

Automatic mission planning algorithms for aerial collection of imaging-specific tasks

Paul Sponagle^a and Carl Salvaggio^a

^aRochester Institute of Technology, Chester F. Carlson Center for Imaging Science, Digital Imaging and Remote Sensing Laboratory, 54 Lomb Memorial Drive, Rochester, NY, 14623-5604, USA

ABSTRACT

The rapid advancement and availability of small unmanned aircraft systems (sUAS) has led to many novel exploitation tasks utilizing that utilize this unique aerial imagery data. Collection of this unique data requires novel flight planning to accomplish the task at hand. This work describes novel flight planning to better support structure-from-motion missions to minimize occlusions, autonomous and periodic overflight of reflectance calibration panels to permit more efficient and accurate data collection under varying illumination conditions, and the collection of imagery data to study optical properties such as the bidirectional reflectance distribution function without disturbing the target in sensitive or remote areas of interest. These novel mission planning algorithms will provide scientists with additional tools to meet their future data collection needs.

Keywords: UAS, sUAS, flight planning, calibration, reflectance, structure-from-motion, BRDF, inspection

1. INTRODUCTION

The small unmanned aerial system (sUAS) market has boomed in recent years with estimates suggesting the market will be worth between \$6 and \$8.4 billion globally by 2018.¹ A variety of payloads are available for the sUAS market to meet an ever-increasing demand for data. Scientific payloads can include traditional cameras for photography or videography, multi- and hyper-spectral sensors, light detection and ranging (LiDAR), thermal infrared, and many other image-collection sensors. Many sUAS permit aircraft to be flown by a remote pilot, autonomously by uploading a sequence of waypoints, or with an on-board computer that will react to changing environmental as well as mission-specific situations. Commercial ground control/mission planning software applications are available to the consumer, but these systems are customized to meet the requirements of the aerial photography community and lack features useful for more specific scientific tasks. Remote pilots wishing to fly complex flight profiles to meet technical requirements will likely find themselves struggling to enter many waypoints manually, and a researcher may need to fly similar flight profiles at various different locations. This paper documents algorithms that produce three specific flight profiles, deemed to be of particular use to researchers, and provide the end-user with a way to create their own scripts to generate flight patterns particular to their research.

2. BACKGROUND

The sUAS technology boom has triggered a surge of mobile applications that can be used to control these unmanned aircraft vehicles. Many of these applications enable a user to enter waypoints on a touch screen, permitting the aircraft to fly a preprogrammed mission. Many applications even permit the user to specify a polygon on a touchscreen, a flying altitude, and an image overlap, outputting a flight plan to acquire image data of the entire area inside the polygon. While these tools are useful, they fall short in several areas required by imaging scientists:

Further author information: (Send correspondence to P. Sponagle or C. Salvaggio)

P. Sponagle: E-mail: pas5923@rit.edu

C. Salvaggio: E-mail: salvaggio@cis.rit.edu, Telephone: +1 (585) 475-6380

1. Many imaging tasks require a very large number of waypoints to be entered manually. A small modification to the flight plan, like aircraft altitude, may require all of the points to be adjusted, which is excessively time consuming and error prone. At the fine resolutions that these systems might collect, latitudes and longitudes coordinates often are required to be specified using up to seven decimal places.
2. Existing photogrammetry flight profiles, those used to collect data for the creation of a topological mapping product, that exist in current commercial mission planning applications do not permit the user to specify a location for calibration targets that may need to be overflown several times, depending on weather conditions. These calibration panels that are typically required to be used when surface reflectance values are required. These reflectance data are often more useful than radiance or raw digital count in remote sensing applications since reflectance values are a target specific property that is independent of the atmospheric and illumination conditions under which the data was collected. Imaging and remote sensing scientists apply various band or channel indices to solve particular agricultural evaluation tasks, such as the normalized differential vegetation index (NDVI), using these calibrated reflectance values. Reflectance values also permit the application of anomaly detection and sub-pixel target detection algorithms under a variety of illumination and meteorological conditions.
3. Typical application tools are either limited to terrain following or constant altitude flights and are usually used with the camera fixed and pointing in the nadir direction. These profiles can be problematic due to the tendency for structures to occlude themselves when viewed from nadir, resulting in a reduced number of tie points in structure-from-motion (SfM) applications. If wanting to inspect a structure, a sUAS would need to fly at various altitudes and varying camera angles. Figure 1 shows a pipebridge with many trusses that generate occlusions. Varying view angles are also necessary when attempting to make in-field bidirectional distribution reflectance function (BRDF) measurements, where it is necessary to stare at the same ground spot regardless of platform location.² Inspection of wind turbine blades also require precise and automated flight plans to prevent damaging expensive and sensitive components.³



Figure 1. A pipebridge crossing the Cayuga-Seneca canal illustrates how trusses can occlude features of a bridge.

sUASs are becoming more common for infrastructure inspection. While long stretches of gas pipelines can be inspected from the inside of the pipe, using a pipeline inspection gauge with ultrasound sensors, to find structural anomalies,⁴ a sUAS operating near the pipe also has the potential to detect anomalies in the pipe by using a combination of sensors to detect uneven thicknesses of a pipe that might indicate the need for maintenance or replacement. These sUAS platforms have the potential to offer rapid inspection, while minimizing risk to inspectors, when an inspection of a pipe's exterior requires accessing hard to reach locations, such as a river crossing or gorges. Another advantage of inspecting the pipe's integrity from the exterior also means that the inspection will not impede the transportation of natural gas.

