Technical Paper, AUVSI UAS Competition
University of California at San Diego

Faculty Advisor: Dr. John Kosmatka

Karthik Balakrishnan, Thomas Hong, Øyvind Stamnes,
Kyle Aguilera, Sascha Ishikawa, Shane Grant,
Brent Taylor, Adrian Davalos, Tim Palmer,
Grace Bohn, Mike Pattanachinda, Kevin Groves

June 2007

auvsi@ucsd.edu

Abstract

This paper provides an overview of UCSD's AUVSI aircraft designed to meet the requirements laid forth by the judges of the 2007 AUVSI Student UAS mission. The aircraft, an RnR APV-3, is controlled in the air (after takeoff until landing) by a Crossbow μNav autopilot. While in the air, the aircraft first flies through a designated set of waypoints, then continues on to search a field where it captures and analyze images for potential targets using a camera and on-board computer system. While the systems goals are complex, we have used commercial off-the-shelf products allowing for both low cost and high degree of flexibility and simplicity. Safety is also a priority; the aircraft can hand control to an R/C pilot either through a software command or a hard switch, and a hard-over and engine kill can be conducted if necessary.

Contents

1 Airframe

1.1 Vehicle Introduction

The AUVSI UAS competition calls for an unmanned aerial vehicle that is capable of gathering images of a search area while flying through specified waypoints under autonomous control. Given the mission requirements, the vehicle must have stable flight characteristics in order to improve the performance of the autopilot and a large internal volume to carry the autopilot, camera, and onboard computer systems necessary to complete the mission. In addition, an airframe with a large wing loading will enhance image processing as wind gusts will be less likely to bounce the plane around and cause image distortion when compared to a lightly loaded r/c trainer.

1.2 RnR APV-3 Overview

Given the constraints laid out in section 1.1, we have selected an RnR Products APV-3 airframe (figure 1). This is the same model that was flown in the 2006 competition, and we now have a second aircraft serving as a backup. The primary airframe is powered by a Fuji Imvac BT-64A engine (5.7 HP) providing 35lbs of thrust while the backup uses a larger Fuji Imvac BT-86 engine (7HP) which provides 40lbs of thrust. We chose the BT-64A for the primary airframe because it is less sensitive to throttle inputs, the result of lessons learned during last year's competition.

Both airframes have a 12.4 foot wing span, are 6.4 feet long, and are 2.8 feet tall at the top of the t-tail. The payload tray attaches underneath the fuselage, has an internal volume of 1.5 cubic feet, and is interchangeable between the two aircraft. In addition, the APV-3 can hold up to 2 gallons of fuel, allowing for a flight time of up to 8 hours while cruising at 55 mph; for this competition, the aircraft will be outfitted with two 50 oz fuel tanks giving a flight time of up to 1.6 hours at cruise.

Figure 1: The two RnR APV-3s

1.3 RnR APV-3 Technical Specifications

	Primary	Backup
Wingspan	12.4 ft.	12.4 ft.
Height	2.8 ft.	2.8 ft.
Wing area	10.5 ft.	10.5 ft.
Aspect ratio	14.7	14.7
Empty Weight	28 lbs	30 lbs
Powerplant	Fuji BT-64A	Fuji BT-86
Thrust	35 lbs	40 lbs
Maximum speed	80 mph	110mph
Range	85 miles	85 miles
Wing loading (empty)	42.6 oz/ sq. ft.	45.7 oz/ sq. ft.
Thrust / weight (empty)	1.25	1.3

1.4 Fabrication

Donations of the fuselage, nose gear, and hatch covers by RnR Products left the fabrication of the flying surfaces to the students. Our method of vacuum bagging composites over foam cores was used and several improvements were incorporated into the process this year.

