Controlling Unmanned Aerial Vehicles During Crisis Response

Lt Col Ricky E. Sward and Lt Col Stephen Cooper

Institute for Information Technology Applications
Untied States Air Force Academy
Colorado

IITA Research Publication
Technical Report Series–06-1

April 2006

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **APR 2006**	2. REPORT TYPE	3. DATES COVERED **00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE **Controlling Unmanned Aerial Vehicles During Crisis Response**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Institute for Information Technology Applications,HQ USAFA/DFPS,2354 Fairchild Drive Suite 6L16D,USAF Academy,CO,80840-6258**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**18**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

ABOUT THE AUTHORS

Stephen D. Cooper

Lt Col Cooper is currently the Director of Operations for the UAV Research Group and an operational Air Force pilot. He is qualified to fly the Piccolo based Silver Fox, Pointer, and the Alpha 60-based UAV systems. He has been doing UAV research for over two years. Lt Col Cooper has a Masters in Information Technology. He also has a BS in Computer Science from USAFA.

Ricky E. Sward

Lt Col Sward retired from the Air Force in August 2006. He served as the Director of the USAFA UAV Research Group from 2004 to 2006. He has been flying UAVs and conducting UAV research for over three years. He is qualified to fly the Desert Hawk UAV system and the Piccolo-based Silver Fox, Pointer, and the Alpha 60-based UAV systems. Lt Col Sward has a PhD in Computer Engineering from AFIT, a Master's in Computer Science from the University of Colorado, Boulder, a Master's in Management of Information Systems from the University of Colorado, Boulder, and a BS in Computer Science from Iowa State University.

Cadets

The cadets in CS 453, Software Engineering I, and CS 454, Software Engineering II were first-class cadets in their last year at the USAF Academy. Their majors included Computer Science, Systems Engineering, and Systems Engineering Management. They have graduated and have been commissioned as Second Lieutenants in the Air Force.

The views expressed in this paper are those of the authors and do not necessarily reflect the official policy or position of the Institute for Information Technology Application, the Department of the Air Force, the Department of Defense or the U.S. Government.

Comments pertaining to this report are invited and should be directed to:

Sharon Richardson
Director of Conferences and Publication
Institute for Information Technology Applications
HQ USAFA/DFPS
2354 Fairchild Drive, Suite 6L16D
USAF Academy CO 80840-6258
Tel. (719) 333-2746; Fax (719) 333-2945
Email: sharon.richardson@usafa.af.mil

Table of Contents

Introduction 4
UAVSAT Overview 4
Field of View 6
Reducing Video Delay 9
Unmanned Aerial Vehicles 11
Cadet Involvement 14
Future Work 15
References 16
Glossary 17
About the Institute 18

Figures

Figure 1: Telemetry and video streams from UAV to users 5
Figure 2: Google Earth used to display location of UAV 5
Figure 3: FOV calculations 6
Figure 4: Automatic orbit capability 7
Figure 5: Correcting for wind 8
Figure 6: The PelcoNet 350T Video Transmitter 9
Figure 7: Video display window 10
Figure 8: Alpha 60 UAVs 11
Figure 9: Piccolo autopilot 12
Figure 10: AFRL/SNAR Gimbal Device 12
Figure 11: Sanyo Color CCD camera 13
Figure 12: Raven UAV 13
Figure 13: Team organization 14

Controlling Unmanned Aerial Vehicles During Crisis Response

Lt Col Ricky E. Sward and Lt Col Stephen Cooper

Introduction

The purpose of this project was to continue our research on Unmanned Aerial Vehicles (UAVs) and their usefulness during a response to a crisis situation. In previous work [1], we have shown that both the video feed and the location of the UAV can be displayed in a command center to increase the situational awareness of commanders. The research presented here will build on our previous work and improve the situational awareness tool available to the commander. This tool is referred to as the UAV Situational Awareness Tool (UAVSAT).

UAVSAT Overview

This academic year, 2005-06, in the Computer Science Department's Software Engineering courses 11 Computer Science students, three Systems Engineering students, and one Systems Engineering Management student worked as a team to develop UAVSAT. During the course of the year, a field of view representation was added onto the map showing where the UAV camera is pointing. We have added an automatic orbiting capability so a user can indicate where the crisis is occurring and the UAV will orbit over that point on the map. We have also reduced the amount of delay between when the video was shot by the camera on board the UAV and when it is presented to the commander. This improves the synchronization of the video signal and the location of the UAV.

As shown in Figure 1, there are two streams of data being sent down from the orbiting UAV. The telemetry stream is received by the Ground Control Station (GCS) and is sent via a network connection to a separate process that stores the telemetry data into a MySQL database. The latitude, longitude, altitude, heading, and serial number from the UAV are stored in the database. Client users across the USAFA intranet use Google Earth to access and display the location of the UAV. Google Earth uses a PhP file and XML to access the telemetry information that has been stored in the database. Google Earth displays the location of the UAV in three dimensions given the telemetry information. The video data stream is sent from the UAV and received by a video receiver on the ground. The video stream is then sent directly to a PelcoNet™ transmitter that distributes the video signal to client users across the intranet. This transmitter will be discussed further below. A more detailed architecture diagram is available from the authors upon request.