A variety of applications are available for iOS and Android mobile platforms that can operate several commercially available sUAS platforms. The ground control system used for planning by the Rochester Institute of

Technology's (RIT) UAS Research Laboratory is UgCS. The UAS Research Laboratory offers researchers the use of the Da-Jiang Innovations' (DJI) Spreading Wing S1000, Matrice 100, and Matrice 600, all of which are compatible with UgCS. The S1000 is capable of carrying large payloads, but gimbal control is yet to be automated. The Matrice 100 is equipped with the Zenmuse X3 camera that provides full gimbal control manually or through UgCS. The Matrice 600 platform will serve as the testbed for new payloads under development. Researchers also have access to low-cost platforms, such the Parrot AR Drone 2.0 and DJI Phantom, for algorithm testing to ensure that flight plans can be successfully uploaded and flown.

UgCS flight plan segments are similar to those found in similar applications used for sUAS control. Segments can be as simple as a waypoint providing latitude, longitude, and height, or a more complex pattern that generates multiple waypoints. Available UgCS segments include:

1. Takeoff and landing. Depending on the platform, many applications permit the application to conduct automatic takeoff and landings.
2. User-specified waypoints. The ability to specify the latitude, longitude, and altitude of multiple points that the user requires. Some mobile applications will allow the user to enter waypoints rapidly using a touch screen, but entering evenly spaced points is a cumbersome process due to the required precision. Entering a very large number of points, required of longer missions, amplifies this problem.
3. Photogrammetry tool. This feature determines waypoints necessary to take photos at a specified ground sampling distance (GSD) within a user-specified polygon. The aircraft adjust its height to obtain the specified GSD when provided with the sensor's focal length and pixel pitch. A digital elevation model is incorporated into UgCS to determine the aircraft's height above ground. Waypoint spacing is determined based on user-specified overlap. Figure 2 show user inputs required to generate this flight profile, along with a visualization of this mission.

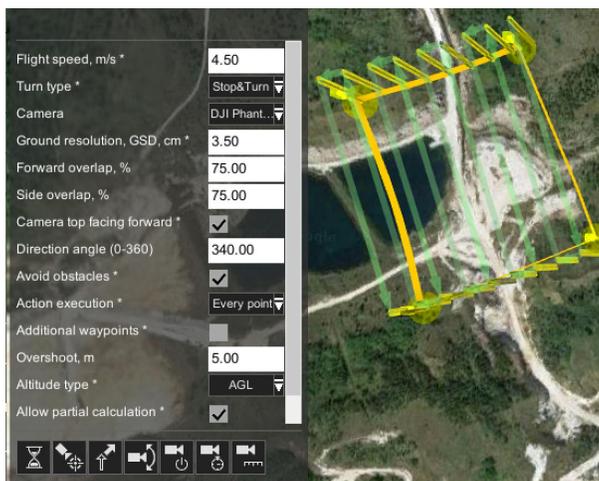


Figure 2. Image captured from UgCS shows an example of the photogrammetry tool with necessary user inputs. Icons at the bottom of the screen show various camera actions available to the user.

4. Area scan. This feature is similar to photogrammetry tools, except that the aircraft flies at a specified altitude above mean sea level (AMSL) and permits the GSD to vary.
5. Circles. The user specifies the center, radius, number of points and altitude and the application determines the waypoint locations to approximate a circle.
6. Perimeter. The user specifies a polygon and altitude, and the application will control the aircraft, such that it follows the edge of the polygon.

Camera actions permit users to control when the camera acquires an image, the zoom level, orientation, time delay, as well as a variety of other features. These actions are associated with one or more of the above segments in the flight plan. UgCS uses the icons displayed in Figure 3 to permit the user to attach camera actions to segments.



