

Tecnológico de Costa Rica

Área Académica de Ingeniería Mecatrónica



Desarrollo de un Sistema de Captación de Imágenes Hiperespectrales Mediante UAVs para  
Monitorización en Agricultura

Informe de Proyecto de Graduación para Optar por el Título de Ingeniero en Mecatrónica con Grado  
Académico de Licenciatura

Autor:

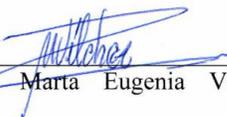
Rodolfo José Piedra Camacho

Informe de 21 de Enero del 2017

**INSTITUTO TECNOLÓGICO DE COSTA RICA**  
**CARRERA DE INGENIERÍA MECATRÓNICA**  
**PROYECTO DE GRADUACIÓN**  
**ACTA DE APROBACIÓN**

Proyecto de Graduación defendido ante el presente Tribunal Evaluador como requisito para optar por el título de Ingeniero en Mecatrónica con el grado académico de Licenciatura, del Instituto Tecnológico de Costa Rica.

Miembros del Tribunal

  
\_\_\_\_\_  
M. Sc. Marta Eugenia Vilchez  
Monge  
Profesor lector

  
\_\_\_\_\_  
M. Sc. Carlos Adrián Salazar García  
Profesor lector

  
\_\_\_\_\_  
Dr. Ing. Juan Luis Crespo Mariño  
Profesor asesor

Los miembros de este Tribunal dan fe de que el presente trabajo de graduación ha sido aprobado y cumple con las normas establecidas por la Carrera de Ingeniería Mecatrónica

Cartago, 20 de Enero 2017

## Declaratoria de Autenticidad

Declaro que el presente Proyecto de Graduación ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos propios.

En los casos en que he utilizado bibliografía, he procedido a indicar las fuentes mediante las respectivas citas bibliográficas.

En consecuencia, asumo la responsabilidad total por el trabajo de graduación realizado y por el contenido del correspondiente informe final.

Cartago, 20 de enero de 2017



Rodolfo José Piedra Camacho

Cédula: 3 - 0484 - 0713

## Resumen

Como parte del proyecto SIMMAP del Ministerio de Economía y Competitividad del Gobierno Español, el cual busca aumentar la competitividad de la industria agrícola con la integración de distintas tecnologías en dicho ámbito, el presente proyecto desarrolla y discute la solución al problema de integrar dos tecnologías necesarias para la monitorización de cultivos a baja altura: Un sistema octacóptero y cámaras con sensores hiperespectrales.

En este trabajo se detalla el proceso de selección del sistema de estabilización inercial para el acople, el diseño de piezas para solventar dificultades en la unión hiperespectrómetro-estabilizador, el proceso de control de cámaras y estabilizadores, mediante programas diseñados en C++, para las labores de captura de cubos hiperespectrales, el desarrollo de script en Matlab para la comparación de información hiperespectral, y finalmente el desarrollo de aplicaciones de prueba para realizar las primeras capturas de prueba en aire.

Entre los principales resultados se tiene el diseño de tres bibliotecas para el control de hiperespectrómetros y el estabilizador y la captura y almacenado de líneas hiperespectrales a un ritmo de 7.5 líneas por segundo.

**Palabras Clave:** Hiperespectrómetro, octacóptero, estabilizador inercial, SIMMAP.

## Summary

As part of the Spanish Government Ministry of Economy and Competitive project SIMMAP, which seeks to increase the quality of the agricultural industry with the integration of a variety of technologies, this project elaborates and develops a solution to the union of two technologies necessary to supervise crops at low heights: An octocopter and cameras equipped with hyperspectral sensors.

This work details the selection process for a gimbal stabilizer, the design of the necessary pieces to solve any difficulty in the union hyperspectrometer-gimbal, the control process of both hyperspectrometer and stabilizer to capture hyperspectral cubes, via C++ programs, the development of Matlab scripts which serve to compare hyperspectral data and finally the design of applications required for the first aerial tests.

Three C++ libraries where designed to control the stablizer and the hyperespectral cameras, hyperespectral lines where captured and saved at a rate of 7.5 lines per second.

**Palabras Clave:** Hyperspectrometer, octocopter, gimbal stabilizer, SIMMAP.

## Dedicatoria

### ***A Dios***

Por darme la vida, así como las oportunidades de desarrollo y mejora personal a lo largo de mi vida, por su amor y misericordia. Que este proyecto sea un genuino esfuerzo de hacer valer, de forma adecuada, los dones que me ha dado.

### ***A mi Madre Lucrecia***

Por su amor y consejo, por haberme inculcado desde joven los buenos hábitos de la responsabilidad y la prudencia en el estudio y principalmente por haber sido parte esencial de la formación de la persona que soy hoy día.

### ***A mi Padre Rodolfo***

Por su amor y apoyo, por haberme impulsado a buscar opciones fuera del país para el desarrollo de este proyecto, por ser ejemplo de la entrega profesional en honor a una familia. En agradecimiento de las horas de desvelo entregadas por dar oportunidades a su familia.

### ***A mi Hermano Gabriel***

Por su amistad y compañía, por ser fuente de alivio en los momentos de mayor estrés y en honor a la búsqueda de un futuro gratificante para ambos.

## Agradecimiento

Agradezco a Dios principalmente por darme los dones y las oportunidades necesarias para poder desarrollar adecuadamente los objetivos de este proyecto, especialmente en momentos de frustración.

A mi familia por el esfuerzo tanto económico como emocional de permitirme realizar el proyecto de graduación fuera del país, vuelvo a Costa Rica en miras de devolver a la familia el amor y esfuerzo invertidos.

Agradezco a los varios amigos y amigas del TEC, no solo por cinco años de compañerismo y esfuerzo en la carrera, sino también por las amistades genuinas, el ocio y la sensación de confianza y cercanía; me han ayudado a gozar la época universitaria de una forma genuina y correcta.

A todos los miembros del GII que de una u otra forma me ayudaron no solo en consolidar conocimientos necesarios para el desarrollo del proyecto, sino también a explorar y conocer más sobre el ámbito de investigación y academia. Por la recepción amigable y la ayuda en épocas de enfermedad o de acople a la zona de Ferrol y finalmente por hacer de mi estadía una experiencia alegre, amena y completa.

Finalmente al Profesor Juan Luis Crespo por la resolución constate de dudas, el apoyo a lo largo de la ejecución del proyecto y principalmente por presentarme la oportunidad de desarrollar mi proyecto de graduación en conjunto con el GII.

Muchas Gracias

## Contenido

Índice de Figuras .....	11
Índice de Tablas .....	15
1. Introducción .....	16
1.1. Planteamiento del Problema .....	17
1.2. Objetivos del Proyecto .....	18
1.2.1. Objetivo General .....	18
1.2.2. Objetivos Específicos.....	18
1.3. Alcances y Limitaciones .....	19
2. Marco Teórico .....	20
2.1. Imágenes y Cámara Hiperespectrales.....	20
2.1.1. Características de los Sensores Hiperespectrales .....	21
2.1.2. Tipo de Hiperespectrómetros .....	22
2.1.3. Forma de Operación y Funcionamiento de Cámaras Hiperespectrales .....	22
2.2. UAV .....	24
2.2.1. Parámetros Limitantes de Operación .....	25
2.2.2. Sistema de Orientación .....	25
2.3. Sistema de Estabilización Inercial: Gimbal.....	26
2.3.1. Lógica de Control .....	28
2.3.2. Instrumentación y Mediciones Necesarias .....	31
2.3.3. Motores.....	32
2.3.4. Cantidad de Ejes a Controlar.....	32
2.3.5. Balance de Gimbal .....	33
2.4. Antecedentes y Estado del Arte.....	35
3. Desarrollo de la Solución .....	39
3.1. Comprensión de Cámara Hiperespectral.....	39
3.2. Selección de Gimbal.....	40
3.2.1. Estudio sobre Gimbal .....	40
3.2.2. Requisitos de Gimbal y Perfil de Búsqueda Inicial .....	41
3.2.3. Búsqueda de Modelos Comerciales.....	42
3.2.4. Discusión y Selección de Modelo Gimbal .....	52
3.3. Configuración de Cámara Hiperespectral UI3360-CP para Operación en Trípode Nexstar .....	54

3.3.1.	Requisitos de Interfaz .....	55
3.3.2.	Comunicación y Operación de Cámara UI 3360-CP .....	56
3.3.3.	Operación con Tripode Nexstar .....	57
3.3.4.	Lógica de Barrido.....	59
3.3.5.	Interfaz Implementada .....	60
3.3.6.	Resultados de Interfaz .....	63
3.4.	Control de Cámara Hiperespectral UHD-185 Tipo Snapshot.....	64
3.4.1.	Características de la Cámara y Requisitos de su Control .....	65
3.4.2.	Método de Comunicación y Control Mediante Comandos .....	66
3.4.3.	Interfaz de Pruebas para Control de Hiperespectrómetro UHD-185 .....	67
3.5.	Programación en Matlab para Manejo de Información Hiperespectral.....	68
3.5.1.	Generación de Cubo de Datos del Hiperespectrómetro Snapshot.....	68
3.5.2.	Comparación Gráfica de Información Hiperespectral entre Cámara Push-broom y Snapshot .....	70
3.6.	Microcontrolador para Control y Operación de Trípode y Cámara Hiperespectral .....	76
3.7.	Mejora de Tiempo de Operación en Cámara Pushbroom .....	77
3.7.1.	Método de Análisis de Optimización .....	77
3.7.2.	Tiempo de Captura.....	77
3.7.3.	Tiempo de Guardado .....	78
3.8.	Control de Gimbal CAME Mini 3-Air .....	89
3.8.1.	Protocolo de Comunicación Serial Simple-BGC .....	90
3.8.2.	Interfaz de Pruebas en C++ .....	92
3.8.3.	Control SBGC en Gimbal CAME Mini 3-air .....	95
3.9.	Acople Físico de Gimbal a UAV y Cámara Hiperespectral.....	97
3.9.1.	Unión Gimbal-UAV .....	97
3.9.2.	Union Gimbal-Cámara Hiperespectral .....	100
3.10.	Bibliotecas de Control Puhsbroom, Snapshot y Gimbal Came Mini 3-air.....	106
3.11.	Programa Final de Pruebas .....	107
4.	Verificación de Objetivos .....	112
4.1.	Selección de Estabilizador Comercial.....	112
4.2.	Sistema de Fijación entre Cámara Hiperespectral Gimbal y UAV.....	112
4.3.	Control de Estabilizador Comercial y Cámara Hiperespectral .....	113
4.4.	Verificación del Proyecto .....	113

5.	Conclusiones .....	115
6.	Recomendaciones y Trabajos Futuros .....	116
7.	Referencias Bibliográficas .....	117
8.	Anexos.....	122
8.1.	-Fuentes Consultadas para Tabla de Comparación de Gimbals.....	122
9.	Apéndices.....	125
9.1.	-Preguntas Sobre Requisitos de Gimbals .....	125
9.2.	Tabla Completa de Valores de Proporción de Imagen y Valor F para Corrección de Tamaño de Píxel	126
9.3.	Biblioteca “Control_Pushbroom_UI3360_CP_GII.h” .....	128
9.4.	Biblioteca “Control_Snapshot_GII.h” .....	144
9.5.	Biblioteca “Control_SBGC_GII.h” .....	147

## Índice de Figuras

Figura 2.1-1 Concepto de Imagen RGB. Extraído de (Scyllarus, 2016).....	20
Figura 2.1-2 Concepto de Imagen Hiperespectral. Extraído de (Photon, s.f.) .....	21
Figura 2.1-3 Sistema Óptico de Cámara Hiperespectral. Imagen proveída por el GII .....	23
Figura 2.1-4 Hiperespectrómetro con Cámara Retiga Exi. Extraído de (Priego Torres, 2011) .....	24
Figura 2.1-5 Hiperespectrómetro con Cámara UI3360 CP-NIR. Imagen proveída por el GII.....	24
Figura 2.2-1 Angulos de Euler sobre un UAS. Tomado de (Fiorenzani, 2016) .....	25
Figura 2.3-1 Imagen de Gimbal/Giroscopio Conceptual. Extraído de (Gyroscopes.org, 2016) .....	26
Figura 2.3-2 Gimball Comercial de Cámara Pequeña. Extraído de (Ebay, 2016) .....	27
Figura 2.3-3 Gimbal Comercial de Cámara Grande. Extraído de (ProKit, 2015) .....	27
Figura 2.3-4 Concepto de Control de LOS. Extraído de (Brake, 2012) .....	29
Figura 2.3-5 Lógica de Control por Orientación, ilustrado por Andrés Elizondo González estudiante de Ingeniería en Diseño Industrial del ITCR .....	30
Figura 2.3-6 Ejes de Gimbal. Extraído de (The Photo Performance, 2015) .....	33
Figura 2.3-7 Diagrama de lógica de balanceo. Desarrollado en X Mind 7 .....	34
Figura 3.2-1 Dimensiones de Balanceo de Gimbal G3-5D de Align. Extraído de (Algin, 2015).....	44
Figura 3.2-2 Dimensiones de Balanceo de Gimbal CAME-Argo. Extraído de (CAME-Tv, 2016).....	45
Figura 3.3-1 Interfaz para Control de Cámara Retiga. Extraído de (Priego Torres, 2011) .....	55
Figura 3.3-2 EJemplos de comandos de Trípode Nexstar. Extraído de (Celestron, 2008).....	58
Figura 3.3-3 Diagrama de Flujo para la Lógica de Barrido. Desarrollado en X Mind 7 .....	59
Figura 3.3-4 Interfaz de Control Hiperespectrometro UI3360 CP con Tripode Nexstar. Desarrollado en Microsoft Visual Studio Express 2012.....	61
Figura 3.3-5 Interfaz con Error al Inicialziar la Cámara. Desarrollado en Microsoft Visual Studio Express 2012 .....	63
Figura 3.3-6 Resultado de Barrido de Interfaz. Obtenido de Script de Matlab proveído por el GII .....	64
Figura 3.4-1 Interfaz Cube-Pilot. Extraído del Programa Cube Pilot.....	65
Figura 3.4-2 Interfaz de pruebas de control de Hiperespectrómetro UHD-185. Desarrollado en Microsoft Visual Studio Express 2012 .....	67
Figura 3.5-1 Diagrama de Flujo de Script para Generar Cubo de Datos. Desarrollado en X Mind 7 .....	69
Figura 3.5-2 Banda 998nm capturada por Snapshot UHD-185. Extraída de script de Matlab desarrollado. ....	70
Figura 3.5-3 Diagrama de Flujo Completo de Script de Matlab para Comparación de Información Hiperespectral. Desarrollado en X Mind 7.....	71