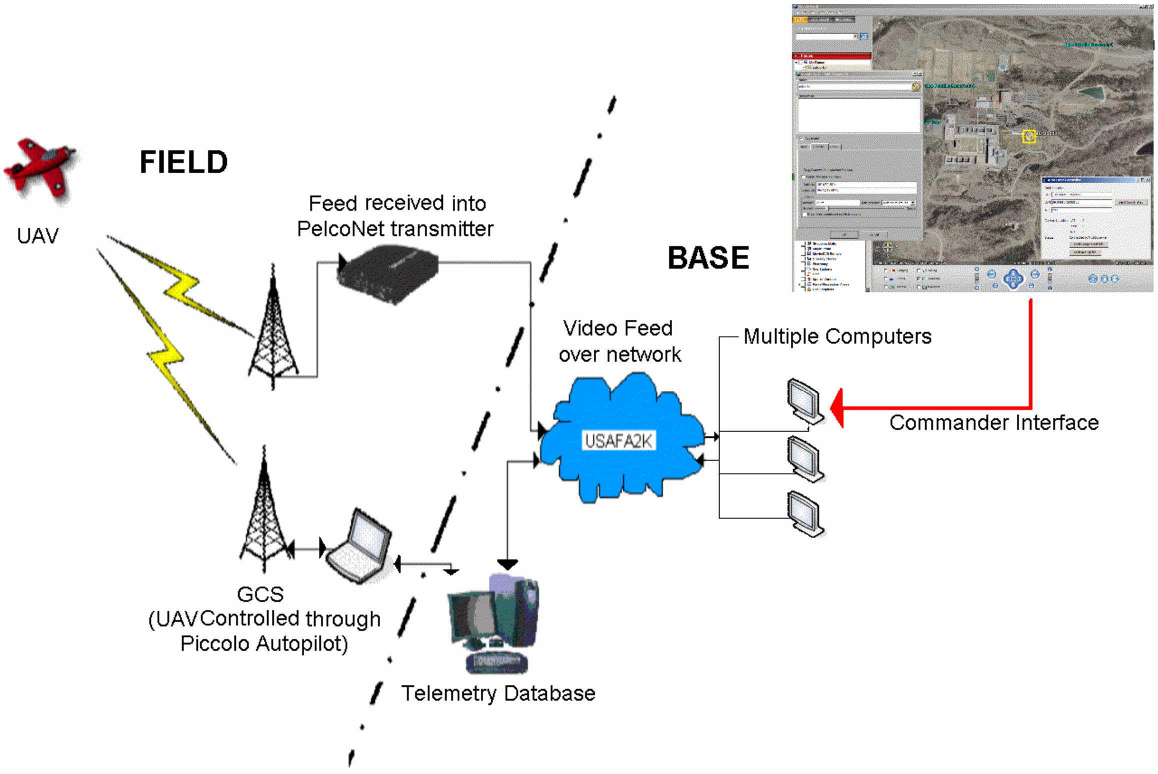


Figure 1 – Telemetry and video streams from UAV to users

The resulting application displays the location of the UAV in three dimensions using the Google Earth tool. The video stream is displayed in a separate application. This is done so that the map can be displayed on one computer monitor or large screen display while the video is displayed on another.

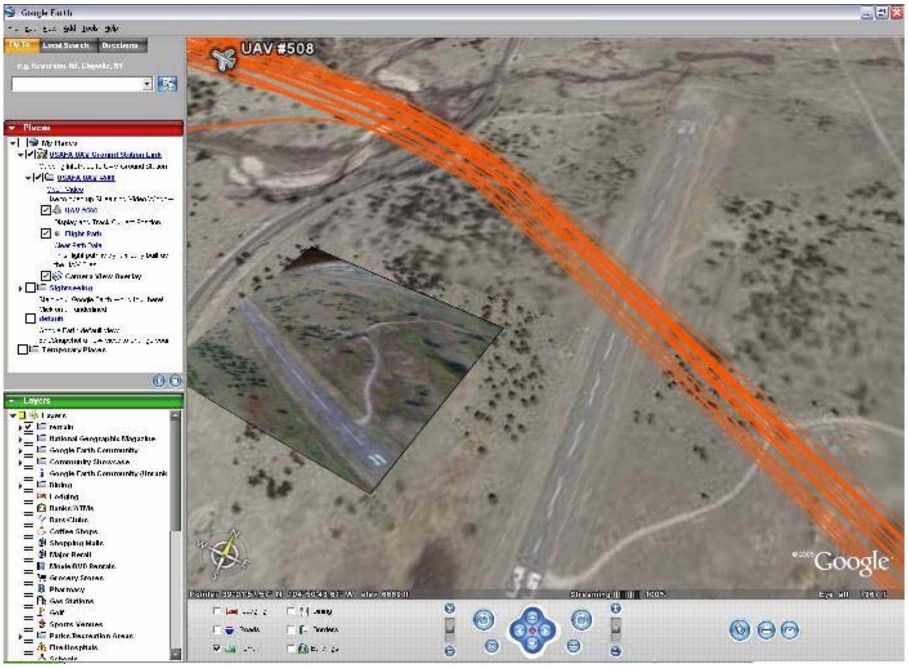


Figure 2 – Google Earth used to display location of UAV

Figure 2 shows a screen shot of a Google Earth display from the prototype application. The location of the UAV is shown as a two-dimensional airplane icon and can be turned on and off. The path of the UAV is shown on the display and can be turned on and off.

