MACHINE VISION AND AUTONOMOUS INTEGRATION FOR AN UNMANNED AIRCRAFT SYSTEM

THE UNIVERSITY OF ARIZONA AERIAL ROBOTICS CLUB

Undergraduate Student Team: James Dianics, Malcolm Gibson, Hans Hony, Jun Li, Elliott Liggett, Michael Palmer, Christopher Poole, James Powell, Joshua Tolliver, Dimitri Ververelli

Faculty Advisor: Prof. Dr. Hermann Fasel

1 INTRODUCTION

1.1 The UA Aerial Robotics Club and SUAS Competition

The Aerial Robotics Club considered the annual AUVSI Student Unmanned Aerial Systems Competition in Fall of 2010 and decided to make this competition the number one priority for generating new student interest, developing new aerial vehicles, and increasing the level of excellence sought by the club. In order to assemble a competition team to design, construct, and test a vehicle of high enough caliper for this competition, the club officers selected two sub-teams. The first sub-team is composed of six aerospace engineering senior students who were responsible for designing and building a complete composite flight vehicle (AVATAR – Aerial Vehicle for Autonomous Target Acquisition and Recognition). The second sub-team is composed of two electrical engineering students, one optical sciences and engineering student, one systems engineering student, and one aerospace/mechanical engineering student. This sub-team was responsible for developing a comprehensive avionics system (LAARK – Low-Altitude Aerial Reconnaissance Kit) that meets all the performance requirements of the 2011 AUVSI SUAS Competition and more within ten months.

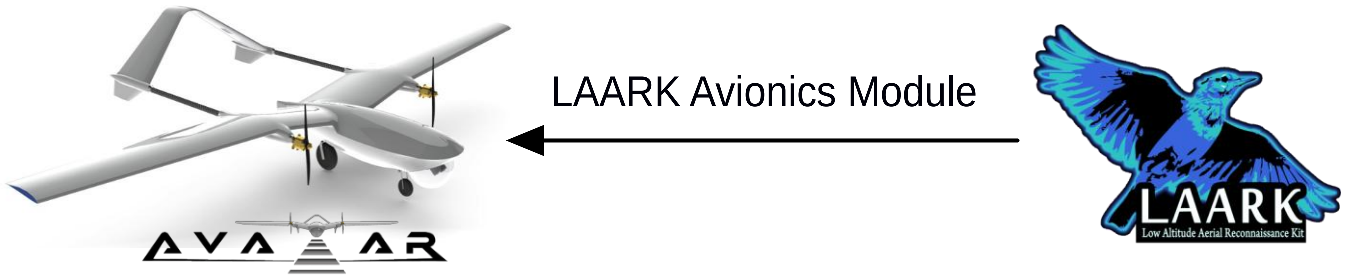


Figure 1: Demonstration of Collaboration between LAARK and AVATAR

1.2 Mission Requirements

The competition rules provided a product simulation scenario for which competing teams had to perform. The challenge was required to have several hard to meet characteristics that effectively simulate the challenge of producing a real-world UAS for a military or other application. These rigorous rules provided production and performance requirements that the teams had to meet under the production scheduled timelines.

- Autonomous flight
- Autonomous target recognition
    - Alphanumeric character and color
    - Target shape and color
    - GPS location and orientation

(Full list of system requirements can be found in the University of Arizona's journal submission to the 2011 AUVSI SUAS competition)