Figura 3.5-4 Diagrama de Flujo de Sección "Busqueda de Imagen Más Clara". Desarrollado en X Mind	772
Figura 3.5-5 Imagen de Cámara Pushbroom Para Selección de Área. Extraído de script de Matlab desarrollado .....	74
Figura 3.5-6 Imagen de Cámara Snapshot Para Selección de Área. Extraído de script de Matlab desarrollado .....	75
Figura 3.5-7 Gráficas Finales de Comparación Snapshot vs Pushbroom. Extraído de script de Matlab desarrollado .....	75
Figura 3.6-1 Mini-PC Fitlet-iA10. Extraído de (fitPC, 2016).....	76
Figura 3.7-1 Mapa Conceptual de Método de Análisis de Opciones de Optimización. Desarrollado en X Mind 7 .....	77
Figura 3.7-2 Análisis de Disco en Mini-PC Fitlet-iA10. Extraída del Administrador de Tareas de Windows .....	79
Figura 3.7-3 Análisis de CPU en Mini-PC Fitlet-iA10. Extraída del Administrador de Tareas de Windows	80
Figura 3.7-4 Secuencia de Trabajo de Programación por Hilos. Extraído de (Wolfgang, 2016) .....	81
Figura 3.7-5 Lógica de Hilos en Captura y Guardado de Imágenes. Desarrollado en X Mind 7 .....	82
Figura 3.7-6 Comparación de Duración entre Lógicas con y sin Hilos. Desarrollado en Microsoft Excel 2013 .....	82
Figura 3.7-7 Captura de Pantalla de Ejecución de Programa de Pruebas con Hilos. Desarrollado en Microsoft Visual Studio Express 2012.....	83
Figura 3.7-8 Captura de Pantalla de Ejecución de Programa de Pruebas con Hilos en Mini-PC. Desarrollado en Microsoft Visual Studio Express 2012 .....	84
Figura 3.7-9 Captura de Pantalla de Falta de Paralelismo en Pruebas de Programa con Lógica de Hilos en Mini-PC. Extraída con la herramienta Recortes de Windows.....	85
Figura 3.7-10 Gráfica de Comparación entre Formatos PNG 16 bit y PNG 8 bit. Extraído de script de Matlab desarrollado.....	87
Figura 3.7-11 Gráfica de Comparación entre Formatos PNG 16 bit y BMP. Extraído de script de Matlab desarrollado .....	87
Figura 3.7-12 Gráfica de Comparación entre Formatos PNG 16 bit y JPG. Extraído de script de Matlab desarrollado .....	88
Figura 3.7-13 Gráfica de Comparación entre Formatos PNG 8 bit, BMP y JPG. Extraído de script de Matlab desarrollado.....	88
Figura 3.7-14 Gráfica de Comparación entre todos los Formatos. Extraído de script de Matlab desarrollado .....	89
Figura 3.8-1 Ejemplo de Comando de la Serial API del Controlador Simple BGC. Extraído de (BaseCam Electronics, 2016).....	90

Figura 3.8-2 Interfaz de Pruebas para Comunicación Serial con Controlador SimpleBGC. Desarrollado en Microsoft Visual Studio Express 2012.....	92
Figura 3.8-3 Captura de Osciloscopio con Mensaje Serial de Encender Motores. Extraído de Osciloscopio Tektronix MSO 4054 .....	93
Figura 3.8-4 Captura de Osciloscopio con Mensaje Serial de Comando de Control. Extraído de Osciloscopio Tektronix MSO 4054 .....	94
Figura 3.8-5 Diagrama de Conexión de Controlador SBGC. Extraído de (BaseCam Electronics, 2013).....	95
Figura 3.8-6 Placa SBGC en Gimbal CAME Mini 3-air. Tomado con cámara de Samsung S3.....	96
Figura 3.9-1 Acople Estandar Gimbal-UAV del CAME Mini 3-air. Extraído de (CAME-Tv, 2016).....	98
Figura 3.9-2 Octocóptero marca Ixion. Extraído de (Ixion, 2016).....	98
Figura 3.9-3 Dimensiones de Octocóptero. Extraído de (Ixion, 2016).....	99
Figura 3.9-4 Octocóptero marca Ixion y Gimbal CAME Mini 3-air Acoplados. Tomado con Cámara Canon EOS 400D.....	99
Figura 3.9-5 Detalle de Acople Octocóptero marca Ixion y Gimbal CAME Mini 3-air. Tomado con Cámara Canon EOS 400D .....	100
Figura 3.9-6 Diagrama de Unión CAME Mini 3-air con Cámaras tipo DSLR. Extraído de (CAME-Tv, 2016) .....	100
Figura 3.9-7 Primera Iteración del Acople Gimbal-Cámara Hipereespectral. Tomado con Cámara Canon EOS 400D.....	101
Figura 3.9-8 Prototipo para Balanceo de Eje Cabeceo. Tomado con cámara de Samsung S3 .....	102
Figura 3.9-9 Prototipo para Balanceo de Ejes Cabeceo y Alabeo. Tomado con cámara de Samsung S3.....	102
Figura 3.9-10 Plano Acople Gimbal y Cámara Hiperespectral, Desarrollado en OnShape .....	104
Figura 3.9-11 Acople de Gimbal y Cámara Impreso en 3D. Tomado con cámara de Samsung S3 .....	105
Figura 3.9-12 Prueba de Balanceo en Eje Cabeceo con Placa de Ajuste. Tomado con cámara de Samsung S3.....	105
Figura 3.9-13 Prueba de Balanceo en Eje Alabeo con Placa de Ajuste. Tomado con cámara de Samsung S3.....	106
Figura 3.11-1 Interfaz Win32 de Programa Final para Control de Gimbal y Pushbroom .....	108
Figura 3.11-2 Diagrama de Flujo de Programa Final.....	109
Figura 3.11-3 Secuencia de Operación de Barrido en Programa Final .....	110
Figura 9.3-1 Posibles Valores de la propiedad triggermode .....	131
Figura 9.3-2 Posibles valores de la propiedad mododisplay.....	131
Figura 9.3-3 Algunos valores posibles para la propiedad colormode.....	132
Figura 9.3-4 Ejemplo de Error en Función grandinicio .....	133

Figura 9.3-5 Tabla de Correspondencia entre Formato de Captura y Bits de Profundidad por Píxel ..... 134

## Índice de Tablas

Tabla 2.3-1 Comparación de Motores de Gimbal .....	32
Tabla 2.3-2 Comparación de Gimbal de 2 y 3 Ejes.....	33
Tabla 3.1-1 Características de Cámaras Hiperespectrales. Dada por el personal del GII y (Cubert) .....	39
Tabla 3.2-1 Primera Tabla de Modelos Comerciales de Gimbal .....	46
Tabla 3.2-2 Segunda Tabla de Modelos Comerciales de Gimbal .....	47
Tabla 3.2-3 Tercera Tabla de Modelos Comerciales de Gimbal.....	48
Tabla 3.2-4 Información de Modelos de Estructuras para Opción DIY.....	50
Tabla 3.2-5 Información de Modelos de Controladores para Opción DIY .....	51
Tabla 3.5-1 Datos para Regresión Lineal.....	73
Tabla 3.7-1 Mediciones de Tiempo de Guardado en Laptop de Pruebas.....	86
Tabla 3.7-2 Mediciones de Tiempo de Guardado en Mini-PC Fitlet-iA10 .....	86
Tabla 3.8-1 Información a Enviar para Encendido o Apagado de Motores de Controlador Simple BGC ...	93
Tabla 3.8-2 Información a Enviar para Comando Control de Controlador Simple BGC .....	94
Tabla 3.8-3 Datos para Comando Control de Prueba .....	94
Tabla 9.2-1 Medición de Proporciones de Imagen y Valor F para Todas las Pruebas .....	126

## 1. Introducción

Como parte de la creciente preocupación por la tecnificación de la agricultura, el Ministerio de Economía y Competitividad del Gobierno Español ha planteado el proyecto “Sistema Integral para la Monitorización Multiresolución en Agricultura de Precisión” (SIMMAP) buscando responder a la necesidad de incorporación de tecnologías TIC en la industria agrícola. El proyecto SIMMAP se desarrolla entre varios centros de investigación así como universidades. Entre estas, el Grupo Integrado de Ingeniería (GII) de la Universidad de la Coruña (UDC), campus de Ferrol, se encarga de la medición hiperespectral mediante vuelo de baja altura. Entre los problemas iniciales que presenta esta solicitud, se encuentra la forma y modo en que se pueden acoplar, controlar y verificar las dos tecnologías principales: los sensores y cámaras hiperespectrales y el UAV.

El estudiante de ingeniería Mecatrónica del Instituto Tecnológico de Costa Rica, Rodolfo Piedra Camacho, carné 201239346, trabajó en conjunto con investigadores del GII en el desarrollo y solución de los problemas iniciales de acople y control del octocóptero y las cámaras hiperespectrales. Dentro del proceso de solución se contempla la selección de un sistema estabilizador comercial, así como el desarrollo de bibliotecas y aplicaciones ejecutables en C++ para el control del equipo a utilizar. Por su parte, el personal del GII aportó el equipo de pruebas necesario así como el apoyo en temas de dirección del proyecto, compra de materiales y ayuda directa del personal.

El Grupo Integrado de Ingeniería es un grupo interdisciplinar de investigación aplicada en ingeniería, orientado a la transferencia de conocimiento y a la generación de nuevos productos en el entorno industrial. Ubicado en el Campus de Ferrol de la Universidad de la Coruña, el GII está formado por un grupo multidisciplinar de profesionales. Dado su origen y vinculación a las Escuelas de Ingeniería de la UDC, se da un predominio en las disciplinas de: ingeniería naval, industrial, eléctrica, informática, telecomunicaciones y diseño. Con más de 200 proyectos de investigación desarrollados, el GII se posiciona como un grupo de referencia en sus ámbitos de actuación, estableciendo colaboraciones estables con organismos relevantes nacionales e internacionales.

## 1.1. Planteamiento del Problema

La agricultura, como medio de producción de comida y sustento económico de una nación, es una actividad que no ha perdido vigencia y que se ha mantenido a lo largo de la historia; es tal su necesidad que se han diseñado una variedad de estrategias y herramientas para su mejora y eficiencia: planificación de cultivos, insecticidas, abonos, entre otros medios. Si bien la mejora y expansión de la agricultura es un beneficio, también contempla un impacto ambiental notable por el consumo de agua, energía y desgaste de suelos.

Ante este contexto se suma el crecimiento tecnológico, el cual ha generado una variedad de dispositivos, sensores o algoritmos que permiten monitorizar el estado de los cultivos o suelos; no obstante, dichos medios son raramente utilizados, sea por la extensión de los campos de cultivos, la falta de lógicas de recolección, la interpretación de datos o bien el elevado costo que contempla la instalación de los sensores a nivel terrestre.

Como forma de aproximarse al reto de unificar los avances tecnológicos con las labores de agricultura, el Ministerio de Economía y Competitividad del Gobierno Español ha planteado el proyecto “Sistema Integral para la Monitorización Multiresolución en Agricultura de Precisión” (SIMMAP), en el cual se aborda la monitorización de cultivos a distintos niveles: baja altura, nivel de suelo, altura elevada, entre otros. En lo que concierne al proyecto, se entiende por “baja altura” elevaciones cercanas a los 100 metros sobre el nivel del suelo. El proyecto se desarrolla entre varias universidades o centros de investigación; entre estos la Universidad de La Coruña (UDC), específicamente el Grupo Integrado de Ingeniería (GII).

Como aproximación a los problemas de recolección y envío de datos en temas de agricultura, el GII ha optado por la opción de aplicar sensores hiperespectrales en conjunto con sistemas de vuelo de baja altura, particularmente drones octacópteros. Todo con el fin de poder obtener información de calidad y en una cantidad tal que permita la toma de decisiones sobre los pasos a seguir en la gestión de cultivos y suelos.

A esta solución se presenta una problemática inicial, la forma y método en que se deben integrar ambas tecnologías, octacóptero y cámaras hiperespectrales, de tal forma que se puedan aprovechar las ventajas particulares de cada tecnología. En este sentido un vehículo aéreo no tripulado, UAV por sus siglas en inglés, presenta la principal ventaja de que puede tomar información, a baja altura, de una expansión considerable de cultivos pero impone dificultades sobre el peso que puede elevar y el tiempo de vuelo.

Por otro lado, los sensores hiperespectrales permiten obtener cientos de imágenes correspondientes a diferentes bandas espectrales muy próximas entre sí. Como resultado se generan cubos de datos, en los que dos dimensiones representan la ubicación espacial de un píxel y una tercera dimensión proporciona los valores de reflectancia de cada píxel a distintas espectro de onda; no obstante, como limitante a estas capacidades imponen requisitos de estabilidad en la captura, así como dificultades en su operación por necesidad de barridos para la captura.

Por estas razones el problema a desarrollar es la etapa inicial de la parte del GII del proyecto SIMMAP. Poder integrar ambas tecnologías, contemplando las dificultades que cada una presenta, en un único sistema que pueda estabilizar las imágenes, mantener una sujeción mecánica adecuada y que pueda controlar debidamente la cámara con programas de bibliotecas modificables.

## 1.2. Objetivos del Proyecto

### 1.2.1. Objetivo General

Desarrollar un prototipo que permita captura de imágenes hiperespectrales mediante un UAV, partiendo de dos cámaras y un UAV previamente seleccionados, velando por aspectos de estabilización, rigidez y control de la cámara.

### 1.2.2. Objetivos Específicos

#### 1- Parte Mecánica:

- Seleccionar el sistema de estabilización comercial considerando su ajuste al sistema de fijación.
- Diseñar el sistema de fijación de la cámara hiperespectral partiendo de un UAV comercial previamente seleccionado.

#### 2- Parte Electrónica:

- Implementar el sistema de control tanto del estabilizador comercial como de la cámara hiperespectral de manera que permita la captura y almacenado de imágenes, así como la estabilización y movimiento adecuado de la cámara.

#### 3- Prototipo:

- Verificación del funcionamiento correcto de las distintas partes del proyecto así como la entrega de un prototipo final listo para realizar pruebas de campo.

### 1.3. Alcances y Limitaciones

En esta sección se detallan los alcances y limitaciones del proyecto; es decir, se hace una mención de los distintos temas que no se trabajan a lo largo del proyecto indiferente de la incidencia de dichas secciones en el proyecto SIMMAP como un todo. Se presenta cada una seguida de una explicación con más detalle.

- **Consideraciones de Alimentación y Fuente de Potencia:** Si bien parte del proyecto implica el desarrollo de la electrónica de control y la elección de un sistema gimbal, ambos con exigencias de energía, contemplar este aspecto queda fuera de las labores desarrolladas; no obstante, si es válido recopilar los datos afines a este tema para apoyar al encargado respectivo del GII en dicha labor.
- **Aspectos a Considerar sobre el Dron:** El proyecto requiere contemplar y considerar el dron a utilizar, esto se extiende al análisis de hojas de datos, dimensiones o planos en caso de ser necesario; mas no existirá incidencia alguna sobre el dron a utilizar o modificaciones que se tendrá sobre este. En este sentido para efectos del proyecto el dron impone condiciones a considerar y no se actúa sobre el mismo.
- **Trabajo Sobre Información Hiperespectral:** El trabajo que se realizó sobre la información proveniente de las cámaras hiperespectrales siempre será en cuanto a la manipulación y presentación de la información, esto abarca la creación de cubos espectrales o la presentación gráfica de la información de una zona específica; no obstante, en ningún momento se interpreta la información hiperespectral capturada por las cámaras.
- **Selección de Hiperespectrómetro:** La presencia de dos hiperespectrómetros, uno tipo pushbroom y uno snapshot, implica que para el proyecto SIMMAP se debe tomar la decisión de cual usar o bajo cuales condiciones se debe usar cada uno. El presente proyecto no abarca la elección en este tema; no obstante, si se considera parte del proyecto las distintas labores requeridas que puedan ayudar a los encargados del GII a tomar la decisión final. En otras palabras si bien la elección no es parte del proyecto, ciertas labores del proyecto si pueden ayudar a tomar la decisión.
- **Comunicación entre el Dron y el Controlador a Implementar:** Dentro de los objetivos se menciona el desarrollo de la electrónica, así como la lógica, necesaria para controlar la cámara hiperespectral y el estabilizador comercial a utilizar. Esto no implica que se diseñara o desarrollara el método bajo el cual la estación de trabajo principal, o en su defecto el sistema de control del Dron, se comunica con la electrónica implementada. En este sentido los entregables son capaces de controlar la cámara y el gimbal según las necesidades del GII pero no se especifica como el dron puede acceder o manipular dichos entregables.
- **Implementación de Estabilizador Comercial de la Cámara:** Más allá de la elección del estabilizador, se debe implementar el mismo de tal forma que pueda estabilizar la cámara hiperespectral; si bien esto puede implicar la modificación y asignación de parámetros tipo PID, el proyecto no contempla un análisis matemático y de teoría de control automático denso. En este sentido se apoya más de los métodos facilitados por el proveedor, por esta misma razón se excluye la presentación de modelos matemáticos o similares en el desarrollo de la solución.