Figure 3. UgCS offers seven camera actions from left to right: Wait, point of interest tool, yaw, set camera angles, still/video, time interval between shots, distance between shots

During a brief flight, the lighting conditions can vary rapidly due to cloud movement. Low clouds, such as cumulus, can produce dramatic effects, casting a shadow on the scene with a clearly visible edge. High clouds are subject to high velocity winds and tend to diffuse light. Perhaps the best example is an airliner’s contrail crossing the path of the sun. In a matter of a couple seconds, the scene will shift from being primarily illuminated by direct solar irradiance to a diffuse downwelling radiance.⁵ Cirrus clouds will have the same diffusion effect, but the change is gradual. The photogrammetry tool is convenient during consistent lighting conditions, but for agricultural purposes, sUAS platforms need to fly over calibration panels so that surface reflectance can be calculated using the empirical line method.⁶⁷ Variable lighting conditions will introduce errors when calculating these reflectance values. If the flight occurs when there are variable lighting conditions present, then it is often necessary to revisit the calibration panels to compensate for these conditions several times during a given flight. The existing method of visiting these calibration panels involves the remote pilot assuming manual control of the aircraft, flying to the calibration panels and then resuming the flight. Ensuring the target is centered in the frame is not always an easy task for the pilot, as the pilot may not have a live video feed from a spotting camera on the sensor.

Another useful pattern absent from these applications is a hemispherical pattern. A hemispherical pattern is useful for measuring bidirectional distribution reflection functions of large-scale targets, as well as creating point clouds using SfM of intricate structures. For this pattern to be useful, the look down angle of the sensor must be adjusted while the aircraft is continuously oriented towards the center of the image. These flight profiles have been used in previous experimentation for the purposes of creating a UAS-based goniometer where ‘mAngle’ software permits flights to be planned and executed in 30 minutes, with 20 minutes was allocated for preparation.² Increased automation of the flight planning process will help reduce this preparation time further. To prevent blurring, the aircraft should be kept as stationary as possible during image acquisition. Some instrumentation may require longer dwell times, increasing the length of the flight. Experience with RIT’s DJI Matrice 100 has shown that uploading many waypoints increases the upload time. There is also a limit to how many waypoints the aircraft’s autopilot can store. Exceeding this value requires that the flight plan be divided into two or more segments.

Bridge inspection is another frequent task required of sUASs that these commercial flight control applications are not optimized for. Inspecting a bridge involves a back and forth motion and altitude adjustments while acquiring imagery. This imagery can be used to create a point cloud using SfM techniques, recording video, or taking several still images from multiple angles. As with the hemispherical pattern, the camera’s gimbal must be adjusted to ensure that the bridge is always in frame. Manually flying close to a bridge adds risk, using automated flight control can significantly reduce the risk of colliding with the structure.

3. METHODS

UgCS was the primary ground control application used for this work. UgCS saves flight plan files in ‘.xml’ format. By analyzing the ‘.xml’ file line by line, the necessary fields can be replicated using ‘.xml’ capabilities residing within MATLAB to create the ‘.xml’ document, node structure, and various parameters.

The saved flight plan format consists of a header and footer common to all files. These sections describe data critical to the flight planning process, such as home location and altitude, height to fly when returning to home,

preferred altitude type (height AGL vs AMSL), what to do if GPS is lost, and what to do if the communications link is lost, etc.

Beyond this, the file is broken down into segments. Each segment may consist of multiple camera actions. A flight plan was created in UgCS using every segment and camera action that UGCS is capable of producing. A MATLAB function was created for each segment and action that would append action nodes to segment nodes, and segment nodes to the main document. Figure 4 shows how each of these elements are structured.

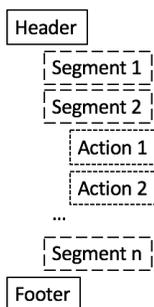


Figure 4. Functions created with MATLAB build a flight plan by first creating a header, adding segments and adding actions to segments as required.

The focus of this work is on three flight profiles that I have determined to be useful to imaging scientists: Agricultural, hemispherical, and bridge inspection. Users with specific needs are now able to create their own flight plans using this process.

3.1 Agricultural Flights

Typical agricultural flights, as shown in Figure 2, require an aircraft to fly a raster-like pattern across a user-specified area while taking images at nadir (i.e. directly below the aircraft). Since imaging scientists are often concerned with surface reflectance, it may be necessary to fly over and capture images of calibration panels a number of times over the duration of the collection. Once the images are obtained, surface reflectance values can be calculated by applying the empirical line method.⁵ The following algorithm accomplishes the task of collecting images while the aircraft is shooting at user-specified intervals set on the camera:

- Find geo-referenced imagery. National Agricultural Imagery Program (NAIP) imagery is available from '<http://earthexplorer.usgs.gov>' and offers one meter resolution over a large area. This will permit users to specify a region of interest by clicking on an image as opposed to tabulating latitudes and longitudes.
- Reproject the image. Project the image onto a Mercator projection using ENVI software. This projection reproject the image such that all lines of longitude are vertical and all lines of latitude are horizontal. This simplifies calculations when selecting points on an image.
- Reduce size. Full NAIP images are not convenient to use since the coverage area is much greater than what a sUAS can cover. Take a subset of the image while still in ENVI. Take note of the latitude and longitude of all four corners as this will be necessary to convert from row/column to latitude/longitude. Then save the image as a '.tiff' format.
- Import image into MATLAB. Simply use the 'imread' command to read in the image and use latitude and longitude of the top, bottom, left and right of the screen as variables.
- Select Region of Interest (ROI). This can be done by either clicking on the image or specifying the latitude and longitude.