Field of View

The first new capability added to UAVSAT is a field of view (FOV) indicator on the map. The FOV indicator is a polygon that indicates on the map where the image from the camera is located with respect to the ground. The intent of the polygon is to show the *geo-location* of the video image, i.e. where it is located geographically on the map. The edges of the polygon are geo-located on the map by calculating the latitude and longitude of the edges and passing this information to Google Earth. This calculation is affected by the angle of the gimbal device holding the camera, the bank angle or pitch of the UAV, the focal length of the camera, and the terrain below the UAV. The user is able to turn the FOV indicator on and off.

As shown in Figure 2, an image from the video stream is placed inside the FOV polygon. This image is taken from the video stream once each second. The FOV is precisely placed onto the map based on the altitude of the UAV and the focal length of the camera. Although the calculation of FOV is affected by the bank and pitch of the UAV, we did not include these effects in the FOV calculations. We placed a large simplifying constraint on the FOV assuming that the camera is pointing straight down and there is no pitch or bank of the UAV. Admittedly this is a large simplifying assumption, but our plan is to remove this constraining assumption in future research.

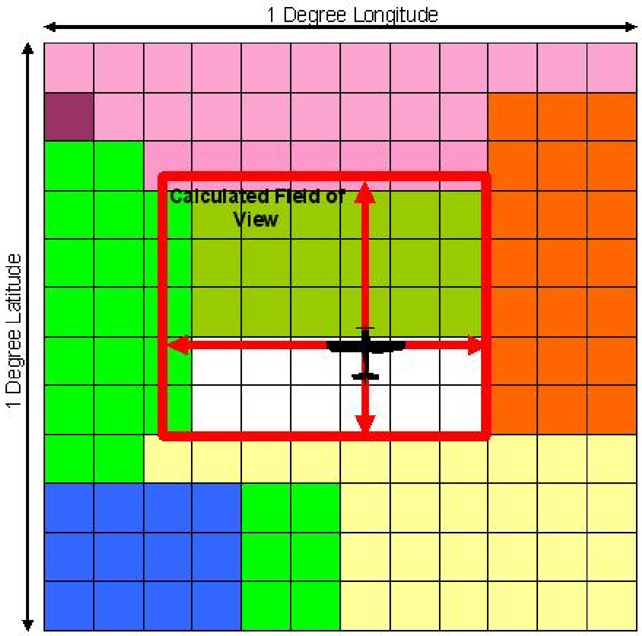
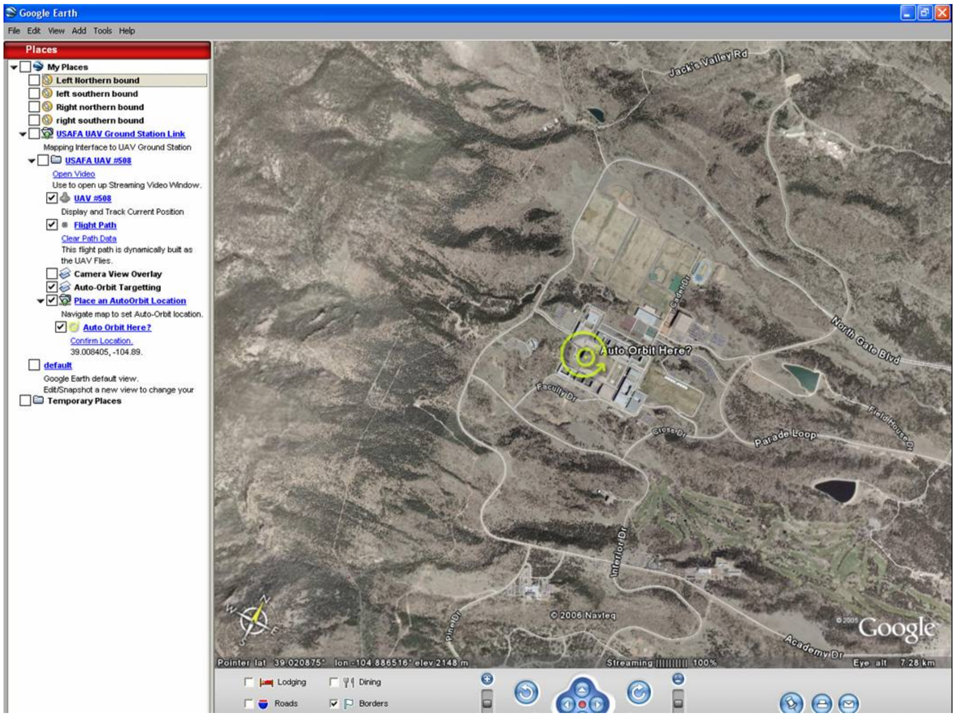