## 2. Marco Teórico

En este capítulo se detallan los distintos conceptos teóricos que abarcan tanto el problema como la solución realizada en el proyecto, su utilidad radica especialmente en las partes iniciales de la solución. Propiamente el marco teórico se divide en tres secciones: Análisis, comprensión y explicación de las imágenes hiperespectrales en conjunto con los dispositivos que las generan, aborde del concepto y términos de interés sobre UAVs y finalmente una caracterización y descripción sobre los sistemas de estabilización inercial. Dado el trabajo realizado se da un énfasis en la primera y última sección, buscando profundizar en una mayor cantidad de conceptos.

### 2.1. Imágenes y Cámara Hiperespectrales

La luz, así como otras formas de radiación electromagnética, se describe principalmente, pero no de forma exclusiva, por su longitud de onda. Como ejemplo de esto se tiene la luz visible, que ronda los 400 – 700 nm y las ondas de radio que rondan los 30 cm. Una imagen hiperespectral es una imagen que almacena la radiación de una gran cantidad de bandas espectrales que emiten los cuerpos terrestres al reflejar la energía del sol u otra emisión de luz (Priego Torres, 2011).

Para comprender propiamente el concepto se puede comparar una imagen hiperespectral con una imagen en escala RGB tradicional. Las imágenes RGB constan de tres capas en las cuales se almacena la luz reflejada por el objeto capturado a las longitudes de onda que caracterizan los colores azul, rojo y verde. Por otro lado una imagen hiperespectral consta de una mayor cantidad de capas, donde cada capa almacena alguna banda estrecha de longitudes de onda específicas. Las figuras 2.1-2 y 2.1-2 sirven de explicación gráfica del concepto explicado.

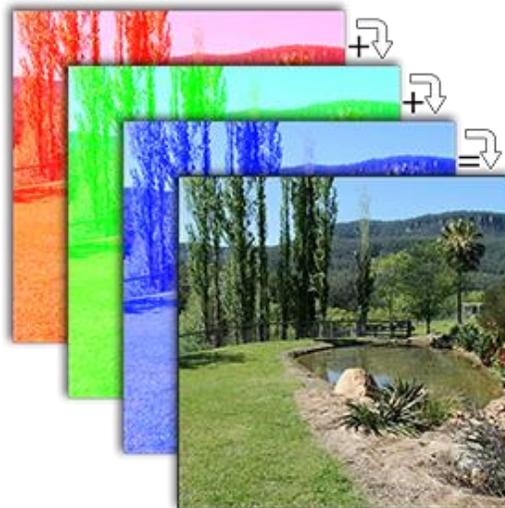


Figura 2.1-1 Concepto de Imagen RGB. Extraído de (Scyllarus, 2016)

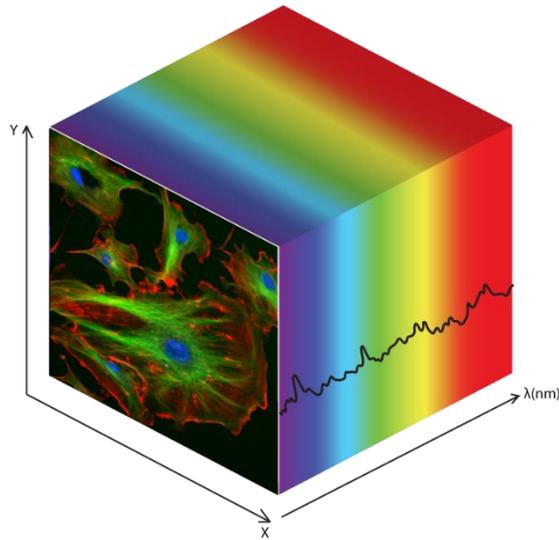


Figura 2.1-2 Concepto de Imagen Hiperespectral. Extraído de (Photon, s.f.)

En la práctica una imagen hiperespectral se puede entender como un cubo de datos. Es decir una imagen donde sus dimensiones son: horizontal, vertical y una dimensión de reflexión a una longitud de onda dada, normalmente entre el visible y el infrarrojo.

La principal utilidad de las imágenes hiperespectrales viene en lo que se conoce como firma espectral, lo cual es la forma específica en que cada material interactúa con las distintas longitudes de onda. Entiéndase la firma espectral como una huella dactilar para un material.

Una imagen hiperespectral se adquiere con un sensor hiperespectral y la cámara encargada de esto se denomina hiperespectrómetro.

### 2.1.1. Características de los Sensores Hiperespectrales

En un sentido propiamente técnico las características de un sensor hiperespectral son:

- **Resolución Espectral de Sensor:** Se refiere a la cantidad de bandas espectrales y el ancho de dichas bandas. Entre mayor cantidad de canales mejor se clasifican los materiales y entre más estrechos los canales menos problemas por promedio de valores.
- **Resolución Radiométrica:** Se refiere a la sensibilidad del sensor, es decir que tanto puede variar el valor almacenado ante una misma longitud de onda. Esto depende de los bits utilizados para codificar la luz reflejada.
- **Resolución Espacial:** Dato que describe la relación milímetros por píxel de una imagen. Propiamente se basa en campo de visión del sensor el cual determina el tamaño y forma de cada píxel.
- **Relación Señal/Ruido:** Relación entre amplitud de señal contra amplitud de ruido. Depende de la sensibilidad del sensor, la anchura espectral y la cantidad de radiación reflejada por el medio.

A estas características se les debe agregar una aclaración, tanto un sensor como una imagen hiperespectral no se definen por la cantidad de longitudes de onda medidas sino por el ancho y continuidad de las bandas como tales: un sensor o imagen hiperespectral requieren de bandas espectrales angostas muy cercanas entre sí. Por ejemplo 20 bandas consecutivas de 10 nm de ancho se considera hiperespectral, pero 20 bandas consecutivas de 100 nm o bien 20 bandas discontinuas se considera multiespectral. (Priego Torres, 2011)

### 2.1.2. Tipo de Hiperespectrómetros

En lo que se refiere a tecnología de hiperespectrómetros se puede reducir las variedades a tres grandes categorías según el tipo de escaneo: Por punto, línea o plano (Priego Torres, 2011).

- **Escaneo de Punto:** mide y toma imagen del espectro en una región pequeña, presenta una resolución espectral alta y estable pero su metodología de barrido es compleja y de alto costo.
- **Escaneo de Línea:** también conocido como escaneo push-broom, la muestra se toma mediante una rendija que limita el paso de la luz para así tomar únicamente una línea de la escena. Para tomar una imagen completa se requiere de un barrido en una dimensión; la velocidad de toma de imagen se ve limitada por la velocidad de captura propia de la cámara y la velocidad de barrido.
- **Plano Focal:** similar a una fotografía natural, donde se obtiene toda la información tanto espacial como espectral. Según el estilo propio en el que se toma la imagen se pueden definir dos tipos de hiperespectrómetros de plano focal (Aasen, Burkart, Bolten, & Bareth, 2015).
  - **Snapshot:** sistemas en los cuales la información espectral se obtiene en una única instancia de entrada de luz. La resolución espacial tiende a verse reducida, por ejemplo en un orden de 1000 a 50 píxeles.
  - **Cuadro de Imagen:** sistemas en los cuales la información espectral se obtiene una, o varias bandas espectrales a la vez pero sin abarcar todo el espectro en una única exposición. Requieren múltiples tomas de imagen para poder completar el cubo de datos, por lo general incurren un cambio de filtro o el uso de sub-cámaras.

### 2.1.3. Forma de Operación y Funcionamiento de Cámaras Hiperespectrales

El funcionamiento de las cámaras hiperespectrales varía de un tipo a otro. En el GII se tiene acceso a cámaras de tipo push-broom de las cuales en el documento (Priego Torres, 2011) explica el principio óptico de las mismas y el método particular con que funciona el hiperespectrómetro basado en cámara Retiga Exi.

En lo que se refiere a óptica la figura 2.1-3 muestra los distintos componentes de un hiperespectrómetro push-broom.

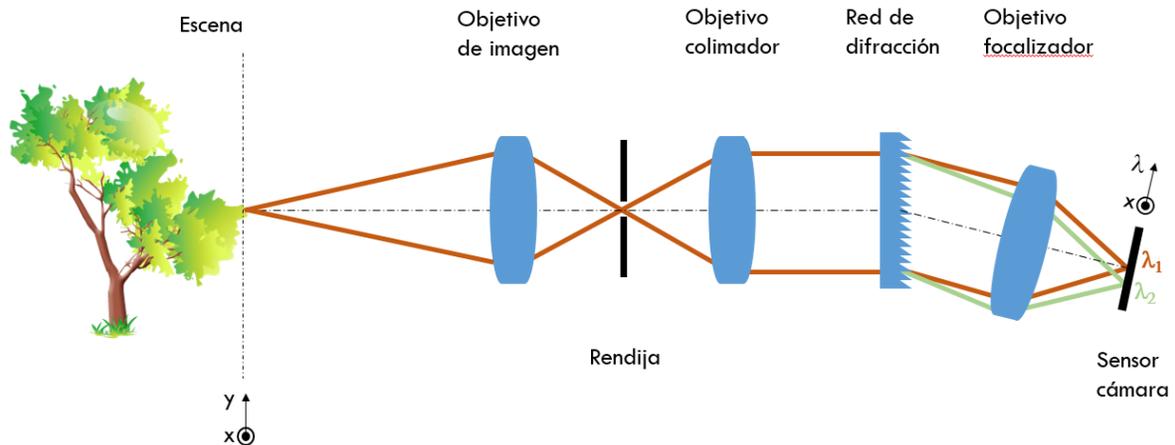


Figura 2.1-3 Sistema Óptico de Cámara Hiperespectral. Imagen proveída por el GII

Las distintas labores de cada elemento se especifican a continuación.

- **Escena:** La escena a tomar imagen, cada captura adquiere una única línea en el sentido del eje X, para poder tener la imagen completa se debe realizar un barrido a lo largo del eje y.
- **Objetivo de la Imagen:** Se encarga de reflejar y enfocar el objeto en cuestión de tal forma que los haces de luz caigan exactamente en la rendija.
- **Rendija:** Permite que solo una banda estrecha de luz sea captada por la cámara, la rendija tiene a ser del orden de los micrómetros. Según la imagen la línea obtenida es del eje X. La rendija debe situarse en el plano focal del objetivo colimador.
- **Objetivo Colimador:** Toma un haz de luz divergente y desordenado y lo convierte en un haz de luz paralelo y homogeneizado.
- **Red de Difracción:** Descompone la luz en las distintas componentes de diferentes longitudes de onda. La cantidad de ranuras y la separación de estas define la cantidad y espesor de las longitudes de onda. Tiene el efecto de alterar la dirección del haz de luz.
- **Objetivo Focalizador:** Similar al objetivo de imagen se encarga de que los haces de luz converjan en el sensor de la cámara.
- **Sensor de Cámara:** Capta la imagen final la cual posee dos dimensiones aun cuando en la realidad solo se está tomando información espacial del eje X. La segunda dimensión posee la información espectral de las distintas longitudes de onda.

La cámara hiperespectral basada en cámara Retiga Exi se basa en el principio explicado y agrega además un sistema de rotación de espejo para poder realizar el barrido y recolección de todas las líneas de la imagen. Inicialmente el hiperespectrómetro entrega en su salida una única imagen que responde a la información de reflexión de todas las longitudes de onda para una única línea espacial. Aplicaciones en C++ y los motores de barrido permiten obtener una carpeta con todas las imágenes de una escena dada.

Por otro lado el hiperespectrómetro basado en cámara UI3360 CP-NIR, comparte el mismo principio de función, pero al carecer de un sistema de barrido tiene por salida una imagen con toda la información espectral de una única línea de información espacial. Para su conexión se puede hacer uso de un cable USB o bien una variedad de pines tipo macho.

En las figuras 2.1-4 y 2.1-5 se pueden apreciar los hiperspectrómetros basados en cámara Retiga y UI3360 respectivamente.



*Figura 2.1-4 Hiperspectrómetro con Cámara Retiga Exi. Extraído de (Priego Torres, 2011)*



*Figura 2.1-5 Hiperspectrómetro con Cámara UI3360 CP-NIR. Imagen proveída por el GII*

## 2.2. UAV

Un vehículo aéreo no tripulado, o UAV por sus siglas en inglés, también conocido como sistema aéreo autónomo, es una aeronave que opera sin la necesidad de un operador humano a bordo. Tradicionalmente de pequeña escala, los UAV son popularmente conocidos como drones cuando el control de los mismos carece de autonomía y se basa en algún tipo de comunicación a distancia como lo puede ser radiofrecuencia o similares; por otro lado existen UAVs que aplican algoritmos para la autonomía en sus vuelos, sea la programación de rutas de vuelo, el control de velocidades y altitud o bien el manejo de sensores para la toma de información (Dingjie, Hanfeng, & Wu, 2016).

Cabe aclarar que dentro del concepto de UAV no solo se contempla el modelo de multirrotores, sino también el llamado dron ala fija; para efectos del presente trabajo los conceptos expuestos son básicos y pueden aplicar para ambos casos, pero se debe tener la noción correcta de la existencia de cada cual.

### 2.2.1. Parámetros Limitantes de Operación

Importante para poder tener una noción general sobre el concepto de UAV, es conocer algunos de los parámetros que limitan la operación de los UAVs. Propiamente (Magnussen, Hovland, & Ottestad, 2014) enumera algunas de las características limitantes en el vuelo de drones:

- Tiempo de Vuelo
- Costo
- Capacidad de Carga
- Operación Dinámica: En este caso se refiere a las capacidades rotacionales de los motores a distintas condiciones de carga
- Revoluciones por minuto de los motores

Los parámetros dados son características que limitan las misiones o labores que puede desarrollar un UAV determinado; otras características que pueden influir sobre las cinco previas o bien que caracterizan a los UAV como tales son: Cantidad de motores, tipo de motores, baterías, peso de aeronave y el momento de inercia. Este último particularmente tiene incidencias en las capacidades de giro según los motores utilizados, lo que a su vez se traduce en la capacidad y agilidad de movimiento.

Finalmente cabe mencionar, que para efectos del trabajo desarrollado, el peso es el principal agente de perturbación sobre la operación de un UAV; es decir a mayor peso mayor potencia se le exige a los motores y por ende menos tiempo de vuelo.

### 2.2.2. Sistema de Orientación

Necesario para el vuelo de un UAV, e importante para el desarrollo del proyecto, es el sistema de orientación rotacional utilizada por este tipo de vehículos. El sistema se basa en el conocimiento de tres ángulos de Euler: Guiñada (Yaw), Cabeceo (Pitch) y Alabeo (Roll); con el conocimiento de estos ángulos un UAV puede no solo definir su orientación en cualquier momento sino también incidir sobre la trayectoria de su movimiento (Barton, 2012). En la figura 2.2-1 se puede observar de forma gráfica la localización estándar de los ejes y el sentido de giro que implica cada ángulo:

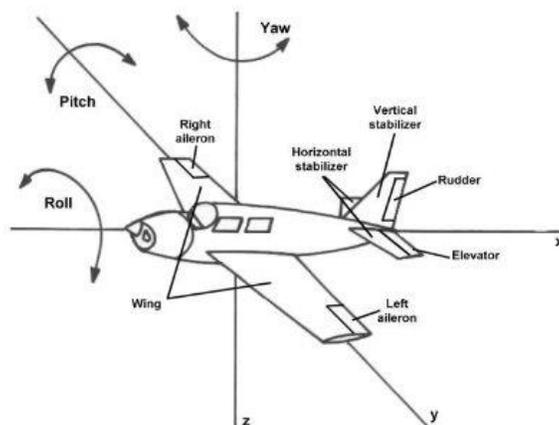


Figura 2.2-1 Ángulos de Euler sobre un UAS. Tomado de (Fiorenzani, 2016)

Si bien en la imagen se presenta un UAV distinto al hexacóptero o cuadracóptero, el principio es el mismo siempre y cuando se considere que el eje longitudinal del vehículo, el cual demarca el frente del mismo, define el eje de alabeo. Específicamente cada uno de los movimientos rotacionales implica:

- Alabeo: Rotación sobre el eje principal, en el caso de los UAV permite el movimiento de izquierda o derecha sin cambiar la dirección en que apunta la parte frontal.
- Cabeceo: Rotación que inclina la parte frontal, en el caso de los UAV permite el movimiento en adelante o reversa.
- Guiñada: Rotación que permite cambiar la dirección en la que el frente del UAV o UAS apunta.