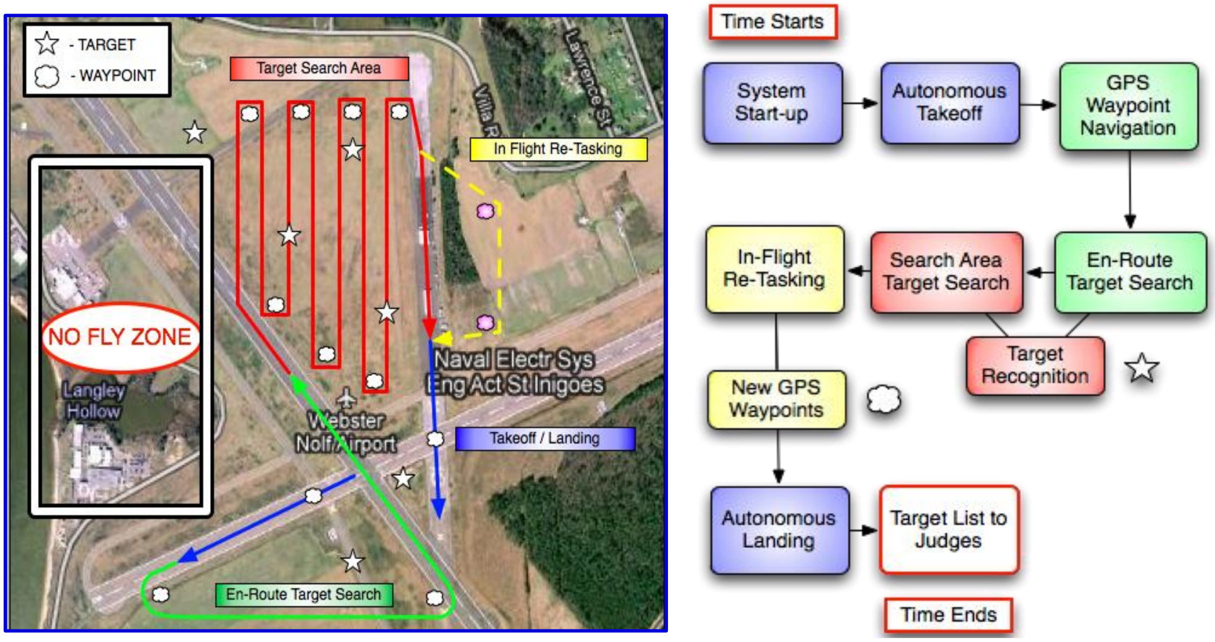


Figure 2: Conceptual Mission Layout and Flow Diagram

1.3 Brief System Overview

The University of Arizona's UAS is composed of two main systems, the aircraft (AVATAR) which includes the airframe, propulsion system, and flight control system, and the avionics (LAARK) which includes the onboard camera system, autopilot, air-to-ground communications, and ground station systems. The airframe was specifically designed for the 2011 SUAS competition and was constructed using modern composite materials in order to carry the mission payload of up to 8 lbs. for 40 minutes. The mission payload (LAARK) developed for the SUAS competition consists of a Piccolo II autopilot system with DGPS for autonomous flight navigation, takeoff, and landing, and an aerodynamically encased gimbaled camera system with dual 10 MPixel cameras capable of capturing up to 3 frames per second. The dual camera system provides a combined 120 degree horizontal FOV and is directed vertically towards the ground while the aircraft maneuvers. This is accomplished using a Trossen Robotics pan/tilt gimbal with a MosquitIO gimbal microcontroller. The imaging system is controlled by an onboard FitPC2 computer that streams imagery and telemetry data to the ground station over optimized 2.4 GHz Wi-Fi. The ground station employs a software suite of image processing software (OpenCV and Matlab) that corrects for image distortion and performs autonomous target recognition and analysis. The entire system is monitored using a network of three graphical user interfaces to control flight navigation, monitor telemetry data, display ground imagery, and present target reconnaissance results.

2 AIRCRAFT DESIGN AND OVERVIEW

The AVATAR aircraft (Figure 3) was designed to meet the requirements of the 2011 SUAS competition and to provide a stable imaging platform for the LAARK avionics module. The aircraft design was based upon the results of trade-off analyses performed for the configuration, powerplant, and control system. The aircraft employs a twin-boom configuration (with 10ft. wingspan) with an inverted V-tail and tricycle landing gear. Two brushless electric motors powered by five 5-cell lithium polymer (LiPo) batteries provide the necessary thrust and are located on the forward ends of each boom. The aircraft is controlled using ailerons for roll control, ruddervators for pitch and yaw control, and differential thrust between the twin tractor propellers for additional yaw control when needed. Inboard flaps and flaperons can easily be implemented with the current configuration. The aircraft has an empty weight of 15 lbs. and a maximum takeoff weight of 30 lbs. with 7 lbs. reserved for batteries and 8 lbs. for payload. With this configuration, the aircraft is able to achieve a flight time of 40 minutes. The aircraft has a takeoff speed of 25 knots, a cruise speed of 45 knots, and a dash speed of 60 knots. The aircraft can be disassembled into 5 sections for easy transportation and has a quick assembly time of 2 minutes by two individuals.