Figure 3 – FOV Calculations

The FOV calculations take into consideration the effect of the elevation of the terrain being flown over by the UAV. The FOV algorithm uses Level 1 Defense Terrain Elevation Data (DTED) to determine the altitude of the terrain below the UAV. Each DTED file contains 1200 x 1200 pieces of ground elevation data over an area of 1 degree longitude by 1 degree latitude. All values from the DTED data file are stored into a 2-dimensional array in Ada, which can be referenced in a constant amount of time. The colors in the chart shown in Figure 3 represent different ground elevations. The FOV algorithm begins at the camera and calculates the expected decrease in elevation along the path to the edge of the image given the change in latitude or longitude. It then compares the expected altitude along this path to the ground elevation. When the expected altitude along the path is less than the DTED ground elevation, the point where the image edge is to be placed on the ground has been found. The algorithm calculates four such points for the front, back, left and right sides of the image. These points determine the extent of the FOV box.

As a result, UAVSAT now includes an effective, yet limited depiction of where the UAV's video image is geo-located on the terrain. The FOV box gets larger as the UAV climbs and gets smaller as the UAV descends. The FOV box changes in shape and size as the UAV flies over mountainous terrain. Using these calculations, UAVSAT improves the accuracy of the FOV indicator and provides the commander a more accurate depiction of where the UAV's image is geo-located.



☐ My Places
  ☐ Left Northern bound
  ☐ left southern bound
  ☐ Right northern bound
  ☐ right southern bound
  ☐ USAFA UAV Ground Station Link
    Mapping Interface to UAV Ground Station
    ☐ USAFA UAV #508
      Open Video
      Use to open up Streaming Video Window.
      ☑ UAV #508
        Display and Track Current Position
      ☑ Flight Path
        Clear Path Data
        This flight path is dynamically built as the UAV Flies.
      ☐ Camera View Overlay
      ☑ Auto-Orbit Targetting
      ☑ Place an AutoOrbit Location
        Navigate map to set Auto-Orbit location.
        ☑ Auto Orbit Here?
          Confirm Location.
          39.008405, -104.89.
  ☐ default
    Google Earth default view.
    Edit/Snapshot a new view to change your
☐ Temporary Places

Pointer lat 39.020875° lon -104.886516° elev 2148 m Streaming |||||||||| 100% Eye alt 7.28 km

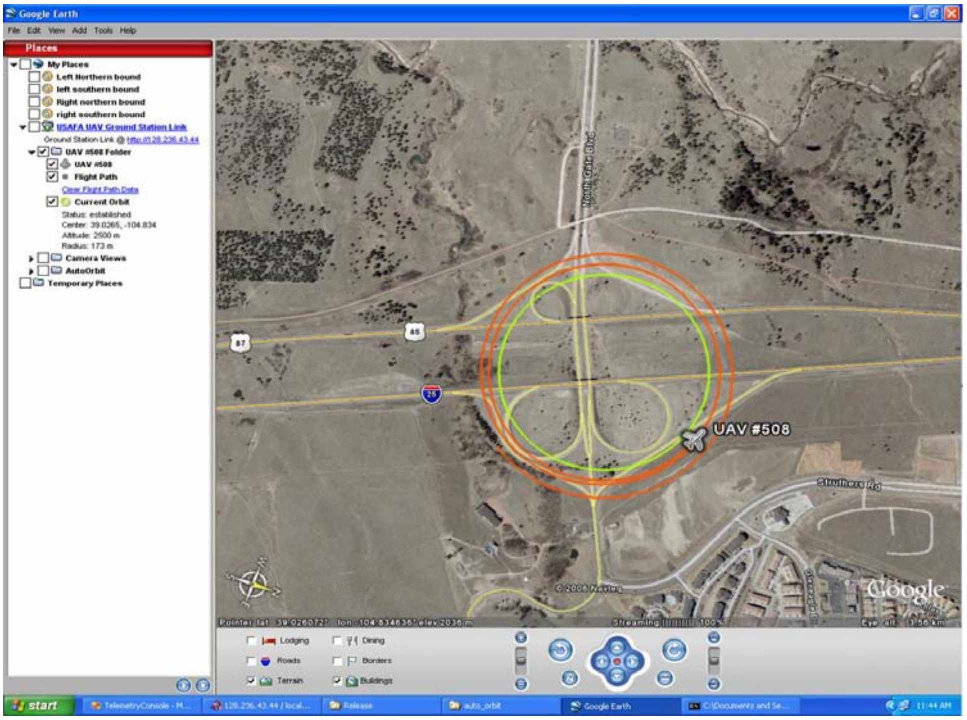
☐ Lodging ☐ Dining
☐ Roads ☑ Borders

Figure 4 – Automatic orbit capability

Automatic orbiting

The second new capability we have added to UAVSAT is an automatic orbiting capability for the UAV. With this new feature the user can indicate via Google Earth where the crisis is occurring and the UAV will orbit over that point on the map. UAVSAT automatically calculates the approach flight path and the proper radius and altitude for the UAV's orbit. This orbit is based on the assumption that the UAV will begin a left-hand orbit at 500 feet above ground level (AGL) and the camera's gimbal angle is fixed at 30 degrees left. UAVSAT calculates the proper radius for the orbit ensuring the camera will point at the selected location. The automatic orbit is designed to provide the best field of view for the UAV's camera giving the commander the best view of the situation.