### 2.3. Sistema de Estabilización Inercial: Gimbal

En principio un gimbal es una estructura mecánica, basada en soportes pivotados, que permite aislar un objeto de su movimiento rotacional, ante un único eje, indiferente del movimiento rotacional que experimente la propia estructura (Chatfield, 2000). De forma práctica y según la aplicación, un gimbal es tanto un instrumento de balanceo como un instrumento de medición que actúa en el campo de la cinemática rotacional.

Originalmente el término de gimbal se utilizó para denotar un único anillo de la estructura presente en la figura 2.3-1. En este sentido un gimbal es básicamente un derivado del giroscopio utilizado principalmente por sus capacidades de conservación de la orientación y no tanto por sus aplicaciones en la medición.

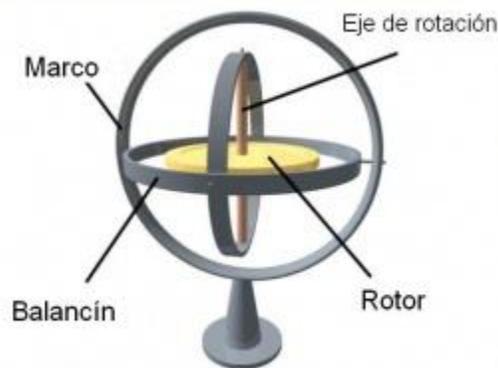


Figura 2.3-1 Imagen de Gimbal/Giroscopio Conceptual. Extraído de (Gyroscopes.org, 2016)

Si bien el concepto de gimbal se remonta al del giroscopio, se debe aclarar que de forma comercial y popular el concepto de gimbal hace referencia a estructuras como las que se aprecian en las figuras 2.3-2 y 2.3-3



Figura 2.3-2 Gimball Comercial de Cámara Pequeña. Extraído de (Ebay, 2016)



Figura 2.3-3 Gimbal Comercial de Cámara Grande. Extraído de (ProKit, 2015)

Los gimbals de las figuras 2.3-2 y 2.3-3 se utilizan principalmente en las aplicaciones de fotografía, cine y lo afín a UAV. En esta variación la labor de los anillos concéntricos se sustituye por la acción de motores, sensores y una lógica de control automático. Todo en conjunto permite el mismo resultado y en el caso de la aplicación en UAV le provee una utilidad nueva: Estabilizar la línea de visión, LOS por sus siglas en Inglés, de una cámara aún en presencia de las interferencias producidas por el movimiento del vehículo (Lee & Yoo, 2007). En ciertos casos los gimbals se pueden caracterizar, entre otras cosas, con valores de precisión angular, con unidades de grados, velocidad de reacción, con unidades de grados por segundo, su tamaño, peso y carga de diseño.

De esta forma un gimbal comercial, también conocido como Plataforma de Estabilización Inercial, ISP por sus siglas en Inglés, es básicamente una estructura metálica que alberga motores y cojinetes que en conjunto con los sensores y lógica de controla adecuados permite aislar un sistema óptico de las perturbaciones generadas por movimiento sea el efecto de la gravedad, vibraciones o movimientos repentinos de la estructura (Abdo, Toloie, Vali, & Arvan, 2014).

En el proyecto presente se utiliza un Gimbal de tipo comercial para mantener la orientación de la cámara hiperspectral. En este momento se procede a declarar la terminología con la cual se hará referencia a las partes del gimbal:

- Marco: La zona donde se coloca la cámara, el marco gira en conjunto con la cámara en el sentido del eje pitch.
- Estructura: El resto del gimbal que no sea el marco, contempla los motores, las conexiones entre ejes y la sujeción a UAV o las agarraderas del usuario.

Para poder comprender con mayor precisión la forma en que funcionan los gimbal comerciales es necesario explicar y/o comentar los distintos elementos, tanto físicos como lógicos, que lo componen.

### 2.3.1. Lógica de Control

Común a toda lógica de control, para poder controlar una o varias variables de interés es necesario tener tanto una forma de medición de la variable, como un método para actuar o alterar la variable; en la mayoría de los casos resulta ser algún otro parámetro previo en la secuencia de funcionamiento. En el caso de un sistema Gimbal el método de actuar es la tensión que alimenta los motores; la tensión provoca que el motor rote y así estos varían el valor de los ángulos alabeo-cabeceo-guiñada para corregir los errores del estado actual (Brake, 2012).

Por otro lado la variable a analizar puede variar según la aplicación específica, existen modelos que controlan la orientación propia de la cámara así como otros que controlan la línea de vista o LOS por sus siglas en inglés; ambos casos dependen e inciden sobre los ángulos alabeo-cabeceo-guiñada pero tienen objetivos y resultados distintos.

- **Control Sobre la LOS:** El control sobre la LOS consiste en que la cámara apunte a un objetivo dado en todo momento; se entiende por apuntar que el objetivo se localice dentro del campo de visión (Brake, 2012), FOV por sus siglas en inglés, el concepto detrás de este tipo de control se puede apreciar en la figura 2.3-4. En este sentido el ángulo con el que se observa el objetivo y la orientación de la cámara pueden variar constantemente, que mientras se mantenga el objetivo en el centro del FOV el lazo de control cumple adecuadamente su propósito.

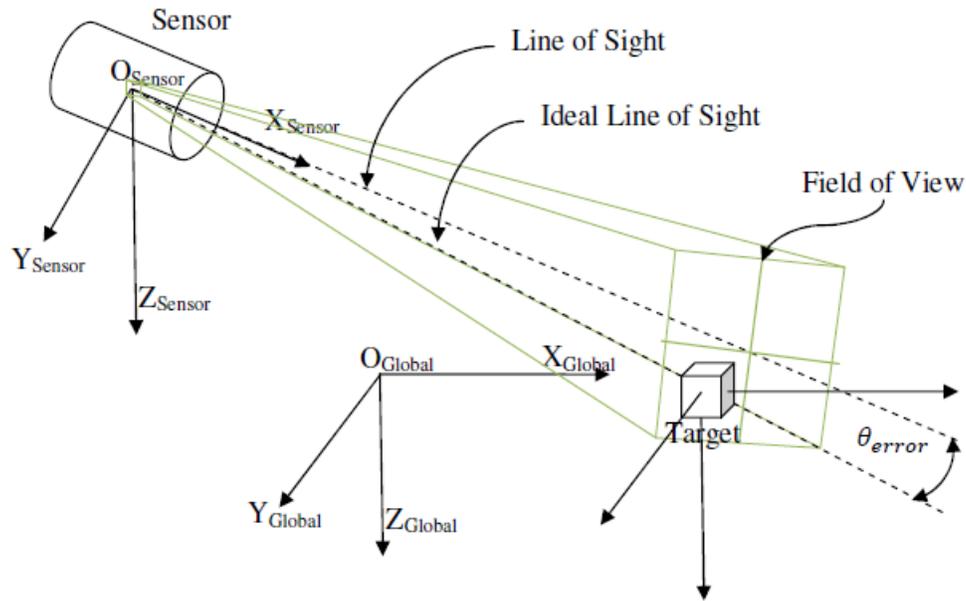


Figura 2.3-4 Concepto de Control de LOS. Extraído de (Brake, 2012)

- **Control Sobre la Orientación de la Cámara:** Comercialmente conocido como “Lock/Hold Mode”, se busca mantener fija la orientación de la cámara; en otras palabras, preservar los valores exactos de los ángulos alabeo-cabeceo-guiñada, respecto a una referencia estática. Puede observarse la figura 2.3-5 para comprender el estilo; en la etapa I se observa la posición básica tanto de la cámara como del dron, en la etapa II se observa como un movimiento del dron introduce una perturbación en la orientación de la cámara (exagerada para fines de la explicación) y finalmente en la etapa III se observa como el control por orientación ayuda a la cámara a retornar a su posición original.
- **Control de Seguimiento:** Común en los Gimbal usados para filmes o fotografía, se basa en seguir la orientación, en uno o dos ejes, del marco estructural que sostiene el gimbal, denominada en este caso “posición neutral”. Normalmente el seguimiento se aplica solamente en los ejes de cabeceo y guiñada. Siguiendo el ejemplo de la figura 2.3-5, el control por seguimiento sería similar solo que en la etapa III en vez de retornar a la orientación original en todos los ejes, solo lo haría en uno o dos.
- **Control de Visión en Primera Persona:** En aplicaciones de UAV y similares el modo de primera persona es un control que busca alinear la orientación de la cámara al marco de referencia del UAV, en términos prácticos busca que el usuario sea capaz de usar el gimbal y la cámara como los ojos del UAV.

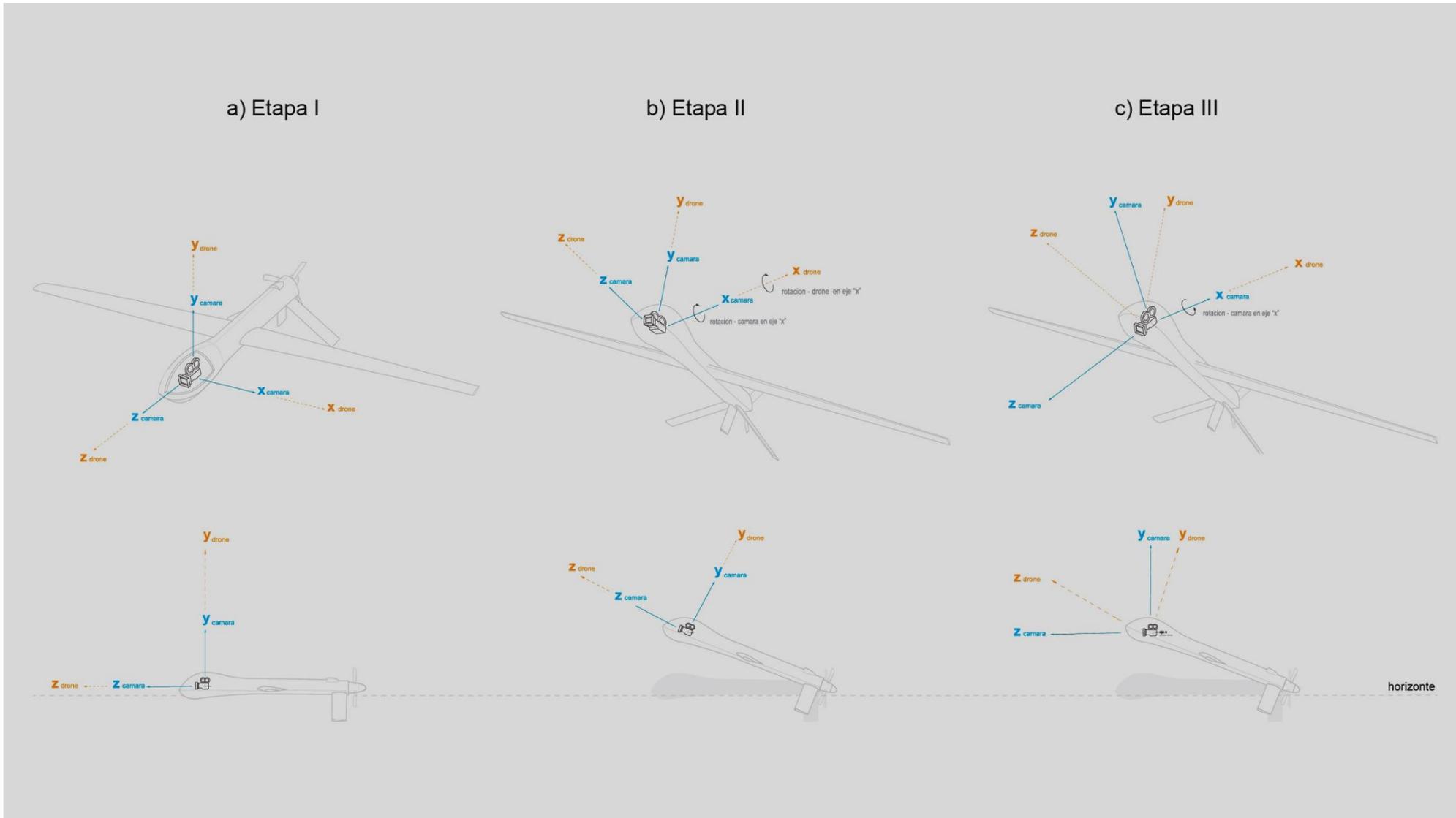


Figura 2.3-5 Lógica de Control por Orientación, ilustrado por Andrés Elizondo González estudiante de Ingeniería en Diseño Industrial del ITCR

### 2.3.2. Instrumentación y Mediciones Necesarias

La metodología de función de un gimbal depende de las mediciones sobre la orientación de la cámara a estabilizar, dichas variables permiten definir la tensión que debe llegar a cada motor para poder asegurar la respuesta adecuada. Para poder acceder a la información de la orientación se hace un uso combinado de acelerómetros, giroscopios y en algunas ocasiones magnetómetros los cuales en conjunto se conocen como IMU o unidad de medida inercial por sus siglas en inglés (Lin & Yang, 2014).

Una IMU es un dispositivo de medición con seis grados de libertad que permite la medición de la aceleración en tres dimensiones y la medición de la velocidad de rotación en tres sentidos de giro (Jarrah & Sahawneh, 2008). Las labores específicas de cada elemento se pueden resumir de la siguiente manera:

- Giroscopio: Instrumento que consiste en una masa rodeado por anillos que le permiten conservar su orientación y momento angular. Permite medir la orientación de un objeto en todo momento.
- Acelerómetro: Utiliza la inercia de una masa para medir la diferencia en la aceleración cinemática respecto a un espacio inercial, esto según su orientación, y la aceleración de la gravedad (Barton, 2012). En un sentido práctico permite la medición de la aceleración que actúa sobre el sensor, al orientarlo sobre distintos ejes puede medir la aceleración en tres dimensiones espaciales distintas.
- Magnetómetro: Instrumento capaz de medir campos magnéticos leves, posee una medición tipo brújula que carece de error por integración; normalmente se utiliza para medir el campo magnético de la tierra y así conocer la diferencia de ángulo contra el Polo Norte terrestre (Shahid, Shahzad, Bukhari, & Abasi, 2016). En el caso de un gimbal permite conocer la rotación sobre el eje guiñada.

En conjunto se puede entender que el giroscopio provee una referencia en el espacio unida al objeto a estabilizar, mientras que el acelerómetro, en presencia de un espacio definido, permite obtener información de la orientación y movimiento del objeto, de esta forma proveyendo información de los ejes cabeceo y alabeo. Por su cuenta el magnetómetro se encarga de la información sobre el eje guiñada (Chatfield, 2000).