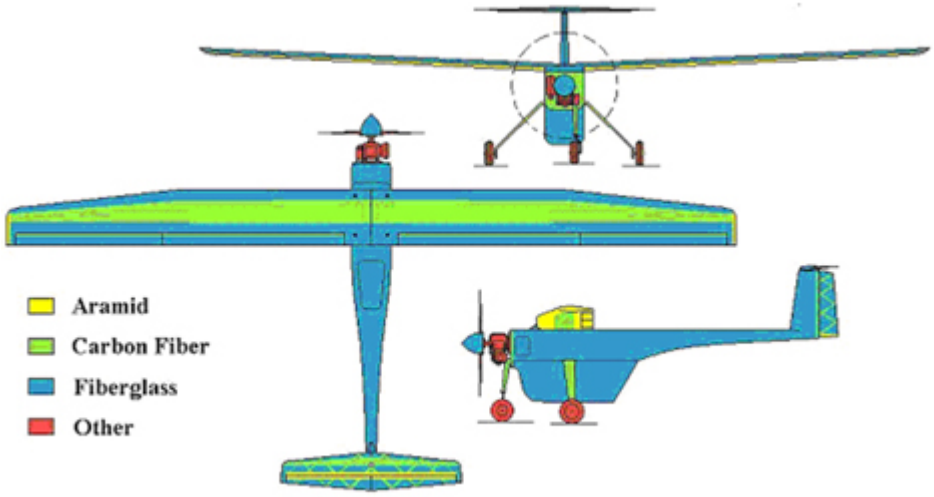

Figure 2: Use of composites over aircraft structures

Most noticeably, we have used Hoerner style wingtips (figure 3) on the main wings. While aerodynamic improvements such as $\frac{L}{D}$ are not significant in such a small aircraft, the new wingtips provide for a reduction in turbulent airflow over the outboard edges of the ailerons [8], a compelling case for the change. In addition, we painted directly on the Mylar during the vacuum bagging process. This method is preferred over using primer on Mylar then painting the surfaces later because this "pre-painting" leaves a mirror smooth finish on the wing surfaces resulting in a lower Reynolds number, and thus, higher lift and lower drag coefficients. Even with these changes, we do not expect to see a near-perfect laminar airflow because of the lack of precision in the wing-bagging process which results in surface waviness. The hollow-molded wing method, which would have resulted in a more improved wing surface, was not used because of the high costs involved with tooling. Finally, kevlar is used in areas where the ability to absorb energy is critical (such as the lower wing surface) and where high strain-to-failure is needed (such as the hingelines of the flaps and elevators). The method of using Kevlar as a hinge is known as a live-hinge and results in smoother airflow over the hingeline and a stronger hinge. For an overview of the materials used in construction, see figure 2. Overall the new aircraft is tougher and more aerodynamic than the old aircraft, and at the same time, maintains the same weight.

1.5 Payload Environmental Isolation

As in virtually every aerospace application, vibration isolation and EM shielding are critical for our payload; this is especially critical in our aircraft where sensitive and critical electrical are present. Early in our testing, we found that our telemetry system would crash within seconds of engine run-up; after probing deeper, the problem revealed itself as the microphonic effect [11], where ca-

Figure 3: Hoerner Wingtips on our RnR APV-3

pacitors placed in vibrational modes create driven electrical feedback in their circuit. In order to mitigate this problem, our electronics have multiple stages of vibration isolation. In addition, we have taken steps to shield our electronics from any background EM radiation. While this shielding does add some weight to the aircraft, it is necessary considering that the engine spark plugs generate electrical noise and the aircraft communicates with the ground through amplified transmitters.

In order to mitigate vibrations, we took numerous steps. First, the electronics are mounted onto a layer of ultra-soft antistatic polyurethane foam (0.75") which provides isolation against low amplitude vibrations. Next, the assembly is mounted in either aluminum or antistatic acrylic cases wrapped in copper shielding. The copper creates a faraday cage around the electronic components, preventing EM interference from affecting performance. Finally, the case is attached to a polyurethane foam bi-layer (1") which provides protection against significant magnitudes of vibration. After taking these steps to isolate our electronics from vibrations, our system has performed beautifully with no recorded microphonic effects.

2 Autopilot

2.1 Hardware Overview

During the competition, the autopilot must fulfill a number of mission-critical requirements including navigating through waypoints (including those sent during flight), able to receive and execute commands specific to aircraft systems such as altitude, orientation, and velocity, and must have the ability to interface with custom payload packages such as a camera and on-board computer. The Crossbow MNAV (μNav and Stargate board) was chosen because it is a reliable and easily customizable system that met or allowed us to meet all of the mission criteria. It effectively flies between waypoints, can fly to new waypoints uploaded during flight, and can send data to custom

payload packages. Because the autopilot software is open source, we can make modifications as needed.