- **Direction.** By default, the algorithm applies principal component analysis (PCA) to determine the prominent direction of the ROI. If the ROI is an elongated rectangle then by default, the aircraft should fly flight paths parallel to the long edges of the rectangle. If the user requires specific track angles, then this step can be commented out and replaced with direction specified in degrees true north. PCA is a tool used in pattern recognition and image analysis. For pattern recognition, when provided with a list of 'x' and 'y' coordinates, such as the latitude and longitude of a region of interest, the directionality of the points can be found by creating a two-dimensional array consisting of a column of 'x' values and a column of corresponding 'y' values. Since this two-dimensional array consists of 2 columns, there are two eigenvectors. The first eigenvector, or Principal Component (PC), describes the principal direction of the points and the second PC is orthogonal to the first PC. The first PC corresponds to the vector with the highest eigenvalue. Each PC can be represented on a 2-dimensional plane as a vector and its angle measured clockwise from the vertical axis, corresponding to the heading and expressed in degrees from true north, minimizing the number of legs required to fly.
 - **Determine leg separation.** A leg represents a straight line on the Earth's surface that the aircraft will fly. The distance between legs is determined based on the necessary overlap, as specified by the user and the swath width. The swath width is calculated based on the sensor size, focal length, sensor orientation and altitude at the center of the frame. Leg separation can be determined by multiplying the overlap by the swath width, as shown in Figure 5 and Equation 1.

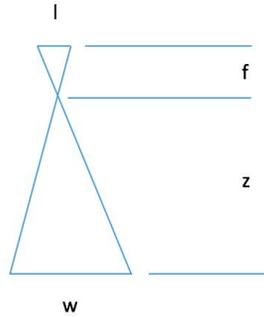


Figure 5. The law of similar triangles is used to calculate the projected swath width of the image assuming that the ground is level. Once the swath width is known, the leg spacing can be determined based on the required overlap. f = focal length [mm], l = sensor dimension [mm], z = altitude above ground [m], w = swath width of the image [m]

$$\Delta y = Xw = \frac{Xlz}{f} \quad (1)$$

- Δy = separation [m]
 - X = User-specified overlap [-]
 - w = swath width [m]
 - l = edge length of sensor (mm)
 - z = height above ground [m]
 - f = focal length [mm]
- **Plot initial points.** Find the center of the ROI and plot points orthogonal to the direction of flight using Equation 2 and Equation 3. (Note that all angles are in radians and not degrees). Continue finding orthogonal points until off the frame. To get the points in the direction of flight this procedure after adding 90 degrees to the bearing.

$$lat2 = \sin^{-1}(\sin(lat1) * \cos(d/R)) + \cos(lat1)\sin(d/R)\cos(\theta) \quad (2)$$

$$long2 = long1 + \text{atan2}[\cos(d/R) - \sin(lat2), \sin(\theta)\sin(d/R)\cos(lat1)] \quad (3)$$

- lat1 = Latitude of the current point [radians]
 - lat2 = Latitude of new point [radians]
 - d = Distance [km]
 - R = Radius of the Earth (km)
 - θ = Bearing [radians]
 - long1 = Longitude of original point [radians]
 - long2 = Longitude of new point [radians]
- Create waypoints. Starting at the center of the plotted points, add 90 degrees to the heading and plot points at distances determined in the previous step. Since the camera's array is not a square, the distance to the next point will be calculated as before, except the other sensor dimension must be used. Repeat until off the frame and then add 180 degrees to the heading and repeat until off the other side of the frame.
 - Order the waypoints. The waypoints are not in a logical order for the aircraft to fly, so they need to be re-ordered. This is easily done when sensor positions are calculated along each leg. Simply flip the positions on every other leg.
 - Remove points outside of ROI. MATLAB has a function called 'inpolygon' that will remove all points outside of the ROI. At this point all points can be expressed in either xy or latitude and longitude coordinates. Functions for changing between coordinate systems were created based on linear interpolation using EQ 4 and 5.

$$x = \left(\frac{long - A}{B - A} \right) w \quad (4)$$

$$y = \left(\frac{lat - D}{C - D} \right) h \quad (5)$$

- x = x-coordinate on the image measured from left to right [pixels]
 - long = longitude of a point [degrees]
 - A = The line of longitude corresponding to the left side of the image [degrees]
 - B = The line of longitude corresponding to the right side of the image [degrees]
 - w = The width of the image [pixels]
 - y = y-coordinate on the image measured from bottom to top [pixels]
 - lat = latitude of a point [degrees]
 - C = line of latitude across the top of the image [degrees]
 - D = line of latitude across the bottom of the image [degrees]
 - h = height of the image [pixels]
- Reduce points. There are a finite number of waypoints that can be uploaded to a sUAS. For the Matrice 100, the limit is 99 points. To reduce the number of waypoints, only the endpoints of each leg are needed, so all points between these points can be removed.
 - Calibration panels. The user will specify the number of required visits to the calibration panels required. If only two trips to the calibration panels are required, then the aircraft will takeoff and immediately fly to the calibration panels before starting the route and then revisit the panels after completion of imaging the ROI. If additional visits to the panels are required, then the algorithm calculates the total time required over the entire ROI and creates breakpoints to permit the aircraft to image the calibration panels. If only one extra trip is required then the accumulated flight time to fly over the ROI can be split into two equal halves. Once the calculated interval is exceeded, the aircraft will revisit the panels at the end of that leg and return to the start of the next leg.

