding the mission requirements. The RFD 900+ modem offers a rate of transfer of up to 250kbps and operates at a maximum output power of 1 watt. An additional 5 GHz link is used for transmitting images. The secondary link is implemented using the Ubiquiti Bullet M5-HP radio, offering a 5GHz link and a throughput of 100+ Mbps while operating at a maximum of 6 watts, though the range is limited compared to the 915MHz telemetry link. In order to mitigate the loss of range, a directional antenna is used at the ground station. Lastly, a 2.4 GHz link is operated by a safety pilot and is used to switch between flight modes and execute manual overrides as deemed necessary.

RESULTS AND CONCLUSION

Unfortunately, logistical challenges and complexities introduced by the helicopter platform prevented us from completing the UAS. However, we were able to achieve proof of concepts for each component with a small team and a limited budget. These results indicate the potential of a UAS built solely from affordable, off the shelf components.

Although the team was unable to compete this year, the team still attended the competition as observers and learned many valuable lessons for subsequent years. Based on observations of the various approaches to the competition and discussions with other teams, it was decided to shift design towards a fixed wing platform in order to leverage the resources available at the mission site and reduce design complexity.

ACKNOWLEDGEMENTS

The Arizona Autonomous Vehicles Club would like to thank our faculty advisor **Professor Michael Marcellin** for his endless support, patience, and guidance. Additionally, we would like to thank all of our team members for their hard work, especially **Mark Omo**. Lastly, we would like to thank the University of Arizona for their support.

REFERENCES

1. "2016 Rules for the 14th Annual Student UAS (SUAS) Competition." *AUVSI SUAS Competition*. AUVSI Seafarer Chapter, 08 June 2016. Web. <http://www.auvsi-suas.org/static/competitions/2016/auvsi_suas-2016-rules.pdf>.
2. Viola, P., and M. Jones. *Rapid object detection using a boosted cascade of simple features*, vol. 1, IEEE, 2001.
3. Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6), 679-698.
4. Sobel, I. (2014). History and definition of the sobel operator. *Retrieved from the Web*.
5. Ramer, Urs. "An iterative procedure for the polygonal approximation of plane curves." *Computer Graphics and Image Processing*, vol. 1, Elsevier B.V, 1972.
6. Smith, R. *An Overview of the Tesseract OCR Engine*, vol. 2, IEEE, 2007.