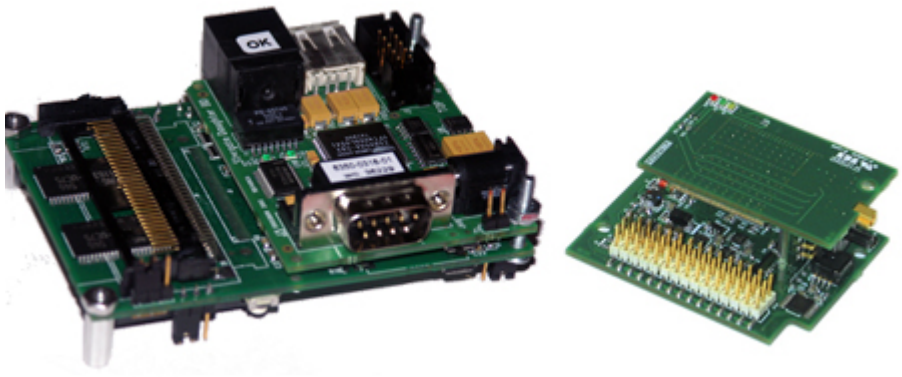

Figure 4: Stargate (left) and Crossbow μNav (right)

2.1.1 μNav Inertial Motion Unit

The Crossbow inertial navigation consists of two components: the μNav inertial motion unit (IMU) that has an array of onboard sensors as well as a processing platform that performs all data filtering and performs all controls and navigation computations. The μNav takes data from static and dynamic pressure sensors, three 3-axis accelerometers, three rate sensors, three magnetometers, three temperature sensors, a GPS sensor, and nine PWM servo outputs. This redundancy allows the unit to continue functioning even in the event of a single component failure; when all redundant components are healthy, the numerous simultaneous data streams allow for efficient Kalman filtering. The μNav does not have the capability to control the plane on its own and instead sends out a serial stream of acquired data for a more powerful computer to process.

2.1.2 Stargate Computer

Data from the μNav flows to the Stargate processing through a 51-pin Hirose style connector over standard GPIO (General Purpose Input/Output) lines. The Stargate handles all the necessary data processing and communications tasks necessary for the aircraft navigation. It features a 400MHz Intel PXA255 Processor running Linux, 64MB RAM, 32MB Flash memory, as well as numerous input and output ports. Because of its available power, it can easily handle added algorithms like preturning, faster datalogging, and enhanced Kalman Filtering.

2.2 Data Flow

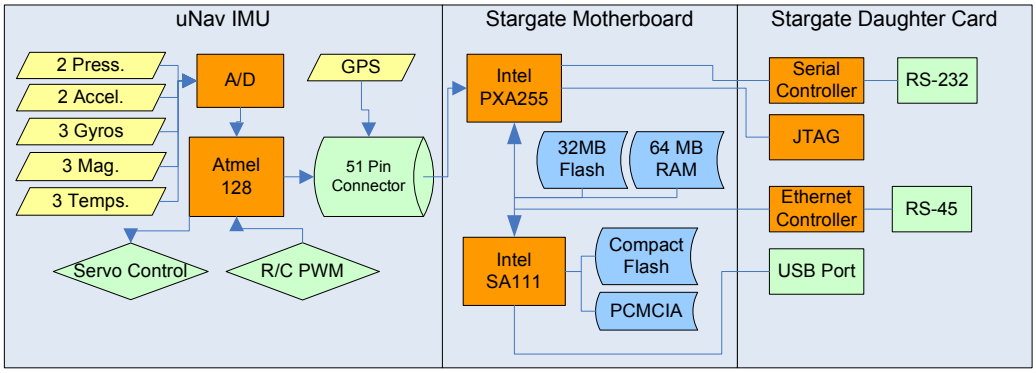

Figure 5: Data flow through the μNav and Stargate systems

While in flight, the aircraft can not wait for data to flow from the IMU to the navigation and control software; efficient data handling is critical for the mission. The autopilot software is written in a combination of C and Assembly, and is rigorously tested to ensure that no timing changes occur. See figure 5 for a detailed overview of the autopilot data flow.

2.3 Navigation

The μNav uses a GPS-driven navigation system which gives a high degree of accuracy while keeping hardware overhead low in comparison to inertial navigation systems. Our operator first selects a series of waypoints on a geo-referenced map file in the Crossbow user interface and then gives inputs various parameters including speed at waypoint and altitude. The Ground Station software then displays the flight track over the given map; if the track is acceptable, the operator uploads the points to the autopilot and the aircraft immediately flies to join the flightpath. In situations where a highly detailed flight plan is necessary, we can import a list of waypoints in .csv format and display them on the screen.