Finalmente cabe destacar que según (Jarrah & Sahawneh, 2008) los IMU comerciales de baja calidad son aquellos que experimentan de una desviación mayor o igual a los seis grados por hora, es decir que requieren de una lógica de calibración constante y que pueden perder precisión en la duración del vuelo. En el mismo artículo se detalla algunos valores comerciales que son tanto accesibles económicamente como de buena calidad en las aplicaciones de gimbal:

- Acelerómetros con capacidad de medición a  $\pm 2g$
- Acelerómetros con sensibilidad de 12.5%/g, siendo esta la medida de precisión del acelerómetro
- Giroscopios con capacidad angular de hasta  $\pm 150^\circ$ . Este valor se traduce en el rango de operación del gimbal y tiende a reducirse en el eje pitch.
- Como regla de mano, a mayor resolución en bit, más preciso es el control que se puede obtener.

### 2.3.3. Motores

De forma comercial los gimbals se pueden encontrar en dos modelos, los que hacen uso de servomotores o los que hacen uso de motores sin escobillas (Liang, 2014).

Un motor sin escobillas es un motor controlado por una señal DC el cual se basa en un principio de imán permanente para su movimiento, por este mismo principio sufre de menor fricción y desgaste. En lo que se refiere a gimbal proveen mejor estabilización, una reacción rápida y mayor fluidez en su movimiento (Liang, 2014).

Por su cuenta el servomotor se basa en una lógica de control a lazo cerrado para proveer una regulación estable de la velocidad, torque o posición (Budimir, 2012). Combina un motor DC o AC, un encoder y un controlador para poder dar movimientos precisos o movimientos a velocidad o torque definidos. En lo que se refiere a gimbal los servomotores son más livianos, más económicos, simples de usar y autosuficientes; no obstante, el resultado final es de menor calidad (Liang, 2014).

Resumiendo la información provista por (Liang, 2014) se presenta en la tabla 2.3-1 la relación de ventaja o desventaja entre los gimbal con servo motores a los de motores sin escobillas. Se marca con una "X" aquel motor que tenga la ventaja en la categoría dada.

Tabla 2.3-1 Comparación de Motores de Gimbal

Categoría	Motor Sin Escobillas	Servo-motor
Peso		X
Precio		X
Sencillez de Uso		X
Simpleza de Instalación		X
Reacción a las Perturbaciones	X	
Repercusión en Calidad de Imagen	X	
Fluidez de Movimiento	X	

Cabe destacar que se pueden encontrar sistemas cerrados de gimbal con motores sin escobillas, en estos casos los problemas de uso e instalación pueden ser obviados ya que forman parte de las consideraciones y características propias del diseño del sistema. Además los modelos con motores sin escobillas predominan en el mercado de UAV o video y fotografía profesional mientras que los gimbal de servo motor predominan en las aplicaciones y comunidades de hobbies DIY o "hazlo tú mismo" por sus siglas en ingles.

### 2.3.4. Cantidad de Ejes a Controlar

Un punto importante en la elección, diseño y estudio de gimbals es la cantidad de ejes del mismo. Esto tiene implicaciones directas sobre el peso, capacidades y motores del mismo.

Los gimbal de dos ejes son sistemas de control comúnmente usados en aplicaciones donde la precisión exacta de la imagen no es un requisito mayor; poseen la ventaja de ser más livianos, de consumir menos energía y proveer mayor agilidad a la plataforma sobre la que se adecuan (Sham, 2015). Como lo comentan (Sham, 2015) o (Brake, 2012) las aplicaciones de fotografía aérea, vuelo en primera persona o

control sobre la LOS se pueden realizar adecuadamente con gimbal de dos ejes. Cabe destacar que los ejes a controlar varían de aplicación a aplicación.

Por otro lado los gimbal de tres ejes sufren de un mayor peso, precio y consumo energético, pero poseen la mayor estabilidad de los tres tipos; en este sentido son la mejor opción para aplicaciones de cine o videos (Sham, 2015). Los gimbal de tres ejes son los únicos capaces de realizar la lógica de control sobre la orientación y tienen una ventaja en el control de seguimiento.

La tabla 2.3-2 resume a manera comparativa las características más importantes los gimbal de dos y tres ejes.

Tabla 2.3-2 Comparación de Gimbal de 2 y 3 Ejes

Categoría	Gimbal 2 Ejes	Gimbal 3 Ejes
Peso	X	
Precio	X	
Consumo Energético	X	
Estabilidad		X
Tipos de Control a Favor	Sobre la LOS Visión en Primera Persona	Orientación Seguimiento

### 2.3.5. Balance de Gimbal

Desde un punto de vista mecánico, un gimbal requiere mantener un estado de balance para poder ejecutar sus funciones debidamente. En la figura 2.3-6 se aprecia un gimbal para cámaras ligeras con los ejes alabeo-cabeceo-guiñada identificados con los números 4-6-2 respectivamente, el estado de balance se alcanza cuando el centro de gravedad del objeto a equilibrar se alinea con la intersección de los tres ejes.

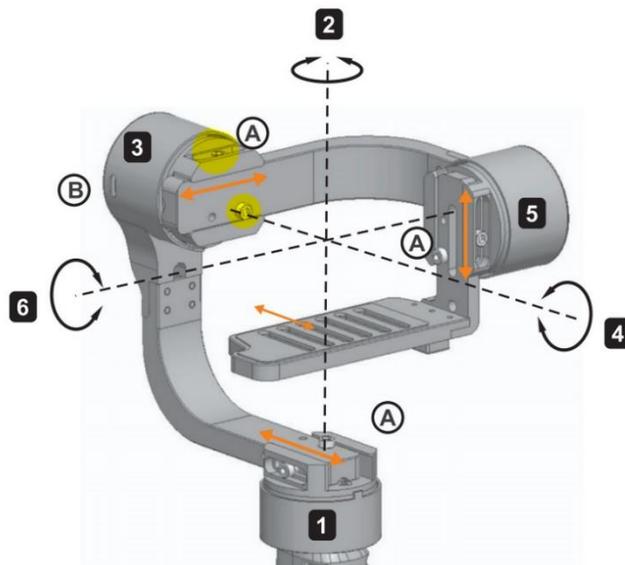


Figura 2.3-6 Ejes de Gimbal. Extraído de (The Photo Performance, 2015)

Como explica (Otlowski & Wiener, 2006), algunos de los efectos negativos de un mal balance en un gimbal son:

- **Oscilaciones:** El objeto a controlar se vuelve susceptible a oscilaciones por estímulos más pequeños, esto reduce la aplicabilidad del gimbal en condiciones con interferencias externas.
- **Efecto del Peso:** La desviación entre el centro de gravedad y los ejes de los motores provocan la aparición de un brazo de palanca el cual, en conjunto con el peso del objeto a estabilizar, generan un torque que rota el sistema.
- **Incidencia de las Vibraciones:** El efecto que provocan las vibraciones sobre el gimbal, ya sea que provengan de las manos de un usuario o el movimiento de algún vehículo, generan fuerzas sobre el centro de gravedad que a su vez se traducen en torques que generan oscilaciones no deseadas.

Todos los efectos tienen por principal consecuencia el desvío de la LOS en la captura de imágenes, lo cual incide en la calidad de esta. Los gimbals comerciales, no asociados a cámaras específicas, cuentan con formas de desplazar el centro de gravedad del objeto de tal forma que el gimbal pueda alcanzar el balance.

En la práctica la metodología de balanceo se centra en variar la posición del centro de masa de tal forma que el objeto no gire por su propio peso. La figura 2.3-7 presenta un diagrama general del proceso de balanceo de un eje, el proceso puede variar ligeramente entre gimbals pero la lógica es la misma en todos los casos.

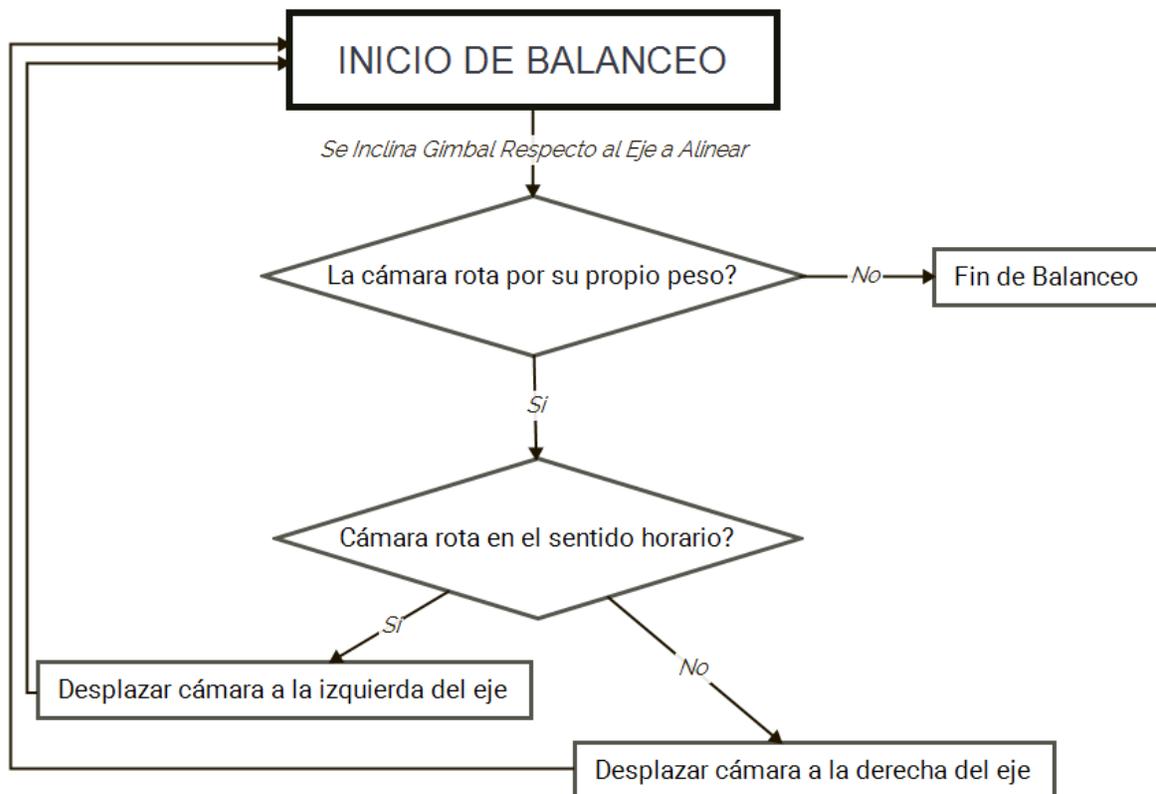


Figura 2.3-7 Diagrama de lógica de balanceo. Desarrollado en X Mind 7

De esta forma quedan presentados y explicados todos los conceptos afines y necesarios para la comprensión del contexto del proyecto, su solución, así como el equipo sobre el cual se estará trabajando. Se rescata de forma especial tanto el concepto de imágenes hiperespectrales como la naturaleza de los estilos de imagen snapshot y push-broom; ambos conceptos marcan un punto de partida en cuanto a las estrategias de algoritmos necesarias para la captura adecuada de un cubo de datos; por un lado las cámaras tipo snapshot requieren únicamente de una estabilización adecuada, mientras que las cámaras push-broom requieren de un barrido para poder capturar la información de una escena. Esta necesidad de un barrido controlado se tomara en cuenta en secciones posteriores donde se seleccione el equipo de estabilización.

Por otro lado se rescatan las explicaciones de ejes de balanceo, guiñada y cabeceo, tanto para comprender las perturbaciones que un UAV introduce al sistema estabilizador como vocabulario base y puente entre los temas de UAV y estabilizadores inerciales.

Finalmente el trabajo de caracterización de los gimbals, así como el análisis de los distintos elementos o lógicas que los rodean, serán de principal utilidad en las primeras fases de desarrollo de la solución, esto por su valor como criterio de toma de decisiones en cuanto a la adquisición de dichos sistemas.

## 2.4. Antecedentes y Estado del Arte

El presente estado del arte cumple tres propósitos: proporcionar una base o contexto de los caminos seguidos para la solución de problemas similares al proyecto SIMMAP, obtener información de equipos o métodos de solución que puedan guiar el proyecto y permitir la toma de decisiones sobre el uso de cámaras snapshot o pushbroom.

El problema de la toma de imágenes hiperespectrales para el estudio de cultivos ha ganado reciente popularidad en los medios de investigación, especialmente al considerar que la tecnología de hiperespectrómetros es cada vez más liviana y de menor tamaño lo que abre la posibilidad de combinar los mismos con UAVs. El estudio realizado por (Aasen, Burkart, Bolten, & Bareth, 2015), es una aproximación que busca utilizar cámaras del tipo snapshot para obtener información sobre la altura, biomasa verde y nivel de clorofila en un campo de cebada; en este sentido su estudio requiere que las imágenes tomadas puedan extraer información de tres dimensiones, otras de las características o especificaciones de la investigación contemplan:

- **Cámara:** Hiperespectrómetro tipo snapshot modelo UHD 185 de marca Cubert, capaz de tomar imágenes de 1000 x 990 en escala de grises al mismo momento de la toma de imágenes hiperespectrales de 50 x 50 píxeles. Logran una resolución en vuelo de 21 cm por píxel en la imagen espectral y 1 cm por píxel en la imagen a escala de grises. Captura en un rango de 450 a 950 nm. La cámara apunta abajo, perpendicular al suelo mediante la ayuda de espejos.
- **Gimbal:** Gimbal MK HiSight SLR2 con un peso de 280 g, usa servomotores para compensar los ejes cabeceo y alabeo, el documento no comenta sobre el dispositivo de control del gimbal pero si aclara que hacen uso de la IMU del UAV.

- **UAV:** MikroCopter Okto XL con carga máxima de 2.5 kg, tiempo de vuelo efectivo de 15-30 minutos. En los experimentos logra cruzar 1500 m<sup>2</sup> tomando 320 cubos de datos. Se encarga además de controlar el eje guiñada de la cámara al no variar su propia rotación sobre dicho eje.
- **Control de Cámara:** Conexión con computador PokiniZ mediante dos cables de Ethernet de 2 GHz, capaz de tomar y almacenar cubos de datos cada 1.6 s es decir 0.6 fps. Impone un peso de 1 kg en la carga.

En su experimento sobrevuelan un área de 1500 m<sup>2</sup> a 30 m de altura, tomando 320 imágenes a lo largo de 10 minutos, aplican calibraciones y correcciones al cubo de datos hiperespectral para luego expandir la misma sobre la imagen en escala de grises, aprovechando tanto resolución como información; esta propagación de información espectral se realiza con el algoritmo de vecino más cercano y da por resultado que un bloque de 20 x 20 píxeles o 20 x 19 píxeles tienen la misma información espectral correspondiente de un píxel del cubo 50 x 50, el proceso introduce un error del 1% en una dimensión de la imagen asociada a la resolución de 990.

Entre las conclusiones de su estudio destacan las ventajas del uso de cámaras snapshot sobre el tipo push-broom, especialmente por temas de facilidad de georeferencia, reducción del peso a elevar por el UAV y sensores secundarios. Recomiendan además reducir el rango de captura de 455-850 nm, esto dado que el sensor hiperespectral utilizado pierde sensibilidad en los extremos de su rango de análisis. Finalmente reafirman que la ventaja de obtener una imagen en escala de grises en conjunto con una imagen hiperespectral se encuentra en el ahorro de sensores y algoritmos de construcción de imagen, estos fueron cambiados por una simple interpolación de valores del cubo de datos sobre la imagen en escala de grises.

Otra aproximación al problema la presentan (Suomalainen, y otros, 2014), en la cual hacen uso de una cámara tipo push-broom para tomar datos de cultivos de cebolla y generar un modelo digital de la superficie en 3D, esto valiéndose de distintas estrategias de procesamiento de imágenes para prescindir de equipo costoso o pesado.