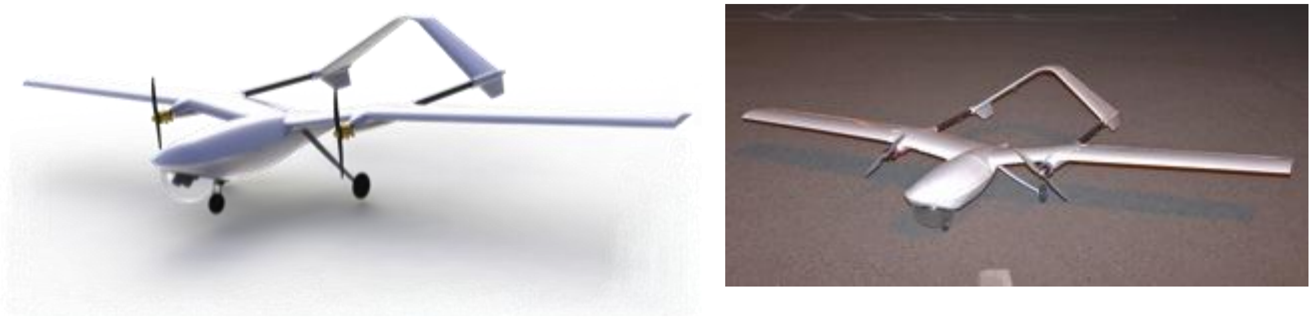


Figure 3: The AVATAR Airframe Pictured in CAD (left) and on the Runway (Right)

3 ONBOARD AVIONICS DESIGN AND OVERVIEW

The AVATAR aircraft was designed for optimal integration with the LAARK avionics package. The LAARK system configuration (Figure 8) was designed for ultimate performance. The design features powerful, cutting edge technology components that achieve all of the functional requirements set forth in the CONOPS documentation.

3.2 Piccolo II Autopilot and Telemetry System

The Piccolo II autopilot subsystem is an industry standard and consists of the onboard Piccolo II autopilot, the Ground Station and the Piccolo Command Center. Waypoint navigation, override controls, and flight sensory data is commanded via the Piccolo ground station. Transmission is sent and received over 900MHz channel to the onboard autopilot via the ground station. The onboard autopilot module connects via RS-232 (serial port) to the aircraft's onboard computer, the FitPC2. A custom autopilot data parser (“`autopilot`”) runs on the FitPC2. This program is run upon the kernel detection of the autopilot's RS-232 to USB adaptor, and is terminated upon device removal, via a `udev` script that identifies adaptor serial number and manufacturer.