Because of inherent drift in the GPS signal, the μNav now has a robust Kalman filter that allows a GPS accuracy of 3 meters [4]. This is sufficient when considering that our path determination is not exact, and so, an exact GPS system would not be necessary. All sensor data is also run through finely tuned Kalman filters which ensure that aircraft noise (EM and vibration) and sensor drift do not compromise aircraft control.

2.4 Safety

Special care was taken to ensure that our aircraft would perform in a reliable manner. The autopilot systems themselves are double and triple redundant (three 3-axis accelerometers, etc), allowing the aircraft to return to base when flying out-of-sight missions even after sensor failures. An electronic switch on the autopilot gives control of the aircraft to the R/C pilot on command; in the event that this switch fails, we have installed a servo-driven switch that mechanically removes the autopilot from the control loop. If all other measures fail, the R/C receiver has the ability to perform a hard over of the aircraft and kill the engine.

3 Payload and Telemetry

3.1 Systems Overview

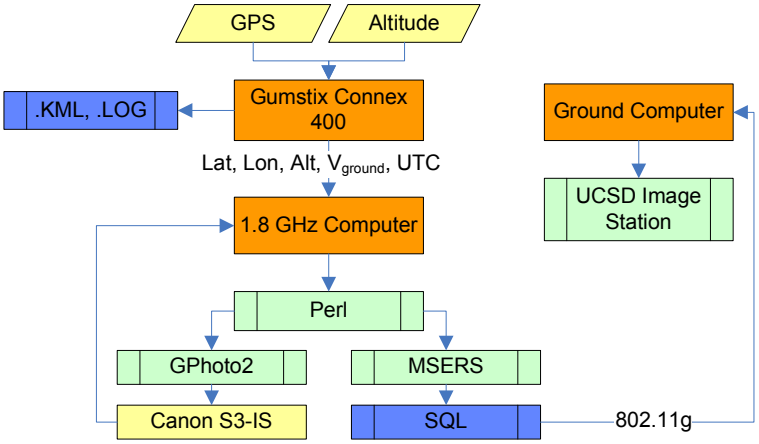

Figure 6: Dataflow through payload systems

Our UAS payload revolves around a 1.8GHz single board computer. This machine pulls in telemetry data from a second telemetry computer and then logs the data; for redundancy, data is also logged on the telemetry computer. At the same time, the main computer fires the camera shutter, downloads the image, correlates it with the telemetry via a synchronized UTC timestamp, and places the image into a queue for image processing. If the image processing algorithm finds a potential target, the telemetry and image information is written into an SQL database which is synchronized with the ground station (during flight through 802.11g or after landing via ethernet) and viewed by the operators.

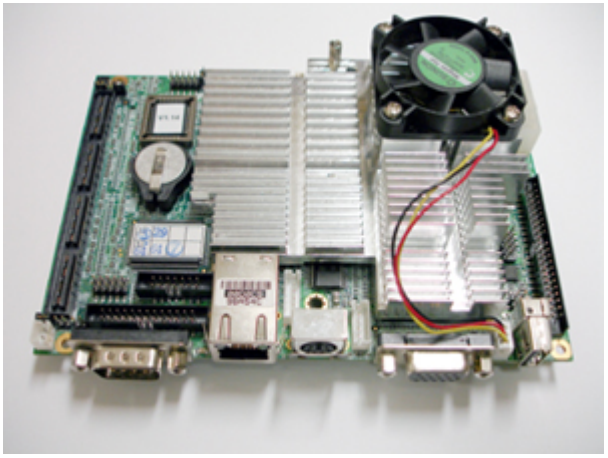

Figure 7: Central Computer

3.1.1 Central Computer

The heavy-duty processing on board our aircraft is performed by an Advantech PCM-9380 single board computer (figure 7). It features a 1.8GHz Intel Pentium M processor as well as 1GB of RAM, an 80GB hard drive, and an additional 4GB of solid state memory. This machine allows our team to process images on board the aircraft as they are captured; the ability to detect targets in real time is highly advantageous as any situation requiring aerial surveillance is inherently highly dynamic.

While the real-time feedback is useful, another upside to having an onboard processing capability is the ability to preserve data in instances where transmission is either difficult or dangerous. During last year's competition, our team noticed that there was a great deal of electrical interference with video transmitted to the ground during the flight and given the nature of electronics work at Webster Field, we believe that similar noise patterns will be present this year. As a result, processing and storing the images on board the aircraft allows us to preserve data in the case of interference. In real combat or surveillance situations, UAS operators may wish not to give away the aircraft location through a constant EM emission, and so, the ability to process and store data with no outside communication is also advantageous.

