

Captura y formación de imágenes hiperespectrales mediante UAV's

Capture and formation of hyperspectral images using UAV's

Christopher Brenes-Zeledón¹

Brenes-Zeledón, C. Captura y formación de imágenes hiperespectrales mediante UAV's. *Tecnología en Marcha*. Edición especial. Movilidad Estudiantil 6, 2019. Pág 24-34

 <https://doi.org/10.18845/tm.v32i8.4561>

¹ Estudiante ingeniería Mecatrónica. Instituto Tecnológico de Costa Rica. Área Académica de ingeniería Mecatrónica. Cartago Costa Rica. Correo electrónico: cj.bz19@gmail.com



Palabras clave

Algoritmo; control; drone; hiperspectral; mapas; misiones; mosaico; panorama; rutas; UAV.

Resumen

En este artículo se explica el desarrollo de las primeras etapas de un sistema capaz de capturar imágenes hiperspectrales mediante un vehículo aéreo no tripulado (UAV), conocidos comúnmente como drones. Esto con el fin de formar una imagen o mapa hiperspectral completo a partir de las capturas realizadas. Además, se explica la teoría necesaria para comprender los conceptos que engloban la solución y los resultados obtenidos, así como las conclusiones de los avances. Lo explicado en el artículo corresponde una parte del trabajo realizado como proyecto de graduación para optar por el grado académico de Licenciatura en Ingeniería Mecatrónica, del Tecnológico de Costa Rica.

Abstract

In this article it is explained the development of the first stages of a system capable of capturing hyperspectral images using an unmanned aerial vehicle (UAV), commonly known as drones. The objective is to create a complete hyperspectral map, using those captured images. In addition, it is explained the necessary theory to comprehend concepts related with the solution, results and conclusions of this work. What is explained in this article, is a part of a graduation project for a Bachelor's degree in Mechatronics Engineering, at Tecnológico de Costa Rica.

Keywords

Algorithm; control; drone; hyperspectral; maps; missions; mosaicing; panoramic; routes; UAV.

Introducción

El proyecto nació de una necesidad de aumentar la competitividad en la industria agrícola mediante la agricultura de precisión y la integración de distintas tecnologías para la monitorización de cultivos. El Ministerio de Economía y Competitividad del Gobierno Español ha planteado el proyecto “Sistema Integral para la Monitorización y Multiresolución en Agricultura de Precisión” (SIMMAP), el cual involucra varios centros de investigación, entre ellos el Grupo Integrado de Ingeniería (GII), ubicado en el Campus de Ferrol de la Universidade da Coruña, España. Esto con el fin de responder a esta necesidad y mejorar la calidad de los productos y procesos productivos.

El GII es un grupo interdisciplinar de investigación aplicada en ingeniería, orientado a la generación de conocimiento científico y transferencia de tecnología a la industria con la generación de nuevos productos. La investigación científica que realizan se centra en las áreas de: inteligencia computacional, robótica, dinámica de fluidos, sistemas de medida, sensores y organización industrial. El grupo se ha visto interesado en integrar a estudiantes de la carrera de Ingeniería Mecatrónica del Tecnológico de Costa Rica, quienes han trabajado exitosamente con el grupo en varias ocasiones.

Es importante mencionar que los UAV (Unmanned Aerial Vehicle) también conocidos como drones, juegan un papel muy importante en la monitorización a baja altura de los cultivos. Anteriormente se hicieron avances exitosos en el proyecto, por lo cual se buscó continuar con la solución de nuevos problemas. En este artículo se explican dos etapas del trabajo realizado

con el grupo de investigación, las cuales son el sistema de control para un dron de ala fija, con el fin de que ejecute rutas o misiones de forma autónoma según las necesidades de la cámara hiperespectral, y el desarrollo de los mapas hiperespectrales completos a partir de capturas parciales realizadas.

Fundamentos

Un UAV (Unmanned Aerial Vehicle) o vehículo aéreo no tripulado, como lo indica su nombre, es una aeronave capaz de volar sin necesidad de un operador o tripulación a bordo, como se indica en [1]. También se les conoce comúnmente como drones. Las nuevas tecnologías han permitido que puedan ser utilizados eficientemente en muchos campos de aplicación. Los UAV's llevan cargas de trabajo, las cuales se refieren al conjunto de sensores y dispositivos agregados para cumplir una misión específica [1]. Las características principales del UAV son:

- No necesita un piloto a bordo.
- Es recuperable.
- Lleva una carga de trabajo.
- Posee su propio sistema de vuelo autónomo.

