Georegistration of Imagery from Unmanned Aircraft Systems Using Ada

Ricky E. Sward
MITRE Corporation
1155 Academy Park Loop
Colorado Springs, CO 80910
1-719-572-8263
rsward@mitre.org

ABSTRACT

As Unmanned Aircraft Systems are used for an increasing number of applications, the challenge of precisely locating the geospatial position of imagery coming from the UAS is becoming more crucial. Through work that has been done in conjunction with the US Air Force Academy, Ada has been used to calculate the position and size of UAS imagery based on field-of-view calculations and elevation data. This paper describes the work that has been done this past year to georegister UAS imagery and how Ada has been used previously for this challenging task.

Categories and Subject Descriptors

D.2.11 [**Software**]: Software Engineering – *Software Architecture*

General Terms

Design, Languages, Image Processing

Keywords

Unmanned Aircraft System, Multi-Language Development, Software Architecture.

1. Introduction

Unmanned Aircraft Systems (UAS) [1] are increasing in importance in military and civilian applications and the diversity of the uses of UAS are limited only by our imaginations. For example, users have explored the idea of using UAS as mapping agents, in search-and-rescue missions, for motion picture filming, for land-mine detection, for bridge inspections, and even crop dusting [2].

As these roles for UAS increase, the challenge of determining the exact geospatial location of images coming from a UAS remains a crucial problem. For example, if a downed airman is seen in the video feed from a UAS, where exactly on the earth is that airman? The imagery from the UAS clearly shows the airman, but what is the geospatial location of that image? If the correct position on the earth can be determined for the image, then the correct position of the airman can be determined as well.

The process of finding the correct geospatial location of an image is known as *georegistration* [3]. The proper georegistration for imagery being taken by a UAS is challenging task because there is a delay between when the image was taken onboard the UAS and when the image is received on the ground. Because of this delay, the location of where the UAS was at when the image was taken is not the same as the location of the UAS being reported when the image is received on the ground. The delay in transmission of the image from the UAS to the ground is often variable making this problem even more challenging.

This paper discusses UAS telemetry and video systems, how our system captures images from the video, and how the images are georegistered on the ground. The Ada programming language has been used to determine the field-of-view calculation used in georegistration and we discuss this connection to Ada in the paper. Finally, we discuss how the images are displayed using Google Earth, our results and future work.

2. Unmanned Aircraft Systems

Unmanned Aircraft Systems (UAS) have become increasingly important in military and civilian applications. UAS can perform missions that are dull, dirty and dangerous for human pilots. UAS payloads vary depending on the mission and may include Electro-Optical (EO) daylight cameras, Infra-red (IR) night cameras, or even chemical spraying equipment.

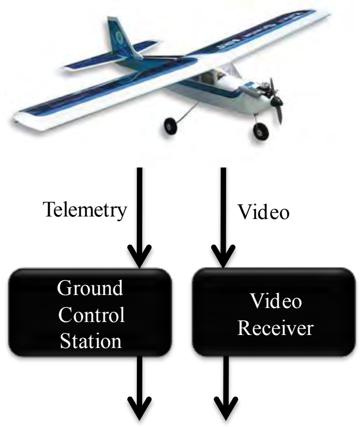

Figure 1 – UAS Telemetry and Video

For example, Figure 1 shows the Kadet Senior unmanned aircraft which includes the Piccolo [4] autopilot. The Piccolo provides telemetry of the UAS including the latitude, longitude, altitude and attitude angles of the aircraft. The telemetry is sent down a 900 MHz communications channel, which is separate from the

SIGAda'09, November 1 - 5, 2009, Tampa, Florida, USA.

video signal. The telemetry is received on the ground by the Piccolo ground control station and the position of the UAS is plotted on a map display. The ground control station also includes the capability to broadcast the telemetry to other computers connected via a network.

As part of the telemetry system architecture that we have developed, an application program is used to store aircraft telemetry into a database. Once a second, telemetry packets are captured from the ground control station, decoded and then stored into a row in the database. Telemetry information that is stored includes the GPS time, latitude, longitude, altitude, pitch, roll, yaw, and airspeed of the aircraft.

The Kadet Senior includes an EO daylight camera payload mounted on a two-dimensional gimbal providing pan and tilt camera movements. The video signal is sent down a 2.4 GHz analog transmitter communications channel. The video signal is received on the ground via a 2.4 GHz receiver and can be displayed on monitors for viewing. Due to the nature of the separate communications links for telemetry and video, there is a difference in the transmission times for the two links.

3. Capturing Video Imagery

Once the video has been received on the ground, software systems can be used to capture single still-frame images from the streaming video feed. The telemetry is also received on the ground and is saved to a database using an application program. In order to georeference the imagery, a program must be built that can access both the video images and the aircraft telemetry at the same time.

Figure 2 – VCE-PRO Application

3.1 Video Capture Card

This year our team used the VCE-PRO video capture card [5] to capture video frames from the Cadet Senior video system. The VCE-PRO card includes a C++ Software Developer's Kit (SDK) that allows the programmer to initialize the video capture card and display the streaming video frames onto an application window. It also allows a software system to capture video images from the video stream. Our team used the SDK to develop an application that initializes the VCE-PRO card and provides the user with a button to capture a frame from the stream video feed.