3.2 Hemisphere

A hemispherical pattern is particularly useful for 3D reconstruction using SfM software, such as Pix 4D Mapper, and for measuring bidirectional reflectance distribution function (BRDF). In either case, the target must be centered in each frame for best results. The following algorithm will generate a hemispherical flight plan:

- User specified data. The latitude, longitude, and the elevation of the target must be known. The latitude and longitude can be determined based on online maps. The elevation can be determined using the aircraft's GPS by setting it on level ground near the target since digital elevations are based on averages over a wider area. The speed is not critical, but faster speeds make it difficult for the aircraft to position itself at a waypoint. The radius of the hemisphere is also user-specified.
- Determine a safe waypoint. Obstacles may be present between the takeoff point and the target's location, so it is a good practice to climb to a specified altitude and position before starting the hemispherical pattern.
- Start at the top. All altitudes of waypoints in the sphere must be calculated in height above mean sea level, since the ground may not be level. The first point is directly above the target at an altitude that is calculated by adding the radius of the hemisphere to the elevation of the target. Set the camera look down angle to 90 degrees.
- Fly concentric circles. Fly North to the next waypoint. Adjust the altitude to the required zenith angle and use point at POI action to yaw the aircraft towards the target and take a single frame. Fly concentric circles until the necessary zenith and azimuth angles are covered, as defined in Figure 6. Adjust the camera look down angle at the start of each circle. The look down angle is the angle of the sensor below the horizon, as shown in Figure 7.

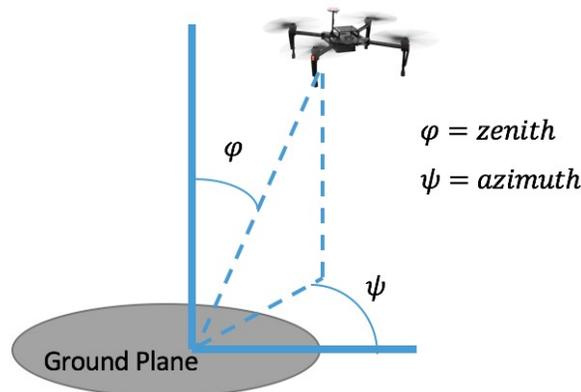


Figure 6. Image shows the zenith angle measured from zero and azimuth angle measured from an arbitrary direction. When determining the position in the hemisphere, the angle is simply the bearing, which is the negative of azimuth as measured from true north.

- Split waypoints. It may be necessary to generate multiple files depending on the number of waypoints required. The algorithm permits the user to specify the number of points that can be uploaded to their specific aircraft. If the aircraft's capability is exceeded, then multiple flight plan files are created.
- Manual control. At the end of the flight, the aircraft will enter a hover. This permits the remote pilot to assume manual control and safely land the aircraft. This reduces the potential for the aircraft to hit an obstruction between the target and the landing site. The user can add additional waypoints as required for safety purposes.

The algorithm will produce waypoints and camera angles as illustrated in Figure 8.