Para este proyecto se utilizó un dron de ala fija. Sus alas se encuentran unidas al cuerpo de la aeronave. La sustentación se genera gracias al diseño del perfil aerodinámico del ala, el cual genera diferencias de presión entre la parte superior (extradós) e inferior (intradós) [2], como lo hace un avión. Esto les permite tener una sustentación muy eficiente, mantenerse volando durante lapsos elevados y ahorrar energía. La desventaja es que necesitan de un sistema o fuerza externa que les permita realizar el despegue y el aterrizaje, y su maniobrabilidad durante el vuelo es poca, por lo que sus movimientos se ven limitados y se les hace difícil volar en espacios estrechos.

Con respecto a las imágenes hiperespectrales, se definen como una colección de imágenes que tienen una cierta cantidad de bandas hiperespectrales a lo largo del espectro electromagnético [3]. En otras palabras, se obtiene un cubo de datos donde se puede apreciar la misma ubicación espacial, a diferentes longitudes de onda. Esto permite realizar estudios más amplios de la imagen porque permite analizar información más allá de la luz visible (comprendida de 400 a 750 nm aproximadamente). Es importante tener en cuenta que la radiación ultravioleta comprende las longitudes de onda más cortas (menos de 400 nm) y la radiación infrarroja comprende las longitudes de onda más largas (más de 750nm). Dichas longitudes proporcionan gran cantidad de información, por lo que las imágenes hiperespectrales permiten hacer un análisis más completo.

El cubo hiperespectral puede entenderse como un arreglo (x,y,z) , donde 'x' y 'y' pueden representar las dimensiones espaciales de la imagen y 'z' la longitud de onda de la imagen [4]. De esta forma se obtiene una gran cantidad de imágenes del mismo espacio, pero a diferentes longitudes de onda, recopiladas formando un cubo, como se muestra en la figura 1. Por otra parte, un mapa hiperespectral es la unión de varios cubos hiperespectrales para formar una imagen completa de una zona que no puede ser capturada completamente por la cámara, donde cada cubo es una sección de ésta. Se hace la integración de los cubos tomando en cuenta la posición real a la cual fue tomada cada imagen.

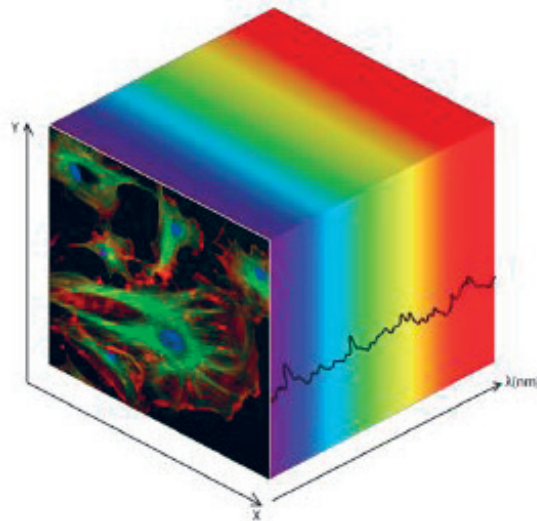


Figura 1. Ejemplo de cubo hiperespectral [4].

Métodos de Solución

Para este caso, se tomó la decisión de utilizar el controlador de vuelo Pixhawk 2. Como se explica en [5], el Pixhawk 2 es de código abierto y se suele utilizar como autopiloto. Además, puede configurarse fácilmente a través de software especializado en vuelo de UAV's. Otro aspecto muy importante para el desarrollo de este proyecto es que el Pixhawk 2 es capaz de enviar información de su orientación a través de las salidas para PWM. De esta forma se puede enviar información para controlar la orientación de la cámara a través de un estabilizador inercial. En resumen, para este proyecto el Pixhawk 2 cumplió cuatro funciones:

- Controlar la orientación del UAV mediante las salidas conectadas a los servomotores, que manipulan los alerones del drone, a partir de la información brindada por su IMU interna.
- Comunicarse de forma bidireccional con la estación en tierra, para enviar datos del estado del UAV y visualizarlos en el PC y a la vez, recibir órdenes desde la emisora (control remoto) y el PC.
- Enviar información de la orientación por las salidas PWM, para que el controlador del estabilizador inercial estabilice la cámara.
- Controlar la posición del UAV con la información del sensor GPS, y poder seguir una misión o ruta programada previamente.