En este estudio la cámara a utilizar no posee un sistema de Gimbal sino un GPS-INS, o sistema de posicionamiento global con sistema de navegación inercial por sus siglas en inglés, el cual se encarga de tomar los valores de orientación tanto en posición global como de rotación para formar la imagen completa en una etapa posterior de procesamiento, valiéndose de la información temporal que relaciona la imagen con los datos de orientación. De forma particular este estudio incorpora algoritmos e imágenes de fotogrametría a la lógica de construcción del cubo hiperespectral.

Utilizan un hiperespectrómetro con rango de 400-950 nm con bandas de 10 nm de ancho, toman imágenes a una velocidad de 25 líneas por segundo, llamado fps por comodidad, con resolución espectral de 9 nm, la calibración del equipo se realiza al momento de trabajo una única vez. En lo que se refiere a las características del vuelo, se realiza el mismo a una elevación de 120 m con una velocidad de 6 m/s, cubre un área entre 20000 a 100000 m<sup>2</sup>. Todo esto permite obtener una resolución espectral de 10-50 cm por píxel.

El sistema es controlado por un dispositivo de señales digitales integrado a la cámara y utilizan un microcontrolador raspberry pi para almacenar la información. Algunos de los problemas o conclusiones importantes de resaltar son:

- El control entre la toma de datos del GPS-INS y la toma de imágenes del hiperspectrómetro requirió una sincronización por la diferencia de tiempo entre el envío de la señal y la apertura del diafragma de la cámara. Específicamente la diferencia fue de 83 ms.
- El uso de fotogrametría para apoyar el sistema inexacto GPS-INS permite reducir el peso del equipo aéreo al costo de algoritmos más complejos de construcción de imagen. Permite además la creación de modelos 3D.
- El peso de la información varía según el tipo de datos. Las medidas de GPS-INS, las imágenes de hiperspectrómetro y el registro de tiempo implicaron una carga de 97.5 MB/min, mientras que el paso imágenes aéreas RGB implican una carga de 570 MB/min. Esta conclusión da una idea de los pesos o flujos de información a trabajar.
- En el octocóptero las velocidades de vuelo inferiores a 2m/s provocan que la lógica de vuelo automático desplace el UAV con saltos que introducen movimientos en el cabeceo indeseados. Este efecto se vería disminuido en un sistema con gimbal.
- La capacidad del sistema de control, el cual se encarga de labores de envío de señales, comunicación con el GPS-INS, almacenamiento de información y comunicación Ethernet redujeron la velocidad de captura de líneas a 25 fps de los 160 fps teóricos.

Siguiendo el mismo estilo de trabajo (Malenovský, Veness, & Wallace, 2014) presentan un estudio basado en cámaras tipo push-broom en el cual obtienen información sobre el nivel de la clorofila tipo a y b, el índice de área de hoja y la biomasa verde de varios campos de musgos en zonas antárticas. Se aplica el mismo principio de GPS-INS para la formación de la imagen; no obstante a diferencia de (Suomalainen, y otros, 2014) carecen de imágenes e información fotogramétrica por lo cual requiere de mayor cantidad de equipo y precisión en las mediciones. Algunas de las condiciones o características del estudio son:

- **Cámara:** Sensor Head-wall Photonics Mucro-Hyperspec VNIR. Peso de 960 g con dimensiones de 110 x 93 x 50 mm. Posee una rendija de 25  $\mu\text{m}$  para el paso de la luz, un FOV de 49.8°, distancia focal de 8 mm, 1004 píxeles en su dimensión espacial, un rango de barrido de 361-961 nm con 324 bandas y un framerate de 93 fps con disponibilidad de 115.
- **Características del Vuelo:** Vuelos de 11 m sobre el suelo con octacóptero, velocidades de 2-3 m/s, tiempos de 3 minutos con poca posibilidad de extensión. Esto permite una resolución de 1 cm por píxel, abarcando así 10.2 m en la dimensión espacial del sensor.
- **Control y Sensores:** Para el control de todo el sistema se utiliza el equipo DVR Express Core de The IO Industries, con un peso de 1.5 kg permite el manejo de todos los datos y el control de captura de imágenes. En lo que se refiere a sensorización utiliza dos GPS, una IMU, una antena y un computador para poder recolectar la información, previo inclusive al Express Core. La recolección de información se realiza a una velocidad de 5Hz.
- **Peso de Información:** Por motivos de resolución, framerate y profundidad de píxel el sistema tiene un flujo de datos de 96 MB/s.

De la misma forma que en (Suomalainen, y otros, 2014) destacan la importancia de la sincronización entre captura de imagen y toma de datos, resaltando que un error de 10 ms podía implicar un error de relación espacial de hasta 2 cm bajo las condiciones de su experimento. Finalmente el estudio marca la importancia en la búsqueda de soluciones ligeras, tanto en la toma de información como en el control del sistema, dado que la carga en sensores y equipo provoco el tiempo de vuelo de tres minutos.

En el caso de (Zarco-Tejeda, González-Dugo, & Berni, 2012) se utiliza tecnología push-broom para analizar los niveles de clorofila a y b, el índice de reflexión fotoquímico y la temperatura de corona sobre un campo de naranjas y mandarinas; todo con el objetivo de conocer el estado del estrés del agua en los campos. Varios de los aspectos a resaltar del estudio son:

- **Cámara:** Utiliza la misma cámara que el caso previo, Micro-Hyperspec VNIR de Headwall Photonics; no obstante, en este caso hace uso de solo 260 bandas en un espectro de 400-885 nm, resolución espectral de hasta 1 nm y un tiempo de exposición de 18 ms. Calibran la cámara en un ambiente asilado con el uso de un sistema CSTM-USS-2000C de distribución uniforme de luz.
- **Vuelo:** Uso de UAS ala fija con capacidad de carga de hasta 3 kg y autonomía de 1.5 h, vuelo a 575 m con velocidades de 75 km/h sobre un área de 6000 m<sup>2</sup>. Estas condiciones permiten una resolución de píxel de 40 cm.
- **Control e Información para Construcción de Imagen:** Hacen uso de un sistema de GPS-IMU sincronizado a la cámara espectral para recolectar la información necesaria para la construcción de la imagen. El control se realiza a través de una comunicación por radio que a su vez adquiere y almacena la información de orientación y posición. El sistema de adquisición y almacenamiento de información permite trabajar a 50 fps.

Entre las conclusiones principales del estudio se asegura la resolución de 40 cm como efectiva para los análisis realizados y el hecho de que las resoluciones espectrales bajas introducen más error en forma de ruido que la exactitud y precisión al usar valores tan bajos.

Como último ejemplo se tiene el estudio de (Mitchell, 2016), en el cual se utiliza una cámara push-broom para poder distinguir tipos de arbustos como el sagebrush, sandberg bluegrass, cinder rock, entre otros. Algunas de las condiciones a destacar del experimento son:

- **Cámara:** PIKA II, rango de 400-900 nm con 240 bandas, máxima resolución espectral de 2.1 nm pero en la práctica se utiliza hasta 6.3 nm.
- **Vuelo:** Ala fija Arctus T-16 con capacidad de carga de hasta 8 kg. Vuelos a altura de 1676 m recorriendo áreas con radio de 4 km. Esto permite un tamaño de píxel de 1.1 m.
- **Control y Manejo de Información:** Sistema cerrado P-CAQ del mismo proveedor de la cámara que permite la toma de información de GPS e IMU en conjunto con el control de la cámara. La construcción de la imagen se realiza en el software GeoReg del proveedor.

En su estudio se define que alturas de 350 m dan una resolución de 29 x 29 cm la cual se considera muy pequeña para la aplicación. Aun así el estudio no logra obtener resultados de alta calidad y entre algunas de las recomendaciones menciona planear la ruta de vuelo para reducir la distorsión del píxel y minimizar las diferencias de iluminación entre líneas de vuelo.

Como se observa en los casos presentes se puede ver que la aproximación gimbal con push-broom no es tradicional en el campo de los estudios hiperespectrales mediante UAV, esto no implica un pero en el proyecto sino la ausencia de trasfondo del mismo. Las características de alturas de vuelo, velocidades, framerate, áreas y pesos permiten establecer una guía en cuanto al estándar del equipo utilizado.

Por otro lado el uso de tecnología snapshot en conjunto con UAV es bastante escasa pero el estudio encontrado si utiliza un sistema gimbal para la orientación de las imágenes y marca un antecedente que da luz verde a la combinación de ambos sistemas.

### 3. Desarrollo de la Solución

En este capítulo se hace una explicación progresiva de las labores realizadas para obtener la solución final al problema principal. Esto abarca pero no se limita al estudio de sistemas estabilizadores, el diseño de programas para control de cámaras hiperespectrales, las modificaciones físicas al estabilizador para el acople de los equipos y las pruebas desarrolladas para verificar el funcionamiento.

Al inicio de cada sección se detallara el reto y objetivo de la misma.

#### 3.1. Comprensión de Cámara Hiperespectral

Para empezar el desarrollo de la solución, se investigó sobre los sistemas hiperespectrales, con un énfasis en las cámaras del GII.

Los resultados del estudio sobre funcionamiento se encuentran resumidos en el marco teórico en la sección “Imágenes y Cámara Hiperespectral”. Por otro lado en la tabla 3.1-1 se presentan las características de las distintas cámaras hiperespectrales que posee el GII, la información respecto al modelo UHD-185 proviene de la hoja de datos de la misma (Cubert), mientras que la información de los modelos Push-Broom fue provista por el GII.

Tabla 3.1-1 Características de Cámaras Hiperespectrales. Dada por el personal del GII y (Cubert)

Parámetro	Grande Retiga EXL/EXi Blue	Pequeña UI3360 CP-NIR	UHD-185
Dimensiones (mm)	370 x 120 x 85	210 x 50 x 60	195 x 67 x 60
Peso (g)	2120	680	470
Estilo	Push-Broom	Push-Broom	Snapshot
Resolución Espacial (Píxel)	1392	2048	50 x 50
Resolución Espectral (nm)	-	-	8 @ 532
Intervalo de Muestra (nm)	413-1030	385-950	450-950
Canales o Bandas Espectrales	1040	1088	125
Bits de Digitalización	8	12	12
Flujo de Datos (MB/s)	38	508	9.84
Tiempo de Exposición (ms)	-	83	0.01-1000
Tiempo de Captura de Cubo (s)	1 por línea de captura	0.1 por línea de captura	0.2 por cubo de captura
Resultado de Captura	Folder con todas las imágenes tomadas en una sesión. Cada	Una única imagen que contiene la información	Imagen hiperespectral 50 x 50

	imagen responde a una dimensión espacial en todo el espectro especificado	espectral de una única línea espacial	Imagen en escala de grises 1000 x 1000
--	---	---------------------------------------	--

La cámara Retiga EXL no sería implementada dado que el peso de la misma es incompatible con el UAV a utilizar. La información de la misma se presenta para la comparación y referencia.

En lo que se refiere a los canales de las cámaras Retiga y UI3360 se debe aclarar que algunos canales repiten valores de longitud de onda, contrario al caso del UHD.

Para la elección de un gimbal los valores más importantes a considerar son el peso y las dimensiones físicas de cada cámara, en este sentido se usaron las dimensiones físicas y peso del UI3360 CP para la elección del gimbal.

Los valores de tiempo de captura, cantidad de bits en imagen, flujo de datos o demás son de importancia para el desarrollo del control en secciones próximas. Los valores presentes responden a la información de máximos posibles.

### 3.2. Selección de Gimbal

Para iniciar el desarrollo de la solución del proyecto, la primera labor a realizar fue la selección del Gimbal que portará y estabilizará la cámara hiperespectral; para poder llevar a cabo dicha labor fueron necesarias cuatro actividades:

- Investigación sobre gimbals, para comprender las características y métodos de operación, además se busca obtener parámetros que permitan la selección del mismo.
- Diálogo de las características deseadas en el Gimbal por parte del GII para poder realizar un perfil inicial del modelo a buscar.
- Búsqueda de modelos comerciales o soluciones alternativas tratando de acoplarse a las características deseadas
- Discusión en presencia de las opciones para la selección final del Gimbal

#### 3.2.1. Estudio sobre Gimbal

Inicialmente se realizó una investigación bibliográfica afín al tema de gimbals. Dado que el tema a trabajar se centra en la selección y no en el modelado de una planta gimbal se dio énfasis a la investigación de parámetros, características o criterios que permitieran tanto comprender el funcionamiento de un gimbal, y la terminología que lo rodea, como generar un criterio para la elección cuidadosa de un modelo apto.

La versión detallada de la información obtenida a lo largo de la investigación realizada se encuentran en el marco teórico bajo la sección denominada “Sistema de Estabilización Inercial: Gimbal”. En esta sección se presentan solamente una enumeración de los datos de posible interés.

- **Exactitud Angular**
- **Velocidad de Reacción**
- **Características Físicas**
- **Estilos de Control:** Se perfilan cuatro métodos en los que el gimbal realiza una lógica de estabilización o control.
  - Sobre la LOS
  - Sobre la Orientación
  - Seguimiento de Estructura
  - Primera Persona
- **Instrumentación:** Los gimbals combinan acelerómetros, giroscopios y magnetómetros en una unidad sensorial principal denominada IMU.
- **Tipo de Motor:** Los gimbals pueden utilizar servomotores o motores sin escobillas
- **Cantidad de Ejes**
- **Balance**

### 3.2.2. Requisitos de Gimbal y Perfil de Búsqueda Inicial

Terminada la investigación y para poder detallar los requisitos y necesidades del Gimbal, se realizaron una variedad de preguntas a los supervisores del proyecto. El desarrollo extenso de esta sesión de preguntas y respuestas se presenta en primer apéndice del documento.

Las respuestas a las dudas, en conjunto con la información obtenida en la investigación sobre gimbals permiten establecer un primer perfil de búsqueda. Se presenta a continuación el perfil inicial de búsqueda con la explicación según corresponda

- **Estilo de Control:** Dado que el objetivo es mantener la orientación de la cámara en un único sentido desde el inicio hasta el fin de operaciones se entiende que los gimbals a buscar deben poseer la capacidad de realizar el llamado control por orientación o bien lock mode en el mercado.
- **Capacidad de Carga:** Dado que una de las opciones de cámara a controlar tiene un peso de 470g se buscan gimbals diseñados para cámara con ese peso o mayor. En este sentido los sistemas diseñados para cámaras tipo gopro o similares se pueden descartar ya que su diseño contempla cámaras de 200-300g.
- **Cantidad de Ejes:** Dado que se busca una labor de control de orientación completa, se tiene una preferencia por los gimbal de tres ejes, aun cuando estos presentan un mayor peso. Esto no descalifica la posibilidad de usar gimbal de dos ejes.
- **Tipo de Motor:** No se puede perfilar propiamente el tipo de motor dada la naturaleza opuesta de las características de peso de motor contra calidad de estabilización; no obstante dado que en el mercado predominan los gimbal de motores sin escobillas se tiene un leve sesgo en cuanto al tipo de motor a utilizar.
- **Acople con Cámara:** Si bien no es un requisito dado por el GII, se prefiere un gimbal con conexión a cámara más sencillas y menos específicas del modelo; esto puede reducir problemas posteriores.
- **Acople a UAV:** Mismo criterio que el acople con cámara.

- **Lógica de Balance:** Es necesario que el gimbal sea capaz de contener a la cámara hiperspectral y aun así poder ejecutar la lógica de balanceo adecuadamente, los principales factores para definir esto es que la lógica de balance no dependa de piezas diseñadas específicamente para sostener o contener una cámara tradicional específica, sino que sean partes propias de la estructura o de bases a las cuales se acopla la cámara y que las dimensiones internas del gimbal sean lo suficientemente amplias para poder contener la cámara y permitir que esta se pueda mover en búsqueda del balance.
- **Tamaño:** Se sabe que la dimensión vertical del gimbal tiene un límite de tamaño, aunque este no se conoce a priori.