3.1.2 Telemetry Computer

A main component of the competition requires us to report the GPS coordinates of those targets that we find. Our sensors – GPS and altimetry – are connected to a 400MHz Gumstix 400-xm computer. Because the motherboard weighs only 25 grams including the GPS chip and altimeter,

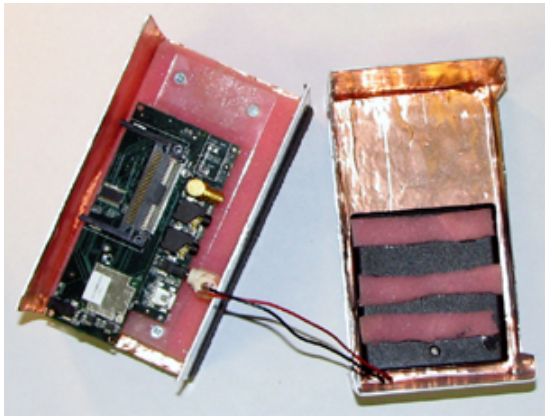

Figure 8: Gumstix in case

the system also places a minimal load on the airframe. The GPS receiver, a uBlox LEA4-H has an accuracy of 2.0m [12] so the target positioning is highly accurate. The system runs a lightweight Perl script on its embedded Linux operating system that pulls in GPS data – latitude, longitude, and altitude – as well as a more accurate reading of the altitude from the altimeter. The Gumstix then sends this data to an SQL database stored on the central computer which we can access from a computer on the ground.

3.2 Image Capture

3.2.1 Camera

In order to select the optimum camera for the mission, we first examined the various known flight and target parameters. In order to find the resolution of the camera R, we start with a flight altitude h, camera viewing angle θ, average target size δ, algorithm mandated target resolution r, and height:width ratio of 3:4. Using simple geometry, we find that:

$$R = \left(\frac{2h\cos(\theta)r}{\delta}\right)^2 * \frac{3}{4} \tag{1}$$

If we fly at an altitude of 500 feet using a camera with a 30° half-viewing angle, look for targets that average 3 feet across, and run through a filter that requires targets of 9 pixels, equation 1 suggests a 5.1 megapixel camera. On the other hand, a slightly faster algorithm might search 10 pixels, and thus would mandate a 6.3 megapixel camera.

We also considered other factors when selecting our camera such as lens quality and distortion, image stabilization, and computer controllability. With these requirements in mind, we selected

a Canon S3-IS which captures images at 6.0 megapixels. This resolution allows our image recognition algorithms to easily distinguish targets from markings on the runway and discolorations in the field. The camera also features a hardware controlled image stabilizer which removes high frequency vibrations from the lens assembly, thus resulting in a clearer picture.

3.2.2 Image Stabilization

Aside from GPS accuracy, the main requirement for accurately locating the target is camera stability. From geometry, it is obvious that as the camera tilts away from the vertical, the center of the image will move away from the measured GPS location. Because the airframe will be turning and the payload will experience vibrations from the engine, an effective stabilization mechanism is necessary.

In order to keep the camera pointed at the ground at all times, we designed and implemented a two-axis gimbal. As the camera moves away from the vertical, the image center moves away from the GPS center of the plane and the picture undergoes a geometric shear. While we could have compensated for this error computationally, such an approach would have added extraneous complexity to the processing system and so we chose a mechanical system. Because it is desirable to keep any antennas pointed orthogonal to the aircraft track along the Earth's geoid [6], we have also attached the antenna to the gimbal.

The gimbal is composed of a two degree of freedom (DOF) pan/tilt assembly which houses the main camera and comuunications antenna. Each DOF is controlled by a servo which is first mounted onto lightweight machined acrylic and then to a 1/8" aluminum frame. During flight, the pan/tilt directs the camera using uNav's PicoTilt module, a PIC processing board that recieves integrated rate measurements from a two-axis accelerometer and outputs a nulling PWM signal to the gimbal servos.

We selected miniature Hitec HS-225MG servos because of their size and weight; at roughly 1 oz, they are lighter than standard servos (about 1.6 oz) and have a similar torque (54 oz-in @ 4.8 V). The servos were mounted low on the gimbal – beneath the plane of the camera lens – to facilitate motion as well as provide a vibration dampening counterbalance.