El software utilizado para el trabajo con el vuelo del drone fue el Mission Planner. Dos de sus funciones principales son: programar en el controlador de vuelo las misiones o rutas que serán seguidas de forma autónoma y calibrar o configurar los sensores y la radio emisora. El Mission Planner es una aplicación que funciona como una estación en tierra para el control del UAV [6]. Algunas de las facilidades que presenta el software son las siguientes:

- Cargar el firmware en el controlador del vehículo.
- Configurar el controlador de vuelo para la lectura de las señales enviadas por los sensores y la 3. Radio emisora (control remoto).
- Almacenar y cargar archivos con las misiones.
- Interfaz con comandos para ejecutar en tiempo real durante las misiones.
- Colocar puntos de ruta a través de Google Maps, Bing, Open Street Maps y Custom WMS.
- Analizar y almacenar registros de datos.
- Interfaz gráfica para visualizar en tiempo real la información del UAV.

Como se mencionó anteriormente, el Mission Planner posee una función para calcular los puntos de ruta que debe realizar el drone, vez que se le indica el modelo de cámara comercial que se va a utilizar. Como se utilizó una cámara hiperespectral, se generó una ruta o secuencia de puntos que considere el porcentaje de solape deseado entre imágenes, y el ángulo de apertura de la cámara. Si se conoce la altura y el ángulo de apertura de la cámara, por trigonometría puede saberse la distancia sobre el suelo que cubre la visión de la cámara.

El algoritmo se realizó en Matlab y utiliza fórmulas para el cálculo de latitud y longitud de los puntos de ruta, tomadas de [7].

Para calcular el ángulo de la línea que une dos puntos en el mapa, se utilizó la siguiente ecuación:

$$\theta = \text{atan2}(\sin\Delta\lambda \cdot \cos\phi_2, \cos\phi_1 \cdot \sin\phi_2 - \sin\phi_1 \cdot \cos\phi_2 \cdot \cos\Delta\lambda)$$

Donde ϕ_1 y λ_1 son la latitud y longitud del punto inicial respectivamente y ϕ_2 , λ_2 del segundo punto. $\Delta\lambda$ es la diferencia de longitudes. Este ángulo se mide desde el norte, en sentido horario.

También se utilizó otra fórmula tomada de [7], para el cálculo del punto de destino, a partir de un punto inicial, el ángulo entre los puntos y la distancia entre ellos. La distancia entre los puntos depende del solape entre imágenes que se desee. En las siguientes ecuaciones se calcula la latitud y longitud del punto de destino respectivamente:

$$\phi_2 = \text{asen}(\sin\phi_1 \cdot \cos\delta + \cos\phi_1 \cdot \sin\delta \cdot \cos\theta)$$

$$\lambda_2 = \lambda_1 + \text{atan2}(\sin\theta \cdot \sin\delta \cdot \cos\phi_1, \cos\delta - \sin\phi_1 \cdot \sin\phi_2)$$

Donde ϕ_1 y λ_1 y son la latitud y longitud del primer punto respectivamente y ϕ_2 , λ_2 del segundo punto. θ es el ángulo de la línea que une los puntos (medido en sentido horario desde el norte), δ es la relación de distancias d/R ; siendo d la distancia en metros que se quiere viajar y R el radio de la Tierra, aproximado a $R = 6372795,477598\text{m}$.

Además de calcular la ruta, se realizó otro algoritmo en Matlab para conocer la altura mínima a la cual debe volar el UAV para lograr un solape deseado. La velocidad mínima para volar sin que el drone utilizado caiga, ronda los 10 m/s, según el peso que porta para este proyecto, y la altura calculada no debe sobrepasar los 120m, pues por asuntos legales en España (donde se realizó el proyecto) no se puede volar un UAV a más de esta altura. Si el valor calculado de altura dice que se requiere volar a más de 120m, entonces no será posible realizar el vuelo para cumplir las condiciones de solape deseadas y tiempo entre captura de imágenes. En la siguiente ecuación se muestra cómo se calcula la altura mínima de vuelo.

$$h = \frac{v \cdot (t_i + t_{exp} + t_w)}{(1 - overlap) \cdot 2 \cdot \tan(\theta/2)}$$

Donde:

h = altura [m].

v = velocidad del UAV [m/s].

t_{exp} = tiempo de exposición de la cámara [s].

t_w = tiempo de escritura [s].

t_i = tiempo deseado entre cada captura de imagen [s].

overlap = porcentaje de solape (de 0 a 1).