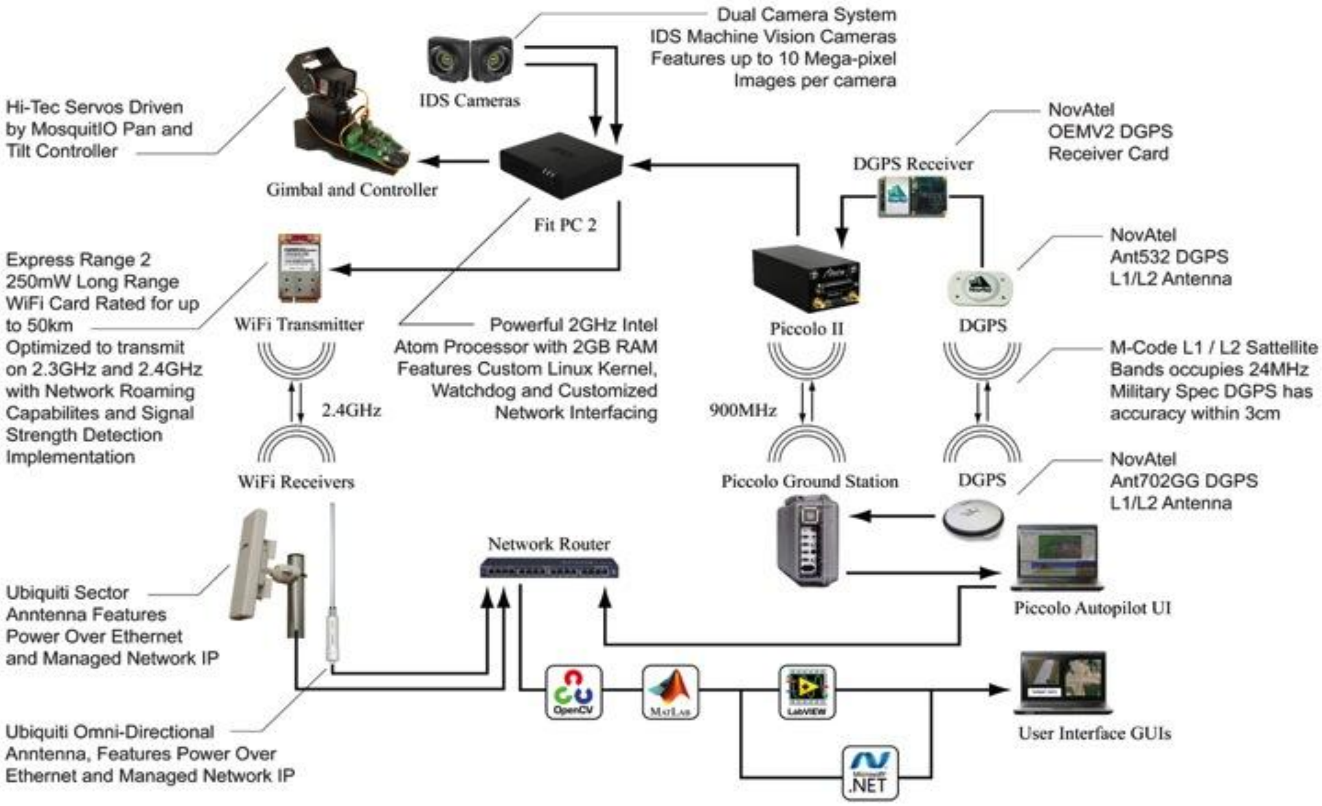


Figure 6: LAARK Avionics System Components Overview

The programming for the Piccolo system is C-based, and the flight data can be requested from the onboard aircraft computer via the UserData class public member assessors that are detailed in the Piccolo documentation. Among these data are the GPS location, flight altitude, heading, pitch, roll and yaw. These data are made available to the gimbal controller and camera client systems via a Linux kernel-managed shared memory segment. Shared memory segments were chosen because they are non-blocking and incredibly fast. The non-blocking aspect was of particular importance. In the event of a program crash, a typical socket or pipe connection must be closed properly before another task can utilize that connection. With a shared memory segment, the kernel simply manages this memory segment as it does any other memory segment in its process tree, handling multiple concurrent requests with ease and efficiency. Since the shared memory segments are ‘keyed’ to a specific file’s inode (the inode is used to synthesize identical memory addresses for each sharing process), shared memory segments persist after program termination and thus are resistant to program crashes.

3.3 Gimbaled Camera System

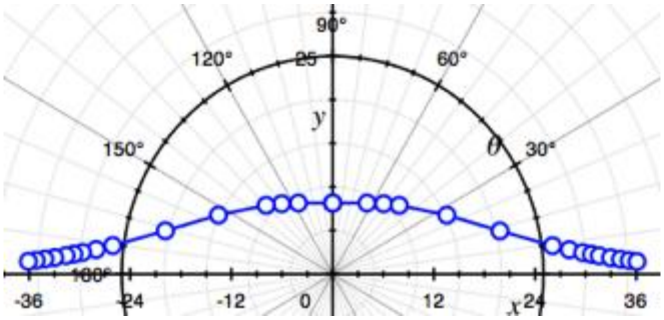
The onboard camera system features two IDS uEye LE Machine Vision cameras (UI-1495LE-C) equipped with Edmund Optics Tech-Spec 4.5mm fixed focal length lenses. The board level cameras have a resolution of 10 mega-pixels each and are controlled using Linux drivers running on the FitPC2 onboard computer’s Linux platform. The cameras are positioned at an angle optimized to provide a 120° horizontal field of view with minimal image distortion.

The cameras are mounted onto a pan-tilt gimbal that is controlled by an external *MosquitIO* (ATMega168 board) gimbal micro-controller that receives aircraft Euler attitude data from the aircraft’s FitPC2 via a gimbal control program (`gimbalctl`). The gimbal control program is written in ANSI-C and is automatically run whenever the Linux kernel recognizes the RS-232 connection to the gimbal microcontroller. This detection takes place via custom `udev` scripts that match manufacturer and serial number of the USB devices. If at any time the gimbal device disappears, the gimbal controller program is terminated until the device reappears.

Aircraft attitude information is received from the autopilot data parser via a shared memory segment using a custom data struct. The gimbal controller first checks the status member of the data struct to see if the segment is currently being updated. This is a custom handshake developed to prevent using a mixture of past and current telemetry information in the event of concurrent access. The gimbal functions to ensure that the camera system remains orthogonal to the ground and the field of view is never compromised. Images are immediately sent via 802.11n to the ground station, and relevant flight sensory data are also sent to ground and air-based MySQL databases.

3.5 Air to Ground Communications

Figure 8: Ideal Antenna Pattern for Aircraft



The camera to ground subsystem consists of an optimized onboard Wi-Fi data-link between the FitPC2 and the 2 ground station access points, the ground network, Labview (human interface), the mirror image folder (on the ground), and the mirror MySQL database (on the ground) of collected image data. The Wi-Fi data-link utilizes a custom multithreaded client-server program to push images and data to the ground for processing using the industry standard TCP/IP protocol. Additionally, the aircraft populates ground and air-based MySQL databases connecting imagery, telemetry, and target results together in one place.