Figure 2 shows the VCE-PRO application. The frame around the image is showing the live video stream from the Cadet Senior. The button labeled "Capture Image" allows the user to capture a single image from the video feed and store it in a file in a specific directory. The name of the file is built by the application and is based on the GPS time taken from the aircraft telemetry.

3.2 Integrating with the Telemetry

In order to georegister the imagery from the video feed, both the captured imagery and the telemetry from the aircraft must be accessible by a single application program. We have developed this program, which is the VCE-PRO application shown in Figure 2. Via the video capture card, the application has access to the streaming video being received from the onboard camera. This application also connects to the Piccolo ground control station via the network and receives the streaming telemetry coming from the onboard autopilot. The Piccolo system provides a C++ SDK that allows application programs to connect to telemetry coming from the ground control station. The application now has access to the video imagery and the information about the location of the aircraft which will be used to perform georegistration of the imagery.

The application in Figure 2 allows the user to press the "Capture Image" button and store an image from the video capture card. Since the application also has access to the telemetry information including the latest GPS time, the name of the captured image can be built using the GPS time. The application polls for telemetry once per second and stores the telemetry into the database using the GPS time as a key field.

When an image is captured, the latest GPS time from the telemetry is used for the file name. This GPS time is then used to update the database row with the matching GPS time. The row is updated to store the name of the imagery file into the appropriate field. This saves the name of the imagery file along with the row of telemetry information that tells where the aircraft was at when the image was taken.

Because of the difference in transmission times between the video and telemetry communication links, the connection of the name of the captured image with the position of the aircraft when the image was taken will be inaccurate. The telemetry for the aircraft arrives before the video imagery, so the captured files will be associated with telemetry that does not accurately indicate where the aircraft was at when the imagery was taken. This is a well-known problem faced by all UAS systems. One potential solution will be discussed below.

4. Imagery Georegistration

The first step in georegistering video imagery is to use the telemetry information about where the UAS was at when the image was captured. Figure 3 shows the Cadet Senior and an image captured from the onboard camera. If we assume that the camera is pointing straight down from the aircraft and that the aircraft is flying straight and level, we can determine the center point of the captured image.

Figure 3 – Image from Cadet Senior

Using the two assumptions, we know that the image was taken directly under the aircraft, so the center of the image is at the latitude and longitude of the aircraft. If the camera is mounted in a fixed position on the aircraft pointing straight down, then the first assumption is valid. If the camera is on a gimbal as in the Cadet Senior, then the angle of the gimbal must be taken into consideration when determining the center of the image. The second assumption presents a bigger challenge. The aircraft is often pitching and banking during flight. This means that the location of the imagery will be inaccurate if the aircraft is not straight and level when the image is taken.

Figure 4 – Effect of transmission delays

In our application, the connection between the captured image and the telemetry associated with it in the database is inaccurate, as described in the previous section. This causes the georegistration of the image to be inaccurate as well. Figure 4 illustrates this problem. The aircraft on the right illustrates the position of the UAS as stored in the database row connected to the imagery file name. The grayed out aircraft on the left shows the actual position of the UAS when the image was taken.

In our experience with our application this year, we have found that the georegistration error is approximately 100 meters. To overcome this problem, the user must do a manual georegistration of the image using reference imagery. This is done by placing the captured image onto the reference imagery and aligning it with known features such as the edge of the runway or roads.

4.1 Field-of-View Calculations

After the center of the image is determined by using the latitude and longitude of the aircraft, the positions of either the edges or corners of the image must be determined. Our visualization tool requires the positions of the edges to be given, so we needed to determine the latitude and longitude of the edges. Several factors are used to determine this part of the georegistration.

To georegister the edges of the image, the field of view (FOV) [6] from the camera must be determined. FOV is based on the focal length of the camera, the altitude of the aircraft and the elevation of the ground under the aircraft. The focal length is specific to each camera and can be found in the manufacturer's documentation. The altitude of the aircraft is obtained from the Piccolo telemetry as stored in the database.

The elevation data is determined from Defense Terrain Elevation Data (DTED) [7] files. DTED information has been built for certain areas of the earth and gives an average elevation value for a specific latitude and longitude. The altitude and elevation are used to determine how far above the ground the aircraft was when the image was taken, which provides the distance from the camera to the subject.

The FOV calculation returns the horizontal and vertical dimensions of the image in feet. The georegistration of the edges of the image are then determined by splitting these dimensions in half and then determining the latitude and longitude of the point which is the correct distance from the center point of the image.