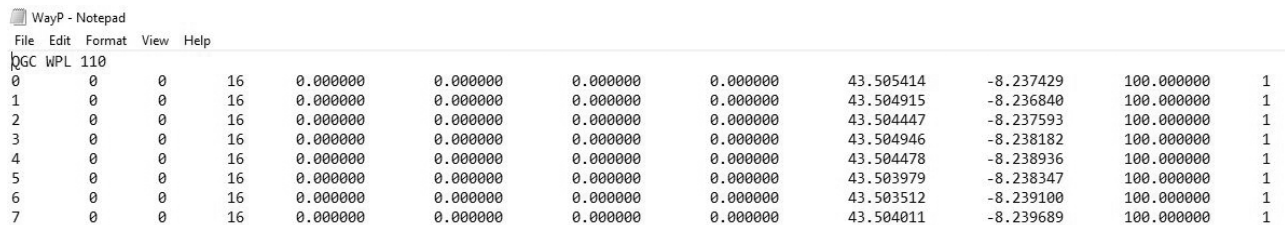
θ = ángulo de apertura de la cámara [rad].

El algoritmo genera un documento de texto con los puntos calculados, según el formato en el cual el Mission Planner lee las misiones.

Para realizar los mapas hiperespectrales se modificó el algoritmo de Matlab para unir imágenes, llamado "Panorama". Este código permite traslapar varias imágenes insertadas manualmente y formar un panorama más grande, que es lo que se busca con este proyecto. Se modificó el código y se crearon nuevas funciones para aplicarlo a cubos hiperespectrales en todas sus bandas, como se explica en [8].

Resultados

El uso del Mission Planner permitió llevar a cabo la programación de misiones y calibración de los sensores para trabajar con el UAV Skywalker X8. Para el caso del algoritmo de generación de rutas para la cámara hiperespectral, se obtuvieron resultados exitosos. Mediante el algoritmo se calcularon correctamente los puntos de la ruta, y se generó el archivo de texto mostrado en la figura 2. En la figura 3 se muestra la ruta en el Mission Planner, al cargar el archivo de texto generado.



| Line | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|------|---|---|---|----|----------|----------|----------|----------|-----------|-----------|------------|----|
| 0 | 0 | 0 | 0 | 16 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 43.505414 | -8.237429 | 100.000000 | 1 |
| 1 | 0 | 0 | 0 | 16 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 43.504915 | -8.236840 | 100.000000 | 1 |
| 2 | 0 | 0 | 0 | 16 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 43.504447 | -8.237593 | 100.000000 | 1 |
| 3 | 0 | 0 | 0 | 16 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 43.504946 | -8.238182 | 100.000000 | 1 |
| 4 | 0 | 0 | 0 | 16 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 43.504478 | -8.238936 | 100.000000 | 1 |
| 5 | 0 | 0 | 0 | 16 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 43.503979 | -8.238347 | 100.000000 | 1 |
| 6 | 0 | 0 | 0 | 16 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 43.503512 | -8.239100 | 100.000000 | 1 |
| 7 | 0 | 0 | 0 | 16 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 43.504011 | -8.239689 | 100.000000 | 1 |

Figura 2. Archivo de texto generado con el algoritmo desarrollado.