The onboard antenna was specially designed to maximize gain at low radiation angles (when the plane is furthest away). It was constructed using a 12-cm radius aluminum plate (1mm thick) with an N-type coaxial connector placed in the center and a 3cm 12-AWG wire connected to the center pin as the vertical radiator. For the ground station, a “keyhole” pattern was constructed in order to cover the far end of the field and the takeoff area. This was done using a Ubiquiti Wireless Bullet M2 with a Comet SF-245W vertical antenna as the omnidirectional access point, and a Ubiquiti Nanostation M2 as the sector access point. To facilitate roaming between these two access points (AP), a custom-roaming algorithm was devised. First, the "`iw`" Linux wireless command was modified to output wireless scan data in easily parsable column format. Data from this scan is then placed into a shell script that looks for the highest signal matching a set of rules (frequency and SSID). If the strongest signal is currently not associated with, the script will compare the current RSSI with the RSSI of the strongest received AP. If the difference is greater than a set threshold, a handoff is initiated. Should the handoff fail, the aircraft will revert to the previously used AP. Any decisions made by this program are logged and may be studied post mission.Another critical aspect of Wi-Fi equipment is band crowding. Band crowding occurs when multiple users and devices attempt to share a given spectrum. On 802.11, these devices may include baby monitors, video surveillance equipment, and computer access points. Problems that may be encountered include increased link delay, loss of signal, and diminished bandwidth. To alleviate these concerns, the Linux Wi-Fi subsystem, kernel, and regulatory domain enforcement was modified to allow for a frequency-agile Wi-Fi system that spans 2.38 – 2.42 GHz.

3.4 Image Client-Server Application

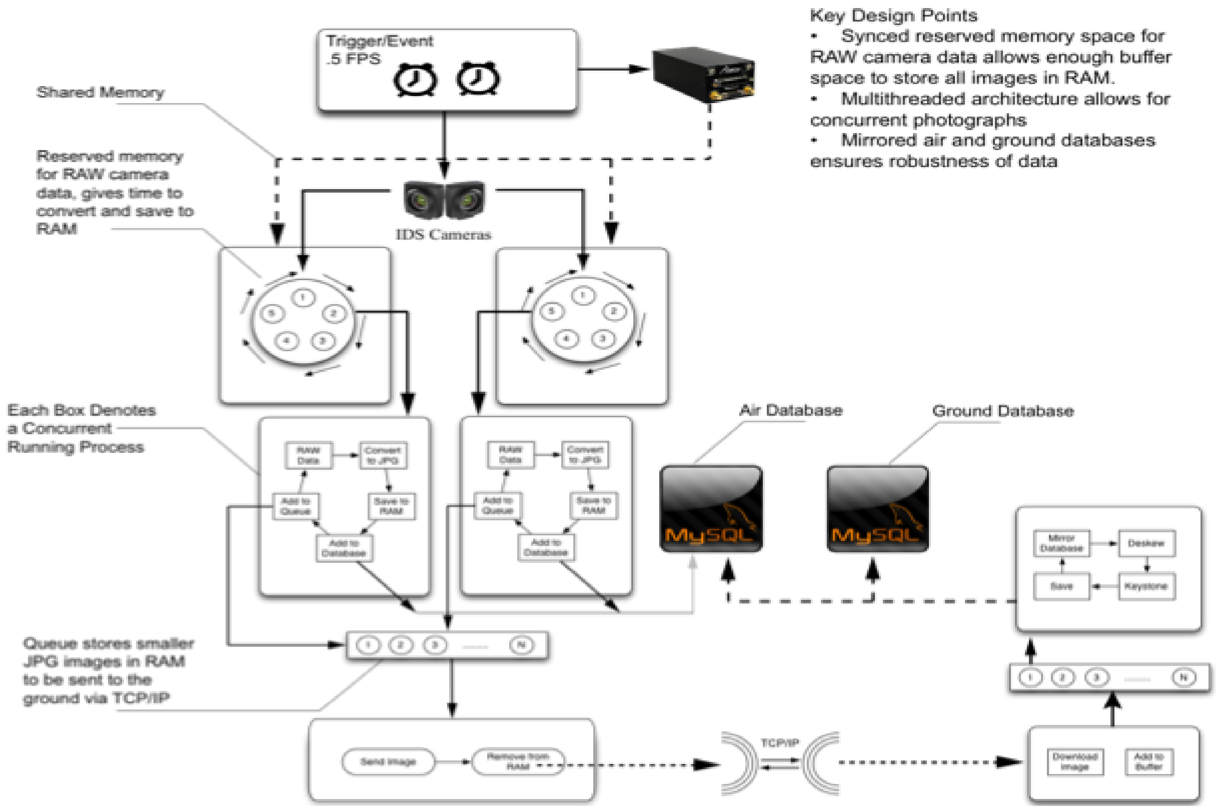


Figure 7: Image Client-Server Model

The client-server application (Figure 9) is a crucial link in the overall system architecture because it controls and interacts with many sub-components. The client is controlling image capture, synced with autopilot data, and controlling this data flow into the ground station. The client system relies on the capabilities of many other components just as other components rely on it; for example images must be obtained at such a rate that all of the search area is covered at the given speed of the aircraft.