Our gimbal provides a full range of motion of 20° from the camera center and most of the weight resides in the camera (18 oz). As a result, the gear selection was constrained by the desired range of motion as well as the needed torque to move the camera. We calculated that $\tau_{required} = 72$oz-in; consequently, a 48-pitch 0.5" pitch diameter gear is mounted on the servo with a corresponding 1.5" pitch diameter hub-mounted gear connected to each pan/tilt frame.

The gimbal is affixed to the aircraft using rubber feet which compress and expand to keep the gimbal as vibration free as possible. While this affords the camera a higher degree of stability, its

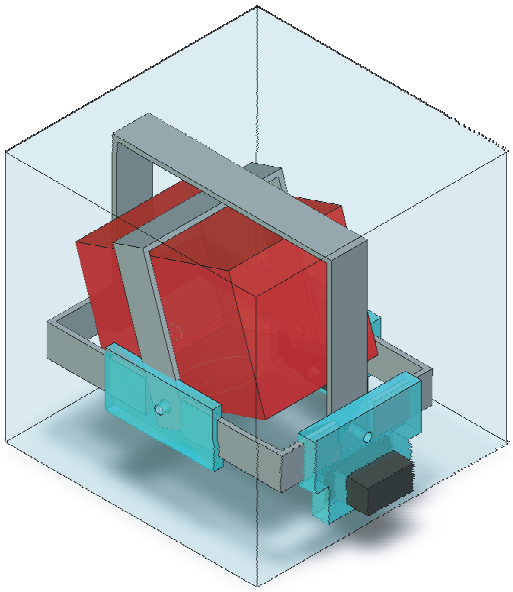

Figure 9: CAD of 2-Axis Gimbal

main advantage is providing a vibration-free environment for the accelerometers. Since vibrations also exist in the lateral plane, parallel to the ground, image stabilization hardware and software on the camera allow a final mode of control against ultra-high frequency vibrations. Together with rubber mounts and the gimbal, our induced GPS error from the platform instability is greatly reduced; as an added bonus, our images are rarely out of focus.

3.2.3 Camera Control

One of the more important aspects of the payload is our camera control mechanism; after all, even the world's best image stabilization hardware and image recognition software is useless if we can not capture any images. The ideal method of control was a point of debate for our team; a mechanical system was the most simple to design and integrate while a computer-based system would allow for the greatest level of dynamic control. We settled on using the open source GPhoto2 software [5] which lets our imagery code fire the shutter when needed and download images as they are captured. This flexibility allows us to improve our payload year after year without the need for hardware changes.

3.3 Communications

3.3.1 Previous Experience

During last year's competition at Webster Field, our team found that there was a great deal of radio interference between our payload package and the 900MHz frequency on base. As a result, we have developed a robust communications network over the 802.11g band allowing us to transmit data at high bandwidth with little interference. Considering last year's experience, we do anticipate some noise on the 802.11g band, especially considering that the military is allowed use of 5 watt amplifiers while civilians are limited to 1 watt amplifiers [10]. Even so, several studies indicate that WiFi is a promising method for communicating with UAS's ([3], [7]), and so, we expect to field a relatively successful data network.

3.3.2 Specifications

Our hardware selection for the communications network was driven by FCC Regulation Sec. 15.204 [1] which notes that "an external radio frequency power amplifier shall be marketed only in the system configuration with which the amplifier is authorized and shall not be marketed as a separate product." Our communications system then consists of two Hawking HWR54G routers and two Hawking HSB2 amplifiers (one pair on the ground and one on the aircraft). This system will give us access to images from the aircraft at high speed($\geq$ 11MBps) during flight, allowing us to classify targets before waiting for the aircraft to land.

3.4 Image Recognition

3.4.1 Mathematical Description of MSERS

An excellent mathematical overview of MSERS is given in [9], but we have provided much condensed summary of the methods below:

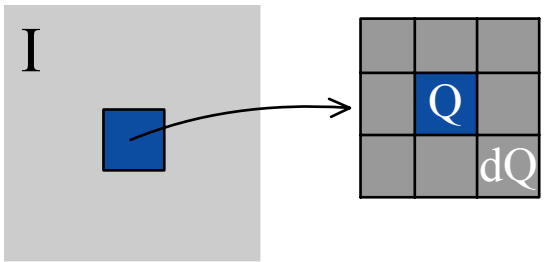

Figure 10: MSERS Region Schematic

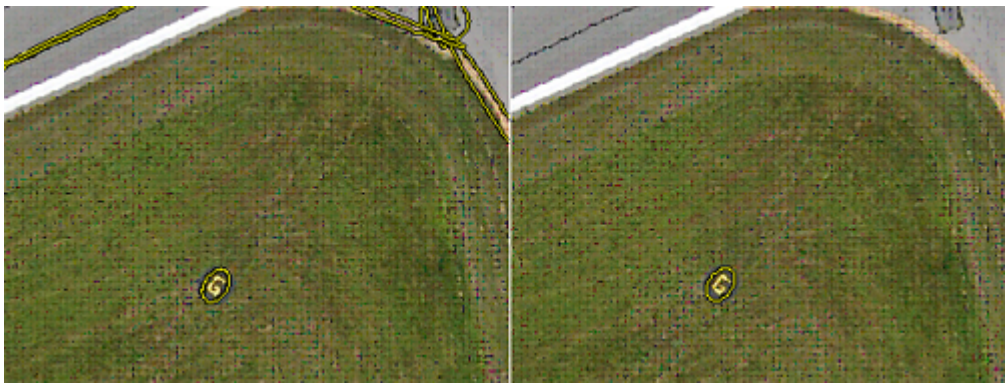

Figure 11: MSER results before and after filtering. Target data courtesy [2]

- Starting with the image I, define a set $S = \{0, 1, ..., 255\}$. This set for S is used as a threshold for greyscale values and the following steps are repeated for each value.
- At each value of S, make a copy of the image. Then, set all pixel values below S as 0 (black) and those
- Select a region of the image Q where analysis will be performed and define the boundary as δQ.
- Define all pixels of value greater than S as white, all less than S as black.
- If the intensity $F(Q)$ is greater than the intensity $F(\delta Q)$ then the region Q is considered a "maximally intense region"
- A "maximally intense region" remains through many values of S and is identified as a possible target.

An easy way to visualize the MSERS method is to think of a 256 frame movie where each frame is increasing in the number of black pixels. Each frame of the movie is the image to be analyzed where regions of exceptionally high intensity are noted as potential targets. Those regions that persist through a great number of frames – relative to other "maximally intense regions" – are classified as targets by the algorithm.

3.4.2 Filtering of MSERS results

While MSERS gives exceptionally accurate results in a very short amount of time, the potential for false positives is quite high as runway markings, vehicles, and patches of dead grass can all be classified as maximally extremal. However, after passing the results through a filter, we can remove virtually all false positives (as seen in figure 11). Because the MSER algorithm reports results by a

center point and semi-minor and semi-major axes of a bounding ellipse, the computational power needed to filter the results is minimal.

- First, the ratio $\frac{semi-minor}{semi-major}$ is taken. If this is less than 0.20 (slightly smaller than the smallest ratio given by AUVSI), the target is removed.
- Knowing the image altitude, we calculate the area of the remaining ellipses. Those with areas less than $3ft^2$ or greater than $75ft^2$ are removed.
- Since the algorithm may identify the border and letter of the target as separate, we remove all but the largest of concentric ellipses.

Because the filter requires very little computation, its overhead increases processing time per image by only a few percent (depending on the number of targets found).

3.4.3 User Interface

A critical portion of our mission is allowing observers at the base station to view target pictures and locations as they are captured. In order to meet this criteria, an integral part of our system is an image station, written in Java. The software continually polls the aircraft's SQL database for changes, synchronizes the database with a similar one on the ground, and then displays the contents of the database in the interface. The table specifies if the MSERS algorithm found a target in the image; the user can then select that (or any other) image; when he does so, the software downloads the image from the aircraft via FTP and displays it in the main window. See figure 12 for a more detailed look at the Image Station.