Figure 4 shows the green circular arrow icon used in Google Earth to establish the auto-orbit. The user is able to turn this icon on or off using the options shown on the left side of Figure 4. Once on, this icon remains centered in the Google Earth image and the user drags the map underneath the icon to indicate where the UAV should establish its orbit. Once the auto-orbit location is placed under the icon, the user confirms the UAV orbit location. UAVSAT then calculates the flight path from the current location of the UAV to the orbit location, calculates the correct orbit altitude and radius, and sends the new flight plan to the UAV. As long as there is a network connection between UAVSAT and the UAV ground station, UAVSAT is able to send the new flight plan up to the UAV. After the new flight plan is sent, the UAV operator in the field must redirect the UAV to this new auto-orbit location. Although it is possible to have UAVSAT send a command to the UAV to have it automatically redirect to the new orbit location, we chose to keep the UAV operator in the command loop. Because of this, there must be some sort of communication channel between the command center and the UAV flight operations area.



Ground Station Link @ http://128.236.43.44

Status: established
Center: 39.0265, -104.834
Altitude: 2500 m
Radius: 173 m

Pointer lat 39.026072° lon -104.834636° elev 2036 m

Figure 5 – Correcting for wind

In order to improve the performance of the automatic orbit capability, UAVSAT includes corrections for the effect of wind, calculations for adjusting the gimbal angle, and the ability for the commander to change the altitude of the orbit. In order to correct for wind, the auto-orbit portion of UAVSAT compares the expected location of the UAV to the actual location on the orbit. The auto-orbit algorithm determines if the UAV should be executing a steeper turn or a shallower turn. The algorithm adjusts the position of the orbit point once each second in order to keep the UAV on its proper orbit flight path. Figure 5 shows the UAV orbiting around a point of interest. In the figure, the inner, green circle is the orbit flight plan being flown by the UAV. The outer, orange circles are the path the UAV has flown. The UAVSAT corrections for wind attempt to keep the UAV on the orbit flight plan.

The auto-orbit algorithm also calculates where the camera gimbal device should be pointing in order to keep the point of interest in view. This is done by compensating for bank and pitch effects from the UAV and calculating the effect on the gimbal angle. This is not a complete end-to-end system solution, but does provide new gimbal angle values from the calculations. These values could be sent automatically to the gimbal control system and automatically adjusting the gimbal device.

The auto-orbit interface also provides the commander the ability to request that the UAV orbit higher or lower. This is done via an option in Google Earth and is done in 100 foot increments. UAVSAT does not allow the commander to select an orbit less than 100 feet AGL.

Reducing video delay

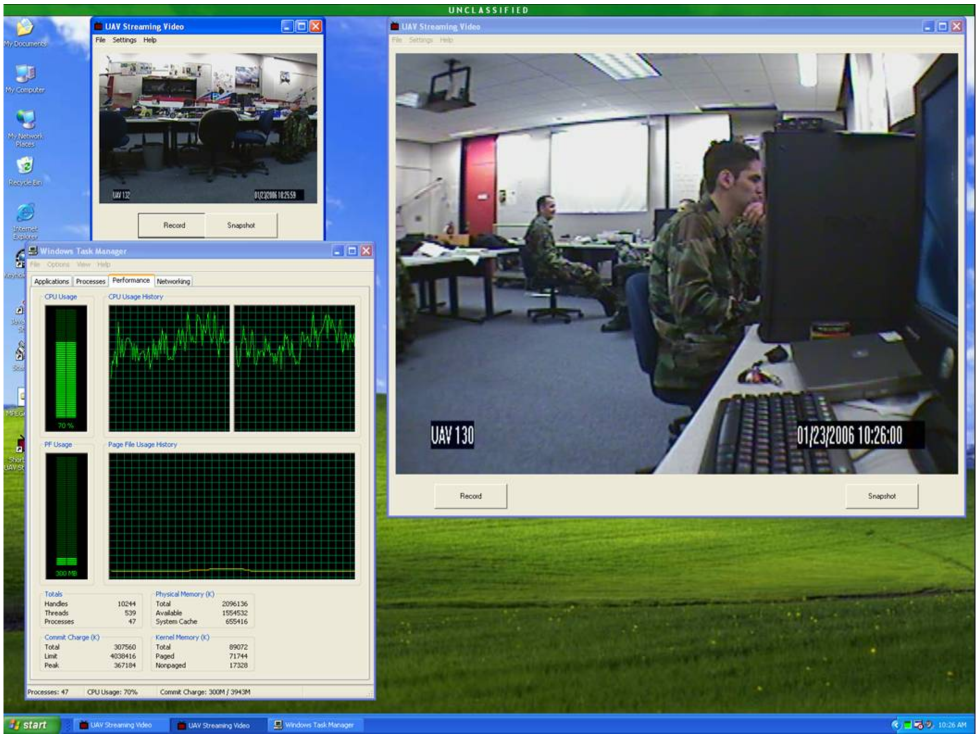
The final capability we have added to UAVSAT is to reduce the amount of delay between when the video is shot onboard the UAV and when it is displayed in the command center. In the situational awareness tool developed in academic year 2004-2005, this delay was between six and nine seconds. This was due to the processing time required to stream the video to multiple users when using Microsoft Windows Media Server [2]. This time delay also causes the video image and the location of the UAV on the map to be out of synch.