Entre otras necesidades o requisitos se puede destacar:

- Costo menor a los €1600.
- Peso menor a 5 kg.
- No se busca filtrar y elegir un gimbal de a primera entrada, se busca generar una gama de opciones para después filtrar en conjunto con los demás participantes del proyecto.
- No es necesario preocuparse por la alimentación del gimbal.

### 3.2.3. Búsqueda de Modelos Comerciales

Con el perfil generado se dio paso a la búsqueda de modelos comerciales. Dicha labor consiste en generar una tabla en la cual se toma nota de las características propias del gimbal. Dado que el GII dejó claro que se debía considerar la opción de unir elementos separados, dígame marco estructural, motores, procesador y sensores; la búsqueda se dividió en dos versiones: 1-Sistemas Cerrados Comerciales. 2- Estilos DIY. En conformidad con esto se tienen dos tablas de búsqueda distinta.

A continuación se presentan los criterios de las tablas, posterior a esto se presentan las tablas y finalmente el proceso de selección.

#### 3.2.3.1. Criterios Presentes en la Tabla

Se presenta cada criterio de la búsqueda de sistemas cerrados, en caso de ser necesario se explica el criterio y/o su necesidad. El orden en el que se presenta es el mismo en el que se encuentran en la tabla; dicho orden no implica un grado de importancia y cumple un único motivo de ordenar adecuadamente la información.

- **Peso**
  - Explicación: Máximo en conjunto con cámara de 5 kg.
- **Precio**
- **Peso de Carga**
  - Explicación: Peso máximo de carga de 680 g.
- **Tamaño**
  - Explicación: La altura al ser el único parámetro crítico se presenta siempre de primero.
  - Razón: La altura es el único criterio a contemplar dado que puede interferir con el aterrizaje del dron.
- **Cantidad de Ejes**
- **Tipo de Motor**

- **Estilo de Control**
  - Explicación: Los tipos de control que el gimbal es capaz de realizar. Se utilizan los mismos nombres a como se presentaron en la sección de *Estudio de Gimbal*.
  - Razón: Permite conocer el método de trabajo del gimbal.
- **Exactitud Angular**
  - Explicación: La exactitud en grados que el proveedor asegura con el gimbal. Puede ser interpretado como el error al orientar la cámara.
- **Tiempo de Reacción**
  - Explicación: La velocidad máxima en grados por segundo con la cual el gimbal corrige el error entre la orientación actual de la cámara contra la orientación deseada.
- **Ángulos de Trabajo**
  - Explicación: Es el rango de ángulo en los ejes cabeceo, guiñada y alabeo en los que el gimbal es capaz de efectuar su labor de estabilización correctamente.
  - Razón: A más altos valores mayores son las perturbaciones que puede resistir el gimbal.
- **Energía de Alimentación**
  - Explicación: Información sobre la alimentación que utiliza el gimbal, principalmente tipo de batería recomendada y rango de voltaje.
  - Razón: Es un dato informativo que permite tener noción de los problemas o retos en lo que alimentación se refiere.
- **Controlador**
  - Razón: Un dato más informativo que de selección que entre otras cosas permite conocer cuál es el estándar del mercado.
- **Control RC**
  - Explicación: Denota si el gimbal incluye un control remoto para su control a distancia. Además en caso de ser indispensable el control remoto también hace mención de esto.
- **Capaz de Control RC Personal**
  - Explicación: Especifica si el gimbal se puede controlar a distancia mediante un control remoto. Además indica cualquier particularidad al respecto.
- **Necesidad de Diseño Físico**
  - Explicación: Describe los posibles problemas, la dificultad y las necesidades que involucra acoplar el gimbal tanto a un UAV como a una cámara hiperspectral no convencional. Se encuentran las siguientes categorías:
    - UAV-Bajo: Cuando el gimbal fue diseñado para montaje en drones, implica modificaciones basadas en la diferencia de tamaños entre el dron de diseño y el dron a utilizar pero ahorra trabajo en lógica de amortiguación y similares.
    - UAV-Medio: Cuando el gimbal posee algún método de acople simple con UAV, se entiende por simple un acople que no contempla reducción de vibraciones o dimensiones estándar.
    - UAV-Medio/Alto: Categoría reservada para los gimbals diseñados para filme y por ende a ser sujetados por humanos.
    - UAV-Alto: Categoría utilizada cuando la forma de la estructura del gimbal no es convencional ni con los modelos de filme en tierra o de acople con UAV.
    - Cámara-Medio: Cuando el acople entre la cámara y la sujeción al gimbal implica únicamente una placa intermedia que se acople a ambos elementos.

- Cámara-Medio/Alto: Cuando el acople requiere únicamente de una placa intermedia pero el diseño de dicha placa se complica por la presencia de ciertos elementos en la zona de montaje.
    - Cámara-Alto: Cuando la base del gimbal se diseña específicamente para cámaras DSLR pero la lógica de balanceo sigue siendo general en algunos ejes.
    - Cámara-Muy Alto: Cuando el gimbal contempla una marca específica de cámara por lo que no presenta forma de realizar el balanceo. En estos casos el acople implica no solo una unión física sino también un diseño que permite realizar la labor de balanceo.
  - Razón: Información para considerar el gimbal en vista de las labores de acople futuras.
- **Balanceo**
  - Explicación: Describe brevemente alguna particularidad sobre el proceso de balanceo del gimbal. Además menciona si el proceso de balanceo depende o no de una herramienta.
  - Razón: Entre más sencillo sea el balanceo más rápido es el proceso de puesta en marcha del gimbal.
- **Dimensiones Internas para Balanceo**
  - Explicación: Define el espacio físico, en tres dimensiones, que posee el gimbal para colocar la cámara. Para efectos de generalización se definen dos frases claves que describen la forma en que se deben interpretar los datos:
    - Marco en el cual se coloca la cámara: Define las dimensiones de la zona donde se debe colocar la cámara, no contempla el espacio vacío entre la base para sujetar la cámara y la estructura general que contiene los motores, aun cuando este espacio es viable para el balanceo del gimbal. Como ejemplo puede observarse la figura 3.2-1, proveniente del Gimbal G3-5D de la empresa Align.

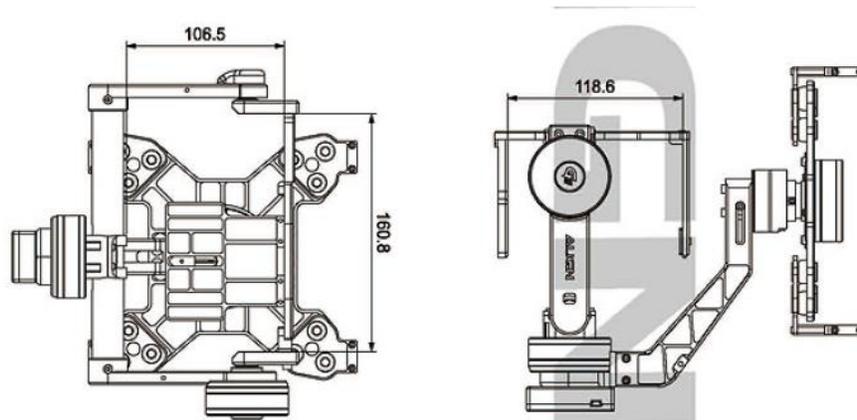


Figura 3.2-1 Dimensiones de Balanceo de Gimbal G3-5D de Align. Extraído de (Align, 2015)

- Distancia máxima del centro de gravedad de cámara a la estructura: Habla de la distancia que hay desde el centro del marco donde se coloca la cámara hasta la estructura del soporte, como ejemplo se puede observar la figura 3.2-2, donde la altura, largo y ancho son 200, 130 y 160 respectivamente.

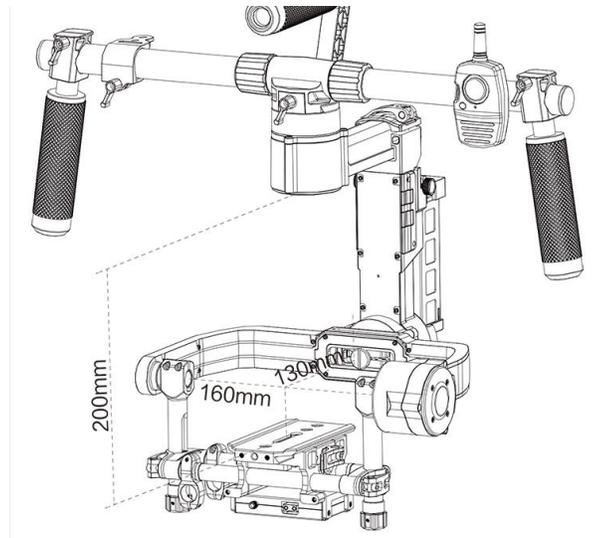


Figura 3.2-2 Dimensiones de Balanceo de Gimbal CAME-Argo. Extraído de (CAME-Tv, 2016)

- Razón: El gimbal no es capaz de realizar su operación de balanceo si la diferencia entre el centro de gravedad de la cámara y la intersección de los ejes es muy amplia. Entre mayores sean los tamaños mayor es el rango de ajuste que se puede realizar.
- Nota: Es necesario aclarar que en los gimbal donde el largo no cumpla la regla dada, se puede considerar la altura ya que la cámara se va a apuntar hacia abajo. Esta forma de cumplir el requisito tiene la desventaja de que la cámara no puede rotar libremente y ciertas rotaciones provocarían una colisión entre cámara y estructura.
- **Soporte Técnico**
  - Razón: Dato informativo que busca aportar información sobre los posibles apoyos al utilizar el equipo.
- **Problemas o Datos Externos**
  - Explicación: Todo problema o dato que no se puede contemplar en las demás categorías.

En lo que concierne a la tabla de la búsqueda del formato DIY se eliminan los criterios de Exactitud Angular, Tiempo de Reacción y Controlador, además se agrega un nuevo criterio:

- **Incluye Motores/Sensores**
  - Explicación: Casilla de si/no donde se dice si el gimbal (estructura) incluye los motores a utilizar, además se adjunta el modelo de motor recomendado. En el caso de los controladores se refiere a si el controlador trae los sensores o IMU necesarios.
  - Razón: Permite conocer si será necesario realizar compras por aparte de los demás elementos y el modelo de motor o sensor específico para el cual se diseñó el gimbal.

### 3.2.3.2. Tablas de Información sobre Modelos de Gimbal

Tanto las casillas en blanco como las casillas marcadas con \* son datos que no se lograron obtener, la diferencia radica que en los datos marcados con \* el proveedor afirmó que desconocía el dato exacto mientras que en los espacios blancos el proveedor no contestó los correos. Se presenta en las tablas 3.2-1, 3.2-2 y 3.2-3 la información de los modelos comerciales investigados.

Tabla 3.2-1 Primera Tabla de Modelos Comerciales de Gimbal

<b>Modelo</b> <i>Compañía</i>	<b>DS1 Beholder</b> <i>ikan</i>	<b>Tarot 5D-TL5D001</b> <i>Tarot</i>	<b>iSEE+</b> <i>Shape</i>	<b>Yelangu G2</b> <i>Yelangu</i>
<b>Peso (kg)</b>	1,1	1,51	1,47	1,6
<b>Precio (€)</b>	711	622 - 871.45	1158	1319-1503
<b>Peso de Carga (kg)</b>	1,67	1,45	0,9	0.8-1.6
<b>Tamaño (mm)</b>	300x 175.26 x 160.02	240 x 246 x 215	295 x 234 x 184	350 x 230 x 400
<b>Cantidad de Ejes</b>	3	3	2	3
<b>Tipo de Motor</b>	Sin Escobillas	Sin Escobillas	Sin Escobillas	Sin Escobillas
<b>Estilo de Control</b>	Orientación Seguimiento	Orientación Primera Persona Seguimiento	Orientación	Orientación Seguimiento Mixto: Orientación en Roll, Seguimiento en Pitch y Yaw
<b>Exactitud Angular (°)</b>	*	Pitch/Roll: ±0.02 Yaw: ±0.03		
<b>Tiempo de Reacción (°/s)</b>	*	Todos los Ejes: 200		
<b>Ángulos de Trabajo (°)</b>	Yaw: ±355 Roll: ±60 Pitch: ±180	Yaw: ±360 (continuo) Roll: ±25 Pitch: -120 a 15	Roll: ±172.5 Pitch: ±60	Yaw: ±170 Pitch: ±90
<b>Energía y Alimentación</b>	3 x 3.7 V Li-Ion 18650	11-26 V Batería LiPo 3s-6s	LP-E6, de proveedor	10-12.6 V Batería LiPo 3s
<b>Controlador</b>	Simple BGC board 32 bit, diseño con intención de reducir tuning	Control PID con placa ZYX de marca propia, software de configuración accesible		
<b>Control RC</b>	No	No.Si lo requiere y si admite el FutabaSBUSReceiver	No	No
<b>Capaz de Controlar RC Personal</b>	Si, mediante simple BGC. No se contempla esta opción en diseño	Si, mediante un canal de Sbus Receiver		
<b>Necesidades de Diseño Físico</b>	UAV: Alto Cámara: Medio	UAV: Bajo Cámara: Muy-Alto	UAV: Medio-Alto Cámara: Medio	UAV: Medio-alto Cámara: Medio
<b>Balaneo</b>	Lógica de balaneo no depende la cámara.  <b>(Sin Herramientas)</b>	Diseño para una cámara, no hay forma de realizar el balaneo.	Balaneo depende de la estructura y no de la cámara a montar  <b>(Con Herramientas)</b>	Balaneo asociado a placa base y estructura, no a la cámara  <b>(Sin Herramientas)</b>
<b>Dimensión Internas para Balaneo (mm)</b>		N/A	Distancias Posibles de Ajuste. Altura: 36 Largo: 140 Ancho: 28 Largo 2: 50 (Mismo eje, otro punto de ajuste)	Marco en el cual se coloca la cámara. Altura: 180 Largo: 100 Ancho: 190
<b>Soporte Técnico</b>	Videos de PID tuning y balaneo  Manuales básicos	Reducido apoyo de comunidad  Manuales detallados	Videos y manuales con explicaciones de uso	Proveedor posee tutorial del balaneo.  Videos varios en youtube
<b>Problemas o Datos Externos</b>		Al no unirse al controlador de un UAV la calidad de control baja. No contestan Correo		No se encuentran manuales. No contestan correos