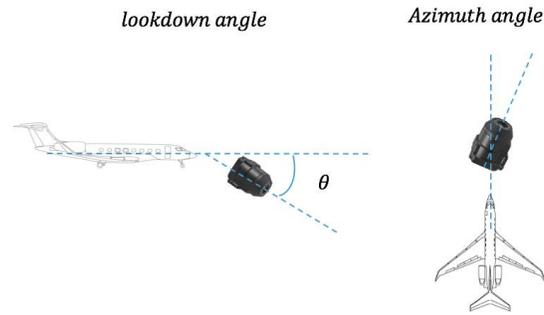


Figure 7. Look down and azimuth angles

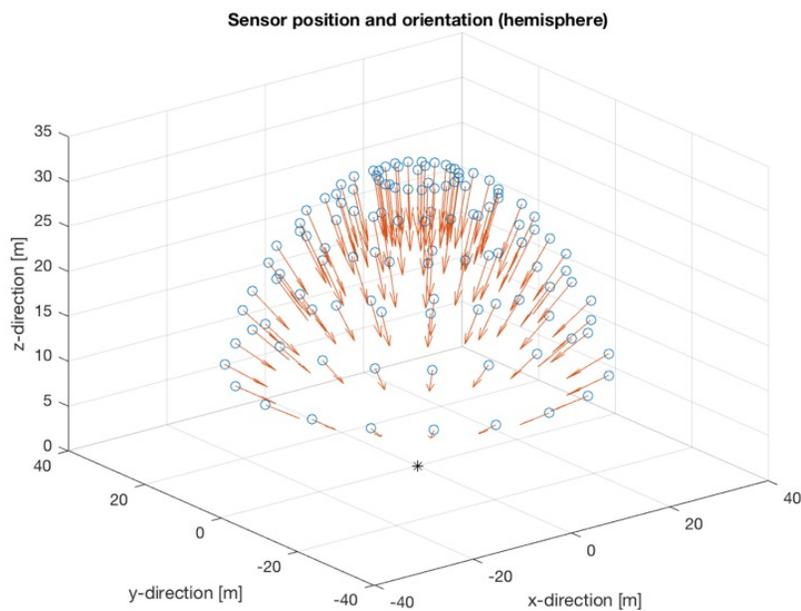


Figure 8. UAS position and gimbal angles always keep the sensor pointed at the base of the hemisphere

3.3 Bridge Inspection

Many bridges have a truss-type construction where the trusses occlude much of the bridge as shown in Figure 1. To resolve these occlusions, multiple images must be taken from multiple vantage points and camera angles. It is important for the focal plane array to be as close as possible to orthogonal from the bridge, as any change to camera azimuth or aircraft yaw will cause the bridge's resolution to be inconsistent across the frame. The autopilot of the Matrice 100 does not accept camera angles that look upward, despite the physical capability of the gimbal do so manually, so any negative lookdown angles are set to zero. If the aircraft does fly below the level of the bridge, the camera will need to rely on its field of view to resolve occlusions, as opposed to adjusting the camera look down angle. Attempting to extend the profile beyond a semi-circle also risks interference of the GPS signal by the bridge.

A flight profile, resembling a circular tunnel surrounding the bridge (see Figure 12) that adjusts the camera look down angle, meets these requirements and is created using the following algorithm:

- User-specifications. The user must specify the endpoints, ground speed, angles, elevation of the bridge, and the radius of the flight path.
- Calculate end points. If viewed from either end of the bridge, the aircraft will fly a semi-circular flight path consisting of evenly spaced points. An example of the end points and the flightpath are displayed in Figure 9.

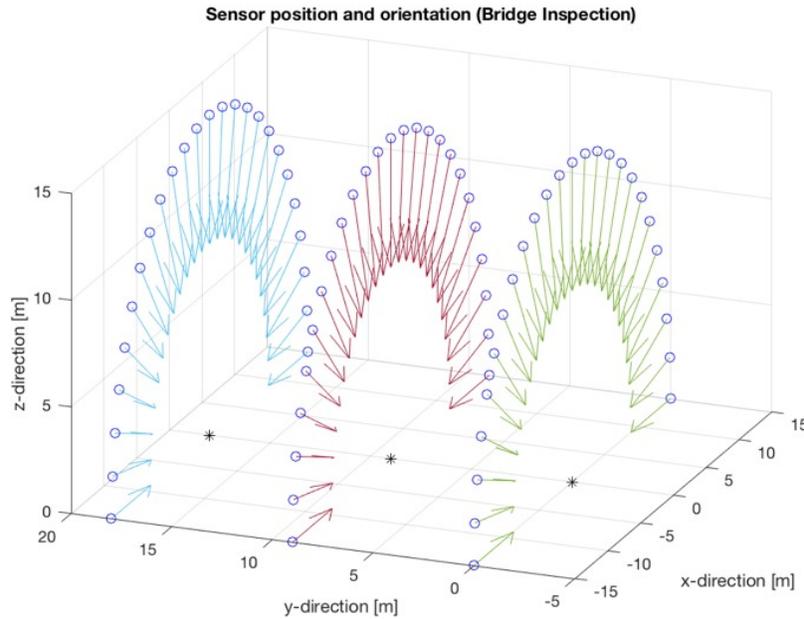


Figure 9. UAS position and gimbal angles always keep the sensor pointed along a linear path denoted by black asterisks

- Calculate look down angle. The look down angle is simply 90 degrees less the zenith angle from vertical, as measured from directly above the bridge. This needs to be calculated for each waypoint.
- Calculate yaw angle. The aircraft must orient itself so that the frame is parallel to the direction of the span. The yaw angle will be either 0, 90, 180, or 270 degrees, as measured from the aircraft's track. The angle can be selected by noting a pattern of the required yaw. When passing along the apex, the yaw angle should be either 90 or 270 degrees, such that bridge is parallel to the longer edge of the camera's frame.

4. RESULTS

All flight profiles discussed have successfully been loaded into UgCS as can be seen in Figures 10, 11, and 12. The agricultural flight plan is the only profile yet to be flown due to a lack of opportunity, but testing is slated for spring 2017. Both the hemispherical and bridge inspection flight profiles have been loaded on at least one sUAS platform and successfully executed.