By using new hardware from the Pelco Company to stream the video image, we have reduced the video delay. We use the PelcoNet 350T video transmitter (see Figure 6) to stream the video across the existing USAF Academy intranet. We have tested the connection from the UAV lab in Fairchild Hall to the command center and the delay in video transmission was less than one half of a second. The PelcoNet uses MPEG-4 video compression technology to transmit the video across IP. Multiple users can log into the PelcoNet's IP address and view the video.

Figure 6 – The PelcoNet 350T Video Transmitter

UAVSAT includes a standalone Windows application that connects to the PelcoNet and to view the video. The PelcoNet 350T is placed in the field and connected to the receiver on the ground that receives the video stream from the airborne UAV. The video signal is then converted to MPEG-4 format by the PelcoNet 350T. The PelcoNet device is connected to the intranet and the video is then available for viewing. This means there must be a network drop near the UAV flight operations area in order to connect the PelcoNet 350T to the intranet. We do not currently have such an intranet connection near our UAV flight operations, but will continue to pursue this in the future.

Figure 7 shows three separate Windows applications. The first application is in the upper right-hand corner of Figure 7. This application is the UAVSAT application connected to the PelcoNet 350T and this application is displaying the video in the largest possible resolution. The second application is in the upper left-hand corner of Figure 7. This application is also the UAVSAT application connected to the PelcoNet and this application is displaying the video in a smaller resolution. Currently, only these two resolutions are available with the UAVSAT application.



Totals
Handles 10244
Threads 539
Processes 47

Physical Memory (K)
Total 2096136
Available 1554532
System Cache 655416

Commit Charge (K)
Total 307560
Limit 4038416
Peak 367184

Kernel Memory (K)
Total 89072
Paged 71744
Nonpaged 17328

Processes: 47 CPU Usage: 70% Commit Charge: 300M / 3943M

Figure 7 – Video display window

The third application is a Windows utility application that is displaying the CPU usage and memory usage needed to display these two UAVSAT applications. If you look closely, the memory usage is small and constant. The CPU usage, however, is quite high. This particular machine has two CPUs and can easily handle both two video feeds. The performance of the video feed is much more dependent on the CPU than on memory.

In summary, the UAVSAT tool is now capable of displaying a video image from an airborne UAV once per second in Google Earth. It calculates the position of this still image on the terrain using DTED terrain elevation data and adjusts for altitude and terrain elevation. UAVSAT allows the user to select an orbit point and will automatically calculate the correct flight path so that the UAV orbits the point of interest. The user is allowed to move the orbit higher or lower. UAVSAT also provide the ability to connect to the UAV video feed with a delay of less than one half of a second. These features combine to provide the commander a powerful situational awareness tool to be used during emergency situations.

Unmanned Aerial Vehicles (UAVs)

The UAVs we used this year are based on the Alpha 60 remote controlled model airplane built by the Hangar 9 Corporation. Due to limitations on obtaining operational UAVs and the cost of these UAVs, we continued to use the Alpha 60 model airplane for our research. Figure 8 shows the UAV based on the Alpha 60 configured to fly with a Piccolo autopilot. The same Ground Control Station (GCS) that is used for the Silver Fox UAV is used to control our UAV. As part of our work this year, we have converted the UAVs from glow fuel to electric motors. These motors provide more power and are not affected by the high altitude at the Academy.

Figure 8 – Alpha 60 UAVs

The UAVs we used for this research are based on the Piccolo autopilot system. Figure 9 shows the Piccolo autopilot [3]. We used the Software Development Kit (SDK) provided with the Piccolo operator interface to develop UAVSAT. This SDK allowed us to download the UAV telemetry, such as latitude and longitude, once per second. It also allowed us to upload new orbit flight plans for the UAV at least once per second, if needed. The Piccolo autopilot is used in operational UAVs such as the Silver Fox [4] and the Pointer [5].

Figure 9 – Piccolo autopilot

Our UAVs include a video system based on an onboard camera mounted on a gimbal device. The gimbal device (see Figure 10) was developed by the Air Force Research Lab Sensors Directorate (AFRL/SNAR) and provides two degrees of motion controlled by two servo motors. Using a software application, the gimbal can be controlled from the ground to slew up and down or left and right via the servos. We have not completely included this software capability into our UAV system at this time, but have used the gimbal device for testing.