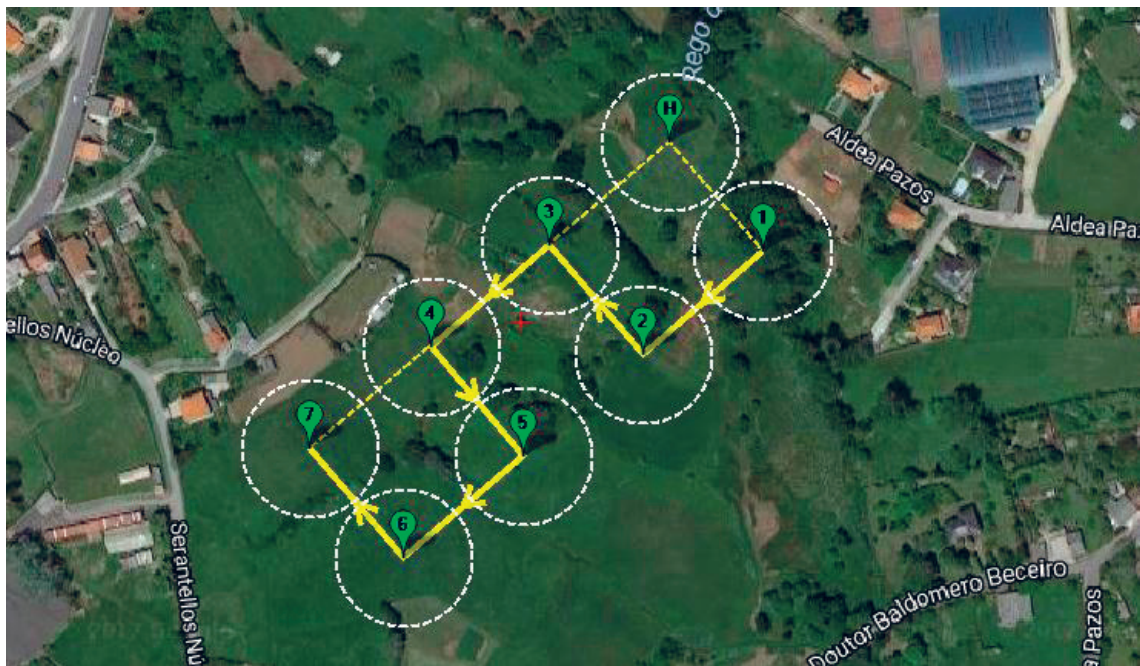


Figura 3. Ruta generada con el Mission Planner a partir del archivo de texto.

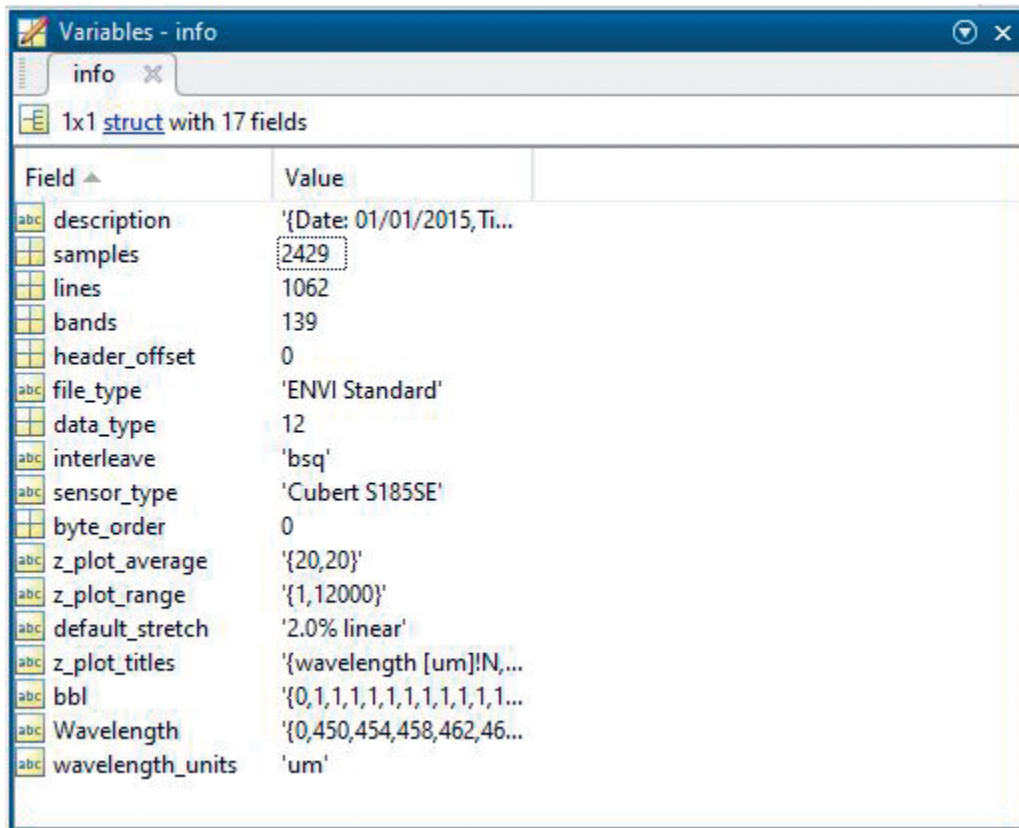
Este archivo de texto generado debe cargarse en el Mission Planner una vez colocado el punto de despegue y los puntos necesarios para que el UAV alcance la altura y posición adecuadas para comenzar con la ruta, según el criterio de quien diseña la misión. Luego se deben colocar manualmente los puntos de aterrizaje, tomando las consideraciones de disminución gradual de la altura. En otras palabras, se colocan manualmente los puntos para despegar y ubicarse en la zona que se quiere mapear, luego se carga la ruta generada con el algoritmo mediante el comando “Load and Append” y por último se colocan manualmente los puntos para hacer el aterrizaje.