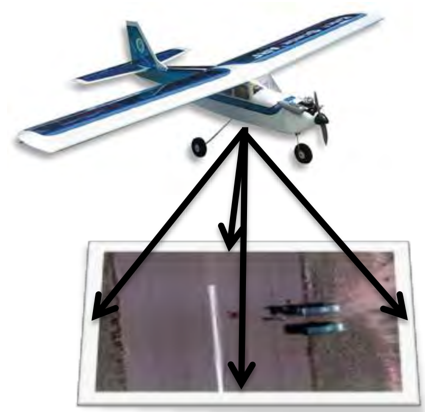

Figure 5 – FOV Calculations

The accuracy of the FOV calculation is affected by the pitch and roll of the aircraft. These aircraft movements not only cause the position of the center point of the image to move, but also cause the image to be distorted as it gets farther away from the point directly under the aircraft. Our application did not consider pitch and roll in the georegistration of the imagery.

4.2 Stabilized Gimbal

As discussed above, the accuracy of the image georegistration is affected by the pitch and roll of the aircraft. In order to minimize this inaccuracy, our application uses the pitch and roll information from the telemetry to stabilize the gimbaled camera. If the aircraft pitches up five degrees, then the application program sends a command to the gimbal to tilt down five degrees. This zeroes out the pitch-up action from the aircraft and keeps the camera pointing straight down. The roll action of the aircraft is also zeroed out through our application. The telemetry of the aircraft is examined 10 times per second to stabilize the gimbaled camera.

4.3 Using Ada for Georegistration

In 2006, the cadets at the US Air Force Academy developed an Ada function that calculates the proper FOV for the camera used on the Cadet Senior UAS [8]. This Ada function was used again this year in our application to determine the locations of the edges of the imagery. The Ada function is built as a dynamically linked library and is called from the C++ image capture application.

Given the altitude of the aircraft and the elevation of the ground under the aircraft, the Ada FOV function uses the focal length of the camera to determine an offset for the edges of the image. When added to the latitude and longitude of the center point, these offsets determine the points along the edges of the imagery that are the correct distance from the center point of the image. These offsets are stored in the database in the row with the telemetry and name of the captured image.

5. Displaying Imagery in Google Earth

In order to visualize the location of the georegistered, captured images, we have used Google Earth [9] to show the images and their location on the earth. All the information that is needed for this visualization is stored in the database row for the image. The name of the image is stored in the row that holds the telemetry associated with the image and the offsets for the FOV determination are also stored in this row.

Figure 6 – Georeferenced Image in Google Earth

Figure 6 shows an example of a georegistered image that is being displayed using Google Earth. As you can see, the georegistration is not accurate. For example, the edges of the runway in the Google Earth image do not align with the edges in the captured image. This is because of the georegistration error caused by the delay in the transmission times for the telemetry and video. In Google Earth, the user is able to manually georegister the image and align the edges and major features.

Each image is time stamped so that different images can be displayed at one time instead of all the images being displayed at all times. The position of the aircraft as taken from the telemetry is also displayed on Google Earth to show the three dimensional position of the aircraft when the image was taken. Only the rows from the database that have an image name associated with them are pulled from the database and displayed on Google Earth. This limits the number of aircraft positions that are displayed on the visualization tool.

6. Future Work

As discussed above, the georegistration of the images is inaccurate because of the delay in transmission times for the imagery and the telemetry. By associating the captured image with the position of the aircraft after both of these reach the ground, the georegistration of the image is offset. As you can see in Figure 6, the georegistration is not off by a significant amount. In fact the average error was less than 100 meters in our experiments due to the slow airspeed of the aircraft.

Future work in this area is to include an onboard processor that would associate the captured image with the telemetry from the autopilot. This would be done onboard the aircraft and would eliminate the problem with the difference in transmission times. Once the telemetry is associated with the captured image, both would be sent down to the ground for visualization.

Another area for future work is to improve the Ada function that determines FOV calculations. Currently, the FOV calculation accounts for the focal length of the camera and the distance above the ground that the aircraft is flying. To improve the georegistration of the imagery, the FOV function should account for the pitch, roll, and yaw of the aircraft. Dr. Wayne Brown from the Air Force Academy has been working with a cadet to improve this georegistration process [10]. Their work includes ray tracing from the camera to the point on the ground to determine the edges of the captured image. One future goal is to rewrite the Ada FOV function to include Dr. Brown's work and improve the georegistration of imagery using Ada.

7. References

[1] Unmanned Aircraft Systems. Retrieved from http://en.wikipedia.org/wiki/Unmanned_Aircraft_System

[2] Goth, Gregory. Autonomous Helicopters. Communications of the ACM, June 2009, Vol 52, No 6, pages 18 – 20.

[3] Georegistration. Retrieved from http://en.wikipedia.org/wiki/Georegistration

[4] Piccolo autopilot from Cloud Cap Technologies. Retrieved from http://www.cloudcaptech.com/

[5] ImperX VCE-PRO video capture device. Retrieved from http://www.imperx.com/frame-grabbers/vce-pro

[6] Field of View (FOV) calculations. Retrieved from http://en.wikipedia.org/wiki/Angle_of_view

[7] Defense Terrain Elevation Data (DTED) information. Retrieved from http://en.wikipedia.org/wiki/DTED

[8] Final project report for Computer Science 454, Software Engineering, Spring 2006. US Air Force Academy.

[9] Google Earth. Retrieved from http://earth.google.com/

[10] Brown, Wayne. Computer Science 499 Final Report Spring 2009. US Air Force Academy