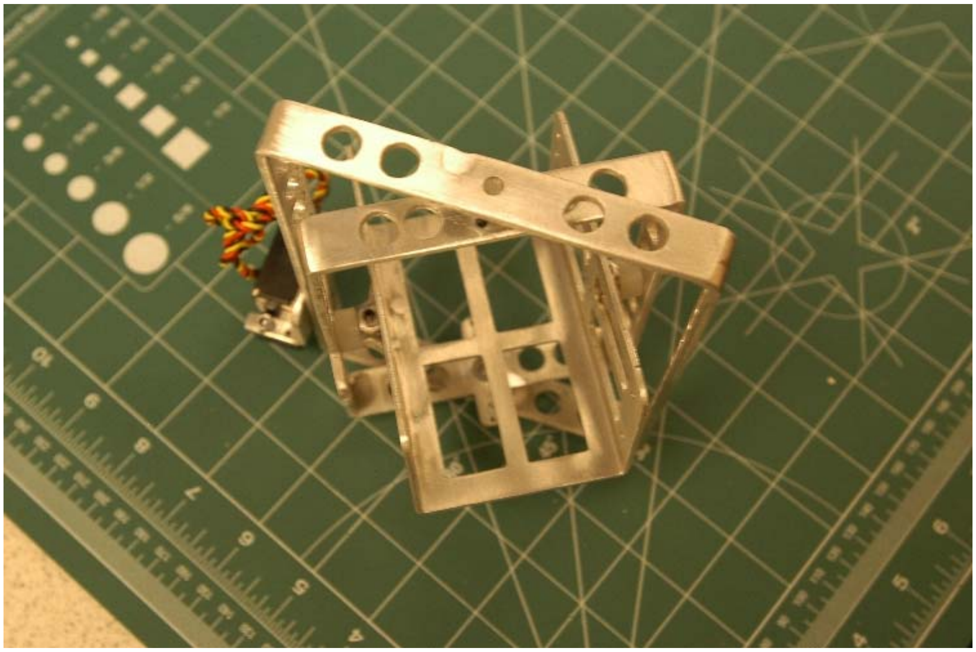


Figure 10 – AFRL/SNAR Gimbal Device

Figure 11 – Sanyo Color CCD camera

Figure 11 shows one of the cameras we use onboard our UAVs. This camera provides 520 lines of resolution and 22X zoom capability. The gimbal device shown in Figure 10 was built specifically for this camera. The camera is connected to an onboard video transmitter broadcasting at 2.4GHz on one of four possible channels. The matching receiver is used on the ground near the UAV flying operations site to receive the video signal.

In order to operationalize the UAVSAT tool, it would be useful to buy operational UAV systems and incorporate them into the tool. The system that we recommend is the Raven UAV system, shown in Figure 12. This system is currently being purchased by the US Army for small area surveillance.

Figure 12 – Raven UAV

Cadet Involvement

The cadets in the Software Engineering I and II courses learned how to develop a software system as part of a large development team. They learned project management skills, integration skills, programming skills, and deployment of systems skills. This year, for the first time, we incorporated Systems Engineering and Systems Engineering Management cadets into the class. Figure 13 shows the organizational structure of the project development team.