El estudio y la modificación del algoritmo de Matlab para crear panoramas, permitió realizar modificaciones y crear funciones nuevas para generar los mapas hiperespectrales. El algoritmo primero lee todos los cubos hiperespectrales de una carpeta, y forma los mapas eficientemente para cada banda. A su vez los va almacenando en una carpeta de forma ordenada. El nombre de cada mapa corresponde al número de banda hiperespectral del cual se formó. Esto permite analizar la información deseada directamente al ver la imagen de cada mapa. Por último se generaron los archivos de datos (.cue) y de cabecera (.hdr) que conforman el mapa completo como tal, con la información de todas las bandas.

Para mostrar la funcionalidad del algoritmo, se capturaron varios cubos hiperespectrales, cada uno de una sección de un edificio. Se formó el mapa que permite ver un panorama del edificio completo. En la figura 4 se observa que en la variable ‘D’ se guardan los datos del mapa, con las dimensiones finales de la imagen y las 139 bandas hiperespectrales. En la variable ‘info’ se guarda la información del mapa proporcionada por el archivo de extensión ‘.hdr’ generado mediante las funciones creadas. Esta información puede verse en la figura 5.

| Workspace | |
|-----------|----------------------|
| Name ▲ | Value |
| D | 1062x2429x139 double |
| im | 1062x2429 double |
| info | 1x1 struct |

Figura 4. Ejemplo de variables creadas por el algoritmo en Matlab.



| Variables - info | |
|---------------------------|--------------------------|
| info | |
| 1x1 struct with 17 fields | |
| Field ▲ | Value |
| description | {Date: 01/01/2015, Ti... |
| samples | 2429 |
| lines | 1062 |
| bands | 139 |
| header_offset | 0 |
| file_type | 'ENVI Standard' |
| data_type | 12 |
| interleave | 'bsq' |
| sensor_type | 'Cubert S185SE' |
| byte_order | 0 |
| z_plot_average | {20,20}' |
| z_plot_range | {1,12000}' |
| default_stretch | '2.0% linear' |
| z_plot_titles | {wavelength [um]!N,... |
| bbl | {0,1,1,1,1,1,1,1,1,1,... |
| Wavelength | {0,450,454,458,462,46... |
| wavelength_units | 'um' |

Figura 5. Información del mapa hiperespectral.

En la figura 6 se muestran algunas bandas del mapa hiperespectral generado. Se capturaron imágenes de un edificio desde el laboratorio, porque las condiciones climáticas no permitían exponer la cámara al exterior y capturar imágenes del suelo. Para el caso de la cámara utilizada, el ancho de banda es de 4nm. Es decir, que cada banda hiperespectral contiene información de un rango de longitudes dado por 4nm.

En la figura 7 se muestra una imagen del mismo edificio capturado con la cámara hiperespectral, pero esta vez capturado con una cámara de teléfono móvil. Esto con el fin de hacer la comparación entre la imagen reconstruida con el algoritmo y una captura del edificio real. Se puede observar que las imágenes del mapa corresponden al edificio. Se requieren hacer pruebas de capturas de suelos, con el fin de analizar el comportamiento del algoritmo y estudiar la continuidad de la imagen final en un escenario similar al que se usará la cámara durante el vuelo del UAV.