All functions used to create these flight plans along with sample scripts are available on Github (https://github.com/paulSponagle/UGCS_MATLAB_scripts) and available for public use with the caveat that the remote pilot is ultimately responsible for verifying a safe flightpath and the safe operation of their aircraft.

Figures 13 and 14 are examples of two point clouds generated using the hemispherical and bridge inspection algorithms. Figure 14 displays intricate details that can be seen along the side of the pipebridge structure, whereas the point cloud in figure 15 shows less detail along vertical surfaces. In the nadir image, gray points appear to be attached to the structure. These points represent background that often get attached to thin

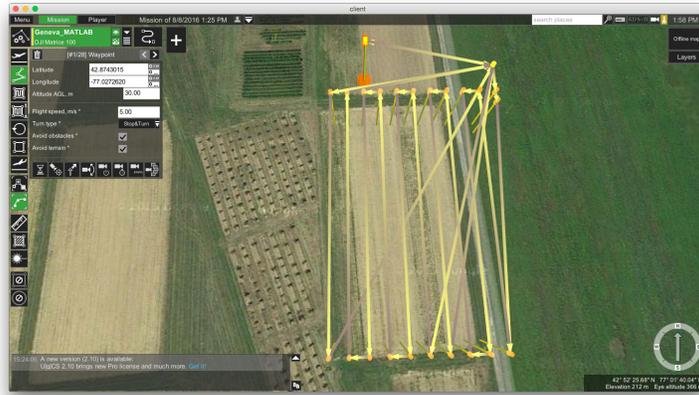


Figure 10. Screen capture of UgCS showing a calibration panel flight profile. Note, panels would be situated in the top right corner of the image.

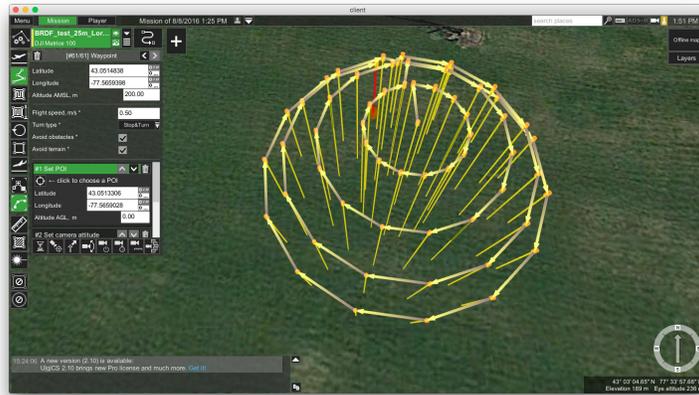


Figure 11. Screen capture of UgCS showing a hemispherical flight profile.

members and can be removed by editing the point cloud. Figure 14 had more editing to remove much of this type of noise. There are several reasons the reconstruction created from images taken at nadir is worse than the one created using the algorithm. The primary deficiency is caused by the pipebridge's self-occlusions that cannot be resolved when imaging at nadir. The nadir reconstruction was also taken from a further distance away from the bridge, due to the presence of obstructions, resulting in a lower GSD and fewer tie points. From experience, it is also interesting to note that flying closer to a structure that has a repetitive pattern is not always better as tie points may be attached to the wrong instance within the pattern. This minimum altitude restriction is a common issue when flying constant altitude flights. The nadir point cloud was also constructed using fewer images at an overlap setting of 80%, in order to mimic typical collections from nadir. While additional photos would have improved results, the self-occlusions would still be present. Imaging from an angles orthogonal to a surface allows the greatest number of tie points to be generated resulting in a better point cloud. A constant altitude flight looking at nadir will typically have fewer tie points on vertical surfaces compared to level surfaces. Each point cloud was also taken with different 4K cameras, but this is comparatively minor issue in comparison to the other discrepancies.

5. CONCLUSION

The work completed provides fellow researchers with three specific flight plans that may be of interest in their research. They can load the outputs of the provided code into an application that is free to download or the

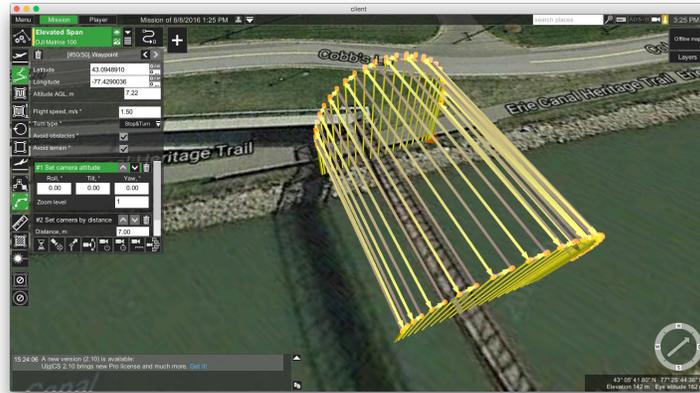


Figure 12. Screen capture of UgCS showing a bridge inspection flight profile. Note panels would be situated in top right corner of the image.