4 GROUND SYSTEMS DESIGN AND OVERVIEW

Figure 10: The Piccolo Command Center (PCC) Graphical User Interface

The LAARK ground station is a network of computers that are used to process the flow of imagery and telemetry data received from the aircraft and display it to the ground station operators in a professionally organized manner. The raw imagery received from the aircraft is stored in an image directory that is accessible by all network computers. The telemetry data and image data are stored in a MySQL database with tables for each camera, autopilot telemetry, and target recognition results. The raw imagery is immediately processed using OpenCV to remove image distortions. The corrected images are then stored in a corrected image directory where Matlab can read images and process them for target recognition and analysis. These ground station processes are all monitored and controlled by two graphical user interfaces, the Sensor Operator View (SOV) and the Intelligence Officer View (IOV).

The other component of the ground station is the Piccolo Flight Command Center (Figure 10) and ground station. Aircraft navigation is controlled by the proprietary Piccolo Command Center (PCC) from CloudCap Technology and this allows the pilot operator to plan and execute flight plans. The pilot may also re-task the aircraft to other mission objectives and monitor all available aircraft sensors. The controller attached to the ground station interface allows the pilot to manually override the autopilot in case of emergency. The PCC interface displays the aircraft's current position over USGS maps in both 2D overhead view and 3D elevation relief view.

4.2 Ground Station Graphical User Interfaces

The sensor operator view (SOV) is written in Microsoft .NET framework and displays aircraft sensor information as well as the telemetry data associated with each image. This offers a time stamp view of the raw images before they enter into the barrel and keystone correction processes. The SOV displays the targets that were detected in the machine vision process by interfacing with the ground MySQL database.

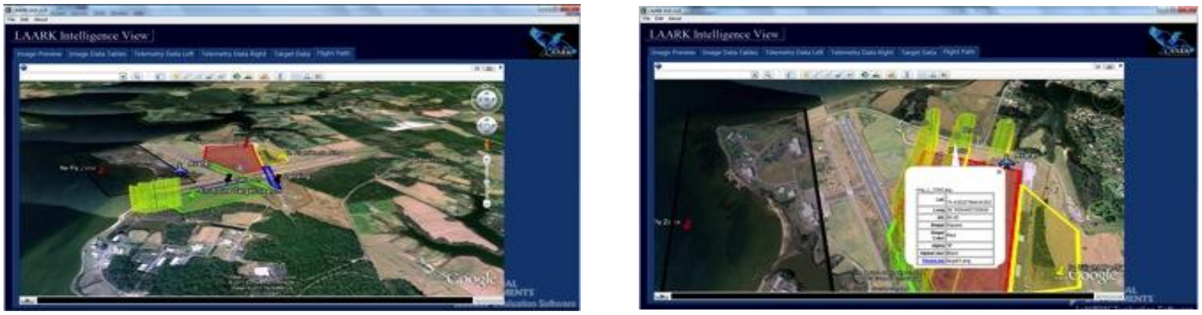


Figure 11: LAARK Ground Station User Interfaces. Sensor Operator View (SOV) (top left), IMAQ Image Analysis Control (top right), Intelligence Officer View (IOV) showing live flight plan view (bottom left), and Real Time Target Info Display on Map (bottom right).

4.3 Distortion Correction in OpenCV

Figure 12: Original Image, Corrected Barrel Distortion, and Corrected Keystone Distortion.

When images are initially received by the ground station they must be processed before accurate machine vision and target recognition can take place. This processing is necessary for two main reasons. One, the images are taken through wide-angle lenses that suffer from severe barrel distortion (fish-eye effect). Two, the cameras are angled 70 degrees outward from the ground's normal vector in order to achieve a combined 120 degree field of view. This angle results in a trapezoidal imaging plane that must be corrected by applying keystone correction.