(a) Banda en la posición 10 del mapa (482-486nm). (b) Banda en la posición 30 del mapa (562-566nm).



(c) Banda en la posición 50 del mapa (642-646nm). (d) Banda en la posición 70 del mapa (722-726nm).



(e) Banda en la posición 80 del mapa (762-766nm). (f) Banda en la posición 100 del mapa (842-846nm).

Figura 6. Bandas correspondientes al mapa hiperespectral formado.



Figura 7. Edificio real capturado con teléfono móvil

Conclusiones

De este proyecto se concluye que se logró diseñar un sistema de control para que un UAV realice misiones o rutas de forma autónoma. Se realizó un código en Matlab para generar la ruta que debe seguir el UAV y hacer la captura de imágenes, cumpliendo con las especificaciones de la misión y de la cámara hiperespectral. Se utilizaron el Mission Planner y el Pixhawk 2 como elementos para realizar el control del UAV.

El algoritmo creado en Matlab para realizar los mapas hiperespectrales fue exitoso. Se obtuvieron mapas hiperespectrales para todas las bandas de un conjunto de cubos capturados. Debido a las condiciones climáticas no se realizaron capturas en el exterior, pero se verificó el funcionamiento del algoritmo a partir de capturas realizadas a un edificio.

Recomendaciones

El Mission Planner permite configurar al Pixhawk para guardar datos en tiempo real durante el vuelo (Data Logs). Se recomienda guardar datos de orientación, posición y tiempo (IMU/GPS). De esta forma se podría hacer un análisis detallado de lo que sucede durante cada vuelo. Con esto también sería posible realizar estudio de fallas y otras tareas.

También es recomendable conocer la posición geográfica (latitud, longitud y altitud) en la cual se capturó cada cubo durante el vuelo. Para esto también se sugiere usar los "Data Logs" guardados en el Pixhawk 2, para obtener la información de posición (GPS). Se podría realizar una comparación entre el valor de tiempo en el cual se capturó cada cubo (calculado a partir de la frecuencia de captura) y buscar el valor de posición correspondiente a ese tiempo aproximado en los datos del Pixhawk 2.

Si se conoce la posición en la cual se capturó cada cubo, se podría implementar un algoritmo para no tomar en cuenta los cubos hiperespectrales que fueron capturados durante el despegue y aterrizaje. Para esto es muy importante definir los criterios bajo los cuales se desecharían estos cubos. Esto porque se puede volver tedioso hacer la selección manualmente, aunque no deja de ser otra opción.

Por último, realizar pruebas de mapas hiperespectrales en suelos es importante para estudiar el comportamiento del algoritmo en situaciones similares a su función final, la cual es la monitorización de cultivos.



Referencias

- [1] C.Calvo, F.Herranz, and P.Calvo. De los UAV a los RPAS. Perfiles IDS, 2014.
- [2] E. Santana. Tipos de drones - conoce todos los tipos de drones que existen - clasificación de drones - categorías de drones [online]. 2017 [visitado el 8 de marzo de 2018]. URL: <http://www.xdrones.es/tipos-de-drones-clasificacion-de-drones-categorias-de-drones/>.
- [3] B.Priego. Método integral para la adquisición y procesamiento eficiente de imágenes y secuencias hiperespectrales, 2011.
- [4] Photonetc. Our technology - hyperspectral imaging [online]. s.f. [visitado el 8 de marzo de 2018]. URL <http://www.photonetc.com/hyperspectral-imaging>.
- [5] ProfiCNC. Pixhawk2.1 [online]. s.f. [visitado el 8 de marzo de 2018]. URL <http://www.proficnc.com/content/13-pixhawk2>.
- [6] ArduPilot. Mission planner home [online]. 2016 [visitado el 8 de marzo de 2018]. URL <http://ardupilot.org/planner/>.
- [7] C.Veness. Calculate distance, bearing and more between latitude/longitude points [online]. 2017 [visitado el 8 de marzo de 2018]. URL <https://www.movable-type.co.uk/scripts/latlong.html>.
- [8] C.Brenes. Sistema de captura y formación de imágenes hiperespectrales mediante un UAV, 2018.