Figure 13. Point cloud reconstruction of a statue with many self-occlusions.

fully licensed version. More importantly, researchers can use the provided code to generate their own customized flight plans. If a user can describe a flight pattern in an x-y-z coordinate space then they can generate their own flight profile in ‘.xml’ format that can be read into UgCS.

6. FUTURE WORK

The efforts presented document the work required to create custom flight plans for use on readily available software. With this complete, efforts will now shift towards processing imagery acquired using these flight profiles.

REFERENCES

- [1] Balog, C. R., Terwilliger, B. A., Vincenzi, D. A., and Ison, D. C., “Examining human factors challenges of sustainable small unmanned aircraft system (suas) operations,” in [*Advances in Human Factors in Robots and Unmanned Systems*], 61–73, Springer (2017).

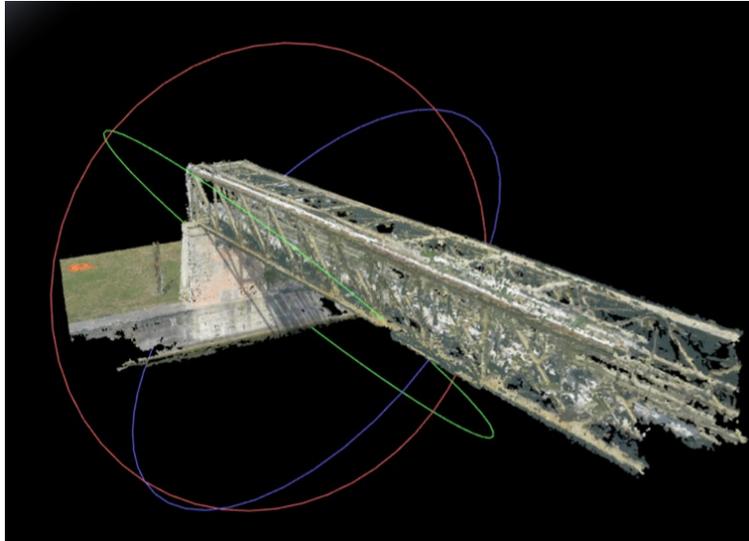


Figure 14. Point cloud reconstruction of a pipebridge using the bridge profile algorithm.

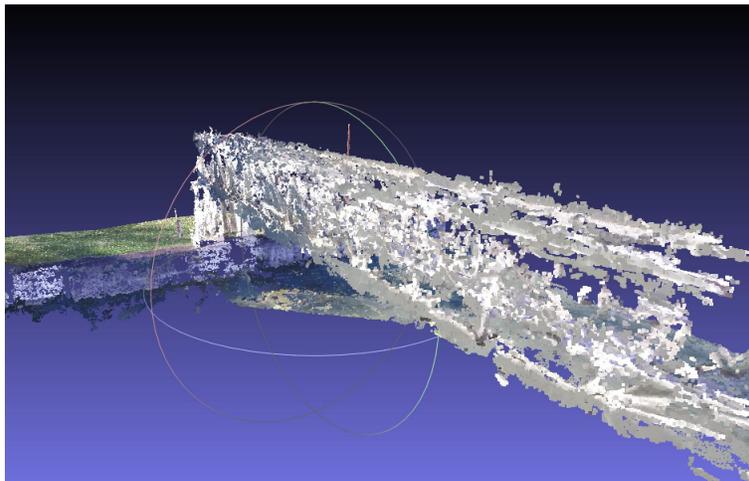


Figure 15. Point cloud reconstruction of a pipebridge using images collected at nadir.

- [2] Burkart, A., Aasen, H., Alonso, L., Menz, G., Bareth, G., and Rascher, U., “Angular dependency of hyperspectral measurements over wheat characterized by a novel uav based goniometer,” *Remote sensing* **7**(1), 725–746 (2015).
- [3] Galleguillos, C., Zorrilla, A., Jimenez, A., Diaz, L., Montiano, Á., Barroso, M., Viguria, A., and Lasagni, F., “Thermographic non-destructive inspection of wind turbine blades using unmanned aerial systems,” *Plastics, Rubber and Composites* (2015).
- [4] Sullins, J., Warren, D., and Wiers, W., “Pipeline inspection pig,” (Jan. 22 1974).
- [5] Schott, J. R., [*Remote sensing: the image chain approach*], Oxford University Press on Demand (2007).
- [6] Smith, G. M. and Milton, E. J., “The use of the empirical line method to calibrate remotely sensed data to reflectance,” *International Journal of remote sensing* **20**(13), 2653–2662 (1999).
- [7] Bondi, E., Salvaggio, C., Montanaro, M., and Gerace, A. D., “Calibration of uas imagery inside and outside of shadows for improved vegetation index computation,” in [*SPIE Commercial+ Scientific Sensing and Imaging*], 98660J–98660J (2016).