Intel's OpenCV is an open source collection of image processing functions that can be used for a variety of image processing applications. Image distortion can be measured by calibrating each camera using a checkerboard template. Distortion coefficients are determined and used to apply corrective manipulations that counter-act the distortion inherent in the optical system. This results in a final image that has constant spatial resolution and is representative of the real physical area. Figure 12 shows each stage of this process.

4.4 Target Recognition and Analysis in MatLab

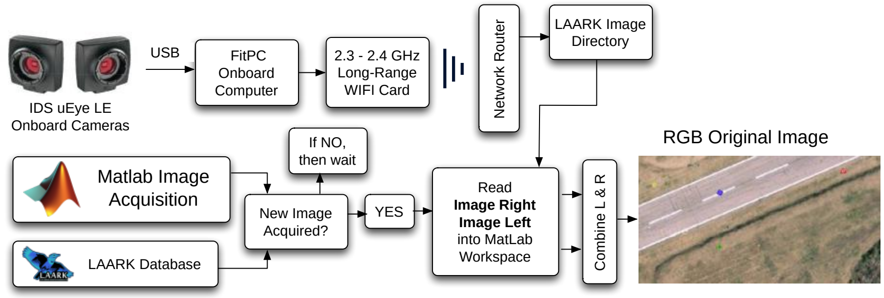


Figure 13: (left) Image Acquisition Process into the Matlab Workspace for Target Identification

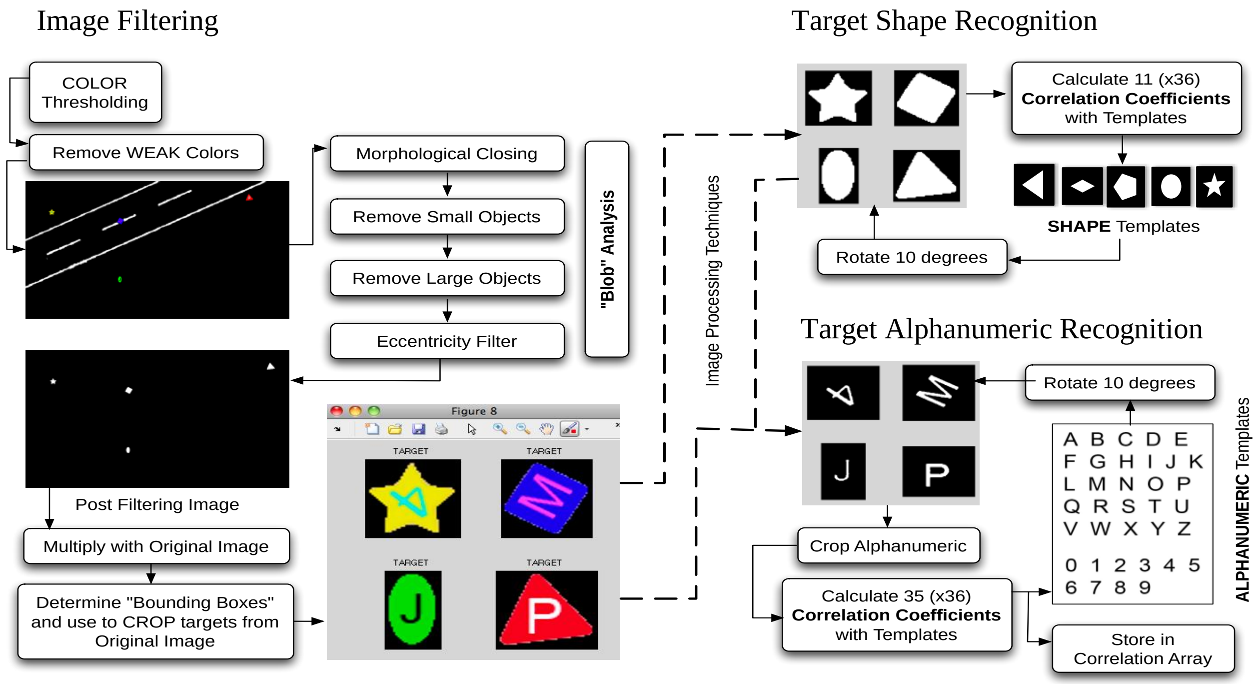


Figure 14: (above) Target Analysis Processes. Image Filtering, Shape and Alphanumeric Recognition

The MatLab Image Processing Toolbox (IPT) is used to process the undistorted aerial images and autonomously identify ground targets between four and eight feet in any direction coplanar with the ground. Matlab processes two images at once, one from each camera taken at nearly the same instant in time. If targets are identified, Matlab crops the target image, saves it to a target directory, and performs in-depth analysis to determine its approximate GPS location, orientation, shape, colors, and alphanumeric characters. The results of the analysis are written into the LAARK MySQL ground database (using Matlab's Database Toolbox) for access by the ground station user interfaces discussed previously. The image acquisition process is outlined by Figure 13.