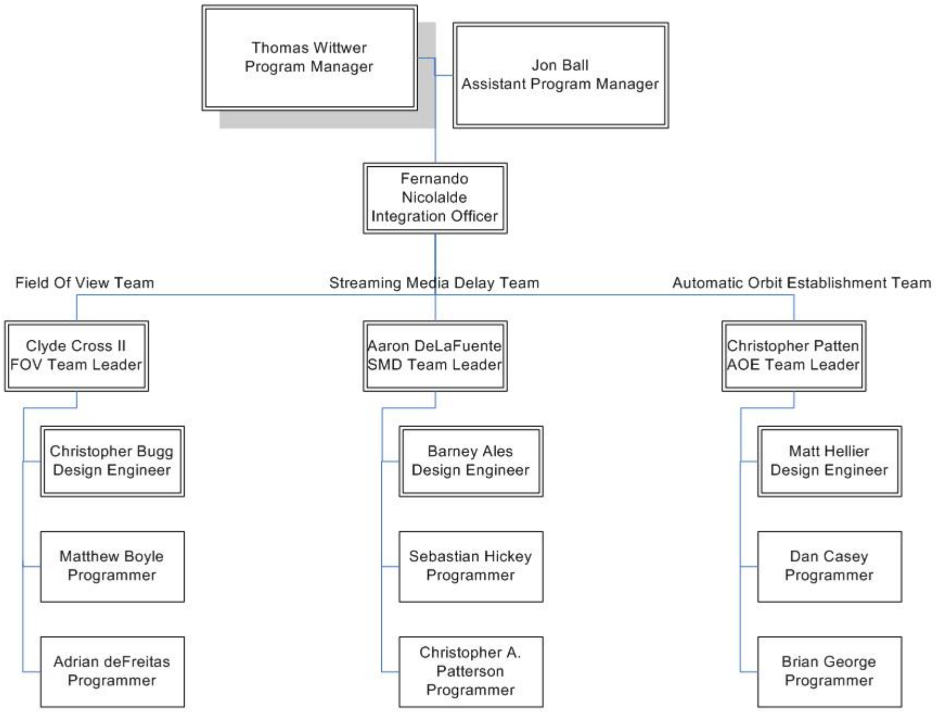


Figure 13 – Team organization

As shown in Figure 13, there was one overall Program Manager along with an Assistant Program Manager. The Program Manager was the one cadet majoring in Systems Engineering Management and this cadet did an outstanding job of managing the project. It worked out very well to have this cadet in this position. The Assistant was a Computer Science major and also did an outstanding job in this position. In fact, during the final two release briefings, the Assistant briefed the overall project due to an illness of the Program Manager.

The three development teams under the Program Manager were organized by functionality, as shown in the figure. Each sub-team had a team leader, a design engineer and two programmers. This year we used the Extreme Programming Agile Development Methodology and it was very successful. The programmers used “pair programming” allowing both programmers to have ownership in the software and stay up to speed on the development of the software. The design engineer was responsible for the development and maintenance of the design documents such as the Use Cases, Swimlane Diagrams, and Test Plans. The team leader coordinated all efforts and reported to the Program Manager during weekly progress reports.

Overall, the cadets enjoyed the project and the organization of the teams allowed for a very productive development environment. The UAVSAT project was very motivational for the cadets and they stayed very productive even during their last few weeks at the Academy.

Future Work

Future research should include efforts to remove the assumptions about the UAV's camera pointing straight down and that there is no pitch or bank. These simplifying assumptions were useful for our work on UAVSAT, but are unrealistic. The correct geo-location of the images coming from the UAV must account for pitch and bank of the UAV as well as the angle of the camera's gimbal. The pitch and bank information can easily be obtained from the UAV's telemetry stream. The gimbal angle must be obtained from the control software used to move the gimbal during flight. This has not been integrated into the current UAVSAT software.

Another possible feature for UAVSAT is to include a Tivo-like interface. This interface would store the images into a database along with the telemetry information used to geo-reference the image. Any pertinent information such as latitude, longitude, pitch, bank, roll, altitude, and gimbal bank angle would be stored along with the image. The Tivo interface would allow the user to go back to previously viewed images and display them in Google Earth. This would give the commander even more situational awareness during crisis situations.

Additionally, procedures for launch, standard loitering patterns, response and recovery of the UAVs should be developed. This is needed in order to operationalize the UAV in the Air Force. Training and education about the use of UAVSAT and the UAVs is needed if the system is to be deployed into a command center.

References

[1] Sward, Ricky E. Grant proposal to Institute for Information Technology Applications, Aug 2004.

[2] Microsoft Windows Media Server. Retrieved from http://www.microsoft.com/windows/windowsmedia on 19 August 2005.

[3] Cloud Cap Technologies. Retrieved from http://www.cloudcaptech.com on 19 August 2005.

[4] Silver Fox UAV. Retrieved from http://www.acrtucson.com/UAV/silverfox/ on 19 August 2005.

[5] Pointer UAV. Retrieved from http://www.aerovironment.com/area-aircraft/prod-serv/ptrdes.pdf on 19 August 2005.

Glossary

AFRL	Air Force Research Lab
AFRL/SNAR	Sensors Directorate of AFRL
AGL	Above Ground Level
CPU	Central Processing Unit
DTED	Defense Terrain Elevation Data
FOV	Field of View
GCS	Ground Control Station
MPEG-4	Lossy video compression algorithm
MySQL	Open source database program
PhP	Scripting language
SDK	Software Developer's Kit
UAV	Unmanned Aerial Vehicle
UAVSAT	UAV Situational Awareness Tool
USAFA	US Air Force Academy
XML	Extensible Markup Language

About the Institute

The Institute for Information Technology Applications (IITA) was formed in 1998 to provide a means to research and investigate new applications of information technology. The institute encourages research in education and applications of the technology to Air Force problems that have policy, management, or military importance. Research grants enhance professional development of researchers b y providing opportunities to work on actual problems and to develop a professional network.

Sponsorship for the Institute is provided by the Assistant Secretary of the Air Force (Acquisition), the Air Force Office of Scientific Research, and the Dean of Faculty at the U.S. Air Force Academy. IITA Coordinates a multidisciplinary approach to research that incorporates a wide variety of skills with cost-effective methods to =achieve significant results. Proposals from the military and academic communities may be submitted at any time since awards are made on a rolling basis. Researchers have access to a highly flexible laboratory with broad bandwidth and diverse computing platforms.

To explore multifaceted topics, the Institute hosts single-theme conferences to encourage debate and discussion on issues facing the academic and military components of the nation. More narrowly focused workshops encourage policy discussion and potential solutions. IITA distributes conference proceedings and other publications nation-wide to those interested or affected by the subject matter.