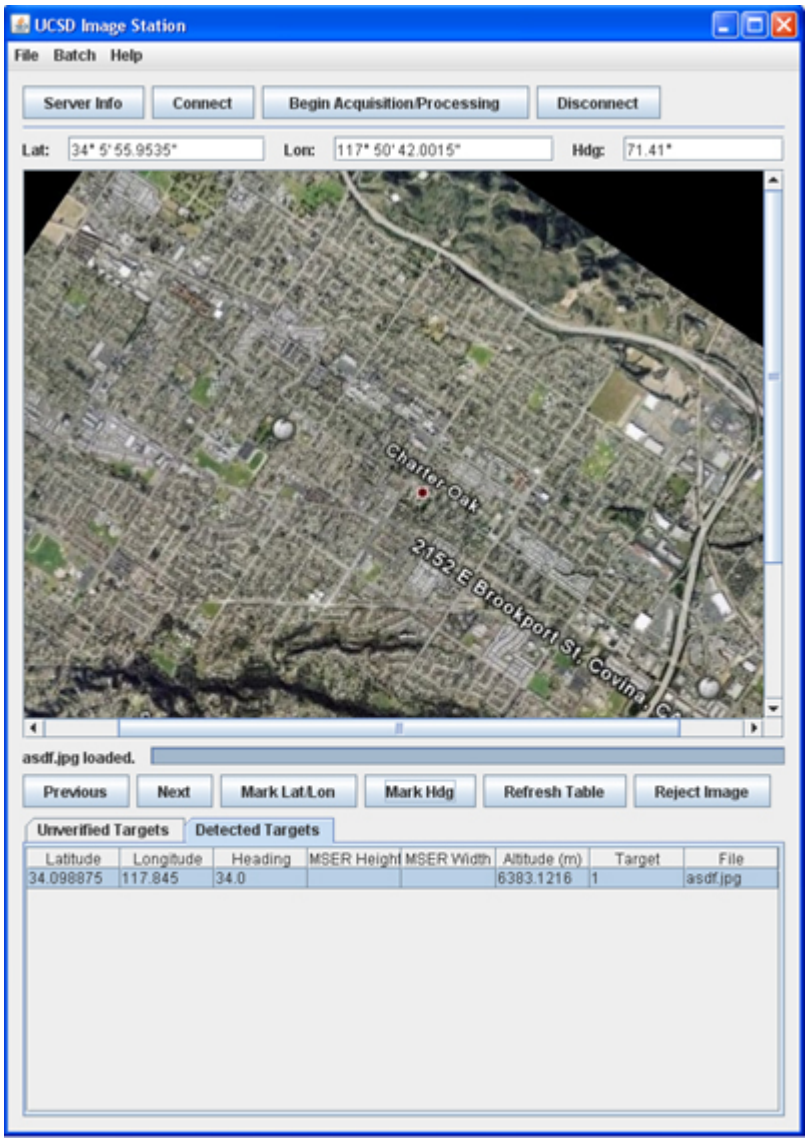

Figure 12: Target image station

4 Acknowledgements

4.1 Sponsors

4.2 Faculty and Staff

We would like to thank the following members of the UC San Diego community for all of the guidance that they have given our team:

- Professor John Kosmatka, Faculty Advisor, Structural Engineering and Mechanical/Aerospace Engineering
- Professor Gabriel Rebeiz, ECE Advisor, Electrical and Computer Engineering
- Carson Richard White, Graduate student, ECE
- Dr. Matt Last
- Professor Rick Ord, Computer Science and Engineering

References

[1] Oet. part 15.204 - external radio frequency power amplifiers and antenna modifications. Technical report, Federal Communications Commission, 2004.

[2] Nc state auvsi, aerial robotics. http://art1.mae.ncsu.edu/page.php?page=Downloads, May 2007.

[3] C. Cheng, P. Hsiao, H. Kung, and D. Vlah. Performance measurement of 802.11a wireless links from uav to ground nodes. Technical report, ICCCN, October 9-10 2006.

[4] Crossbow Technologies. *MNAV Navigation and Servo Control Board Datasheet*, 2007.

[5] Gphoto2. http://gphoto.org.

[6] D. Hague, H. Kung, and B. Suter. Field experimentation of cots-based uav networking. Technical report, Air Force Research Lab, 2006.

[7] S. Herwitz, J. Leung, R. Higgins, S. Dunagan, and J. Arvesen. Remote command and control of imaging payloads using commercial off-the-shelf technology. In *International Geoscience and Remote Sensing Symposium*, June 2002.

[8] J. B. Jarvis and B. P. Selberg. Investigation of aerodynamic improvements using wing tip sails. In *37th Aerospace Sciences Meeting and Exhibit*. AIAA, January 1999.

[9] J.Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. *BMVC*, pages 384–393, 2002.

[10] P. Marshall and T. Valdes. Aries-tern network connectivity experiment - quicklook report. Technical report, Office of Naval Research, Organic Mine Countermeasures and Future Naval Capabilities, June 2004.

[11] R. Nelson and L. Davidson. Electrical noise generated from the microphonic effect in capacitors. *IEEE International Symposium on Electromagnetic Compatibility*, 2:855–860, 2002.

[12] uBlox. *LEA-4H ANTARIS 4 Programmable GPS Module with SuperSense Datasheet*, 2007.