The Matlab Target Recognition Software starts the process by periodically checking the MySQL database for new imagery data. If a new image has been received, the corrected image and time-synced telemetry data (GPS, Euler Angles, etc.) are read into the Matlab workspace for analysis. The target identification process involves a series of filtering techniques to isolate targets from the background image by exploiting key target features such as size, color intensity, and eccentricity. This process is outlined in Figure 14. Once a target has been identified, it is cropped from the original image and sent through a series of analysis techniques to determine its shape, colors, location, orientation, and alphanumericcharacter. Shape and character recognition is achieved by applying a series of image manipulations to obtain a binary image of only the shape and only the alphanumeric. These new images are then compared to a database of 11 shapes and 35 characters by calculating a maximum correlation coefficient. The target colors are determined by finding the average RGB values of both the alphanumeric and shape. The GPS location of each target is approximated by considering the time-synced GPS location of the aircraft, the pixel resolution to geospatial resolution relationship (function of altitude), the time-synced heading of theaircraft, and the pixel centroid of the target in the original image.

5 CONCLUSION

The rigorous work of AVATAR and LAARK took nine months to design, construct, and test a complete unmanned aerial system for competition in the 2011 AUVSI Student Unmanned Aerial Systems Competition from June 15 – 19th 2011. ARC is quite pleased to have met the challenge having received 8th place and been awarded within the Top 10 near many teams which have iteratively designed aircraft for four consecutive years or more. To be located in such a placement deems that the University of Arizona rose to meet the challenges of this project.

This project represents one of the most comprehensive and advanced unmanned aerial system ever developed by University of Arizona students. Future students who compete by extending this project will have to contend with many challenges this AUVSI competition offers. With the LAARK and AVATAR platforms at their availability, many of these challenges will be system engineering challenges with emphasis on compacting into onboard processing in order to streamline the data acquired.

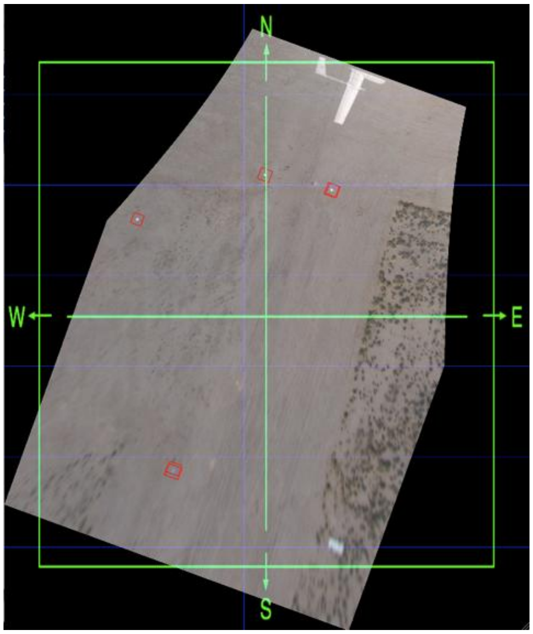


Figure 15: (at left) Machine Vision Generated HUD for purposes of GPS Approximations. Regions of Interest in Red are Detected Autonomously. Data Collected During Preliminary Flight Testing in Tucson, Arizona

6 ACKNOWLEDGEMENTS

The University of Arizona Aerial Robotics Club would like to acknowledge and thank the following groups and individuals for their support and guidance over the course of this project.

Professor Dr. Hermann Fasel - Faculty Advisor, Department of Aerospace and Mechanical Engineering, UA; **Mark Ballinger -** Technical Advisor, Chief Pilot for Unmanned Aerial Systems, Raytheon Missile Systems; **Anton Kochevar -** Technical Assistance, Flight Operations, Raytheon Missile Systems; **David Henz -** UA Engineering Avionics Team Mentor, Product Line Manager, Edmund Optics; **Professor Dr. Ricardo Sanfelice -** Faculty Mentor, Department of Aerospace and Mechanical Engineering, UA; **Professor Dr. Michael Marcellin -** Faculty Mentor, Department of Electrical and Computer Engineering, UA**;** **Bob Perry -** Technical Assistance, Science Applications International Corporation

Additionally we would like to thank: **AUVSI**, **NovAtel**, **Raytheon**, **jjPlus**, **Cloud Cap**, **Brock Technologies**, **Thomas Reprographics**, and **University of Arizona** for donations and guidance.