Tabla 3.2-2 Segunda Tabla de Modelos Comerciales de Gimbal

<b>Modelo</b> <i>Compañía</i>	<b>Defy G2</b> <i>Defy</i>	<b>Brushless Gimbal</b> <i>Axis</i>	<b>CAME-Argo</b> <i>CAME TV</i>	<b>Zenmuse Z15-G</b> <i>DJI</i>
<b>Peso (kg)</b>	0,907	1,79	2,9	1,35
<b>Precio (€)</b>	1100, 685(usado)	623,27	1154	1799
<b>Peso de Carga (kg)</b>	1	2	3	0,534
<b>Tamaño (mm)</b>	200 x 120 x 120	380 x 450 x 220	452 x 288 x 513	230 x 200 x 200
<b>Cantidad de Ejes</b>	3	3	3	3
<b>Tipo de Motor</b>	Sin Escobillas	Sin Escobillas		Servo Motores
<b>Estilo de Control</b>	Orientación Seguimiento	Orientación Seguimiento	Orientación Seguimiento	Orientación Seguimiento
<b>Exactitud Angular (°)</b>	*		Todos los ejes ±0.02	Todos los ejes ±0.05
<b>Tiempo de Reacción (°/s)</b>	*		Yaw: 90 Roll: 30 Pitch: 100	Yaw: 130 Roll: 30 Pitch: 130
<b>Ángulos de Trabajo (°)</b>	Yaw: ±360 (continuo) Roll: ±70 Pitch: ±45		Yaw: ±360 (continuo) Roll: ±25 Pitch: ±360 / 105-190(continuo)	Yaw: ±360 (continuo) Roll: ±40 Pitch: ±360 (continuo)
<b>Energía y Alimentación</b>	Batería LiPo, cargador a 110 V	3s/5s LiPo 8-25 V	4S LiPo 14.8 V	4s-12s LiPo
<b>Controlador</b>	SimpleBGC board 32 bit, diseño con intención de reducir tuning.	Simplebgc board de 32 bits, ningún tuning previo	Simplebgc board de 32 bits, diseño con intención de poco tuning	Procesador no especificado propio de DJI
<b>Control RC</b>	No	Si	Si	No.Si lo requiere y si admite el FutabaSBusReceiver
<b>Capaz de Controlar RC Personal</b>	Si, mediante Simple BGC	Si, mediante simplebgc board	Si, mediante simplebgc board	Si, mediante conexión a Sbus Receiver
<b>Necesidades de Diseño Físico</b>	UAV: Medio-Alto Cámara: Medio	UAV: Medio-Alto Cámara: Medio	UAV: Medio Cámara: Medio	UAV: Bajo Cámara: Medio-Alto
<b>Balaneo</b>	Balaneo asociado a placa base y estructura, no a la cámara <b>(Sin Herramientas)</b>	Balaneo asociado a placa base y estructura, no a la cámara <b>(Con Herramientas)</b>	Balaneo asociado a placa base y estructura, no a la cámara <b>(Sin Herramientas)</b>	Balaneo asociado a placa base y estructura, no a la cámara <b>(Con Herramientas)</b>
<b>Dimensión Internas para Balaneo (mm)</b>	Proveedor no da visto bueno		Distancia máxima del centro de gravedad de cámara a la estructura. Altura: 200 Largo: 130 Ancho: 160	Marco en el cual se coloca la cámara. Altura: 76 Largo: 142
<b>Soporte Técnico</b>	Soporte de balaneo y mantenimiento en red vimeo  Accesible a consultas por correo	Videos generales y apoyo de comunidad	Videos de Comunidad con explicaciones de uso y balaneo	Manuales de usuario  Videos de ayuda en el balaneo  Accesible a consultas por correo
<b>Problemas o Datos Externos</b>	Descontinuado por proveedor, se debe conseguir en reventa			Depende del controlador de la marca el cual no se puede conseguir en tienda del proveedor.

Tabla 3.2-3 Tercera Tabla de Modelos Comerciales de Gimbal

<b>Modelo</b> <i>Compañía</i>	<b>Summer-Air</b> <i>DYS</i>	<b>Ronin-M</b> <i>DJI</i>	<b>Mini 3-Air</b> <i>CAME Tv</i>	<b>G3-5D</b> <i>Align</i>	<b>Defy G5</b> <i>Defy</i>
<b>Peso (kg)</b>	1,38	2,3	1,9	1,25	2,27
<b>Precio (€)</b>	700	1559	1064	1070,85	1500,15
<b>Peso de Carga (kg)</b>	0,85	3,6	1,2	1	3
<b>Tamaño (mm)</b>	270 x 232 x 169	420 x 210 x 500	380 x 490 x 210	236.4 x 236 x 150	370 x 264 x 250
<b>Cantidad de Ejes</b>	3	3	3	3	3
<b>Tipo de Motor</b>	Sin Escobillas	Sin Escobillas	Sin Escobillas	Sin Escobillas	Sin Escobillas
<b>Estilo de Control</b>	Orientación Seguimiento	Orientación unicamente para eje yaw Seguimiento	Orientación Seguimiento Seguimiento de solo yaw	Orientación Seguimiento <i>*Se selecciona por eje</i>	Orientación Seguimiento
<b>Exactitud Angular (°)</b>		Todos los ejes ±0.02		*	*
<b>Tiempo de Reacción (°/s)</b>		Yaw: 90 Roll: 30 Pitch: 100		Todos los ejes en 60	*
<b>Ángulos de Trabajo (°)</b>	Yaw: ±360 (continuo) Roll: ±45 Pitch: ±90	Yaw: ±360 (continuo) Roll: ±25 Pitch: +105/-190	Yaw: ±360 (continuo) Roll: ±180 Pitch: ±360 (continuo)	Yaw: ±360 (continuo) Roll: ±20 / ±30 Pitch: 120 / 90	Yaw: ±360 (continuo) Roll: ±70 Pitch: ±45
<b>Energía y Alimentación</b>	4s 14.8 V	4s Ronin M Battery	3S LiPo 11.1 V	12-16 V Batería LiPo 3s-4s	Batería LiPo, cargador a 110 V
<b>Controlador</b>	Simplebgc board de 32 bits, diseño con intención de que no se realice tuning	Procesador DJI Advanced de 32 bits. Sensor IMU integrado a estructura	Simplebgc board de 32 bits, diseño con intención de poco tuning	Control PD con placa APS-M de los drones de la marca	SimpleBGC board, diseño con intención de reducir tuning.
<b>Control RC</b>	No	Si	Si	No. Si lo requiere y si admite el FutabaSBus-Receiver	No
<b>Capaz de Controlar RC Personal</b>	Si, mediante Simple BGC	Si, mediante conexión a D-Bus	Si, mediante simplebgc board	Si, mediante conexión a Sbus Receiver	Si, mediante Simple BGC
<b>Necesidades de Diseño Físico</b>	UAV: Bajo Cámara: Medio	UAV: Medio-Alto Cámara: Medio	UAV: Bajo Cámara: Medio	UAV: Bajo Cámara: Alto	UAV: Medio-Alto Cámara: Medio
<b>Balaneo</b>	Balaneo asociado a placa base y estructura, no a la cámara  <b>(Híbrido)</b>	Balaneo asociado a placa base y estructura, no a la cámara  <b>(Sin Herramientas)</b>	Balaneo asociado a placa base y estructura, no a la cámara  <b>(Sin Herramientas)</b>	Balaneo en pitch depende de estructura para cámaras Canon 5D  <b>(Con Herramientas)</b>	Balaneo asociado a placa base y estructura, no a la cámara  <b>(Sin Herramientas)</b>
<b>Dimensión Internas para Balaneo (mm)</b>	Distancia máxima del centro de gravedad de cámara a la estructura. Altura: 157 Largo: 75 Ancho: 147	Distancia máxima del centro de gravedad de cámara a la estructura. Altura: 195 Largo: 120 Ancho: 160	Distancia máxima del centro de gravedad de cámara a la estructura. Altura: 170 Largo: 75 Ancho: 147	Marco en el cual se coloca la cámara Altura: 118,6 Largo: 106,5 Ancho: 160,8	Proveedor da visto bueno
<b>Soporte Técnico</b>	Manual de usuario sencillo, videos de ayuda con balance.	Amplio soporte en videos de proveedor Manuales de usuario Accesible a consultas por correo	Videos de Comunidad con explicaciones de uso y balanceo	Manual detallado Apoyo de empresa mediante videos tutoriales. Consultas por correo	Soporte de balanceo y mantenimiento en red vimeo Soporte de comunidad Accesible a consultas por correo
<b>Problemas o Datos Externos</b>	Falta de Información de proveedor y no contesta correo		El gimbal utiliza enconders para mayor precisión en el control de la posición del gimbal	Buena calidad de control depende de controlador UAV de la marca. Bajo rango de ajuste para balanceo (19mm).	Descontinuado por proveedor, se debe conseguir en reventa

La información obtenida proviene tanto de las páginas de internet de los proveedores, los manuales de los gimbals así como de consultas por correo realizadas directamente a los proveedores. Dada la gran cantidad de referencias que implicó este proceso de búsqueda se traslada la cita de las mismas al primer anexo del documento en caso de que se desee observar con mayor detenimiento.

Las tablas 3.2-4 y 3.2-5, que poseen información sobre la aproximación DIY, dividen la información en dos secciones La tabla 3.2-4 para la comparación de marcos estructurales, que contempla la información afín a los motores y la tabla 3.2-5 para la comparación de los controladores, que contempla la información afín a los sensores. Por este motivo ciertos parámetros no aplican a cada tabla.

La falta de modelos DIY que utilicen servomotores se debe a que dichas aplicaciones tienden a depender de computadores como la raspberry pi o arduino en conjunto con lógicas de PID diseñadas desde cero, esta opción aunque viable implica un consumo de tiempo y recursos que no se pueden aportar por temas de plazos de entrega o similares

Tabla 3.2-4 Información de Modelos de Estructuras para Opción DIY

Modelo <i>Compañía</i>	Two dee Evo NG <i>CopterFrames</i>	Tiyaga Minigimbal <i>Minigimbal</i>	DSLR 3-Axis Brushless Gimbal <i>Axis</i>	HIFLY 3-Axis Gimbal Brushless <i>Hifty</i>
Peso (kg)	1500	2000	1300	2000
Precio (€)	Frame: 575 Motores (3): 160.31	Frame: 578.9 Motores(1): 53.43	Frame: 263.70 Kit Con Motores: 587.48	Todo: 711.61
Peso de Carga (kg)	2	1,1	2.5/2 (kit)	2,5
Tamaño (mm)	260 x 200 x 210		380 x 450 x 220	350 x 520 x 300
Cantidad de Ejes	3	3	3	3
Tipo de Motor	Sin Escobillas	Sin Escobillas	Sin Escobillas	Sin Escobillas
Estilo de Control	N/A	N/A	N/A	N/A
Ángulos de Trabajo (°)	Yaw: 360 (continuo) Roll: 150 Pitch: 360 (continuo)		*	*
Incluye Motores/ Sensores	No, compatible con 5208-180T	No, compatible con DYS BGM4108 130T	Si (el kit)	Si, trae tres motores 5208-200T
Energía y Alimentación	N/A	N/A	N/A	N/A
Control RC	N/A	N/A	N/A	N/A
Capaz de Controlar RC Personal	N/A	N/A	N/A	N/A
Necesidades de Diseño Físico	UAV: Bajo Cámara: Medio	UAV: Alto Cámara: Alto	UAV: Medio-Alto Cámara: Medio	UAV: Medio-Alto Cámara: Medio
Balaneo	Contempla lógica de balanceo que no depende de la cámara  (Con Herramientas)	Lógica de balanceo no depende de cámara, le falta placa que facilite el balanceo en pitch  (Híbrido)	Lógica de balanceo no depende de cámara.  (Con Herramientas)	Lógica de balanceo no depende de cámara.  (Con Herramientas)
Dimensión Internas para Balanceo (mm)	Marco en el cual se coloca la cámara. Altura: 225 Largo: 110 Ancho: 158		Marco en el cual se coloca la cámara. Altura: 150 Largo: 200 Ancho: 180	*
SopORTE Técnico	Compañía, contestan correos	Tutoriales varios por parte del proveedor	Contestan correos	
Problemas o Datos Externos		Ensamble de cero por usuario con posibles extras que aumenten el precio. Comercialmente quedan pocos modelos	Desconocimiento de motores apropiados	

Tabla 3.2-5 Información de Modelos de Controladores para Opción DIY

<b>Modelo</b> <i>Compañía</i>	<b>SimpleBGC Alexmos</b> <b>32 bit</b> <i>Basecam</i>	<b>VRGimbal</b> <i>VirtualRobotix</i>	<b>STorM32</b> <i>STorM32</i>
<b>Peso (kg)</b>	22	24	10
<b>Precio (€)</b>	178,12	133,6	22,2
<b>Peso de Carga (kg)</b>	N/A	N/A	N/A
<b>Tamaño (mm)</b>	50 x 50	13 x 51 x 70.5	50 x 50
<b>Cantidad de Ejes</b>	3	3	3
<b>Tipo de Motor</b>	Sin Escobillas	Sin Escobillas	Sin Escobillas
<b>Estilo de Control</b>	Orientación Seguimiento		Orientación Seguimiento
<b>Ángulos de Trabajo (°)</b>	N/A	N/A	N/A
<b>Incluye Motores/ Sensores</b>	Si, propio de la marca pero compatible con otros	Si, MPU 6050 con HMC5883 (magnetómetro)	Si, MPU 6050
<b>Energía y Alimentación</b>	LiPo 3s 11V	LiPo 3s 11V	LiPo 3s-6s 9-25V
<b>Control RC</b>	No	No	No
<b>Capaz de Controlar RC Personal</b>	Si, por PWM o Futaba Sbus, analógico. Control de 3 ejes y un botón de cambio de modo	Si, cuatro pines para manejo de motores	Si, por PWM o Futaba Sbus, analógico. Control de 3 ejes
<b>Necesidades de Diseño Físico</b>	Requiere de una caja para cubrir tanto el sensor como el controlador	Requiere de una caja para cubrir tanto el sensor como el controlador	Requiere de una caja para cubrir tanto el sensor como el controlador
<b>Balanceo</b>	N/A	N/A	N/A
<b>Dimensión Internas para Balanceo (mm)</b>	N/A	N/A	N/A
<b>Soporte Técnico</b>	Comunidad muy amplia ya que se usa mucho en marcas comerciales. Manuales oficiales		Wiki con tutoriales detallados
<b>Problemas o Datos Externos</b>		Considerablemente nuevo y sin aporte de comunidad	

### 3.2.4. Discusión y Selección de Modelo Gimbal

Como se explicó en secciones previas, el proceso de selección se centró en una reunión, específicamente el día 20 de octubre, donde se discutió en conjunto con los encargados del proyecto los distintos gimbal.

En lo que se refiere a los criterios de elección no se definió un orden rígido de prioridad pero si se terminaron de establecer la extensión en la importancia de ciertos parámetros:

- Precio: Presupuesto final de €1600 se debía de respetar pero no tenía mayor incidencia. Es decir un modelo de €700 no tiene ventaja contra uno de €1200 si su único punto a favor es el precio.
- Angulo de Trabajo: El dron a utilizar no se moverá a altas velocidades, esto a su vez implica que el UAV no dará giros de cabeceo o alabeo mayores de los 25-35°. En este sentido el ángulo de trabajo del gimbal en los ejes cabeceo y alabeo no requiere ser mucho mayor a estos valores; por otro lado se busca que el gimbal si sea capaz de trabajar el eje guiñada en un rango de 360° continuos.
- La conexión por control remoto es necesaria para poder operar el gimbal a distancia.
- Cantidad de Ejes: Se definió que el gimbal debía poder realizar el estilo de control por orientación lo cual exige que el mismo sea de tres ejes.
- Capacidad de control: Es intrínsecamente necesario que se le pueda exigir al gimbal la posición a la cual estabilizar la cámara, esto en visión del barrido que requiere la cámara tipo push-broom. Todos los gimbal con controlador simpleBGC son capaces de esta labor así como los de las opciones 2, 8 10 y 12.
- Diseño Físico: Inicialmente se entiende que el gimbal requerirá de modificaciones para acoplarlo al UAV o viceversa, pero en el momento de la reunión se concluyó que sería preferible si el gimbal a elegir ya ahorre trabajo en este sentido. La unión con la cámara si mantiene la misma prioridad que al principio.
- Exactitud Angular y Velocidad de Reacción: Se prescindió del dato por las siguientes razones. 1) Los pocos modelos que presentan el dato dan valores muy similares cercanos a los 0.02°. 2) Los valores que presentan los proveedores asumen que se utilizaran las cámaras y lentes especificados en los manuales y que no se modificarán los parámetros PID de control; ambas condiciones no son realistas con el proyecto, no solo por el hecho de que los gimbals no fueron diseñados para el hiperspectrómetro.

Inicialmente la selección se basó en reducir los trece modelos comerciales a tres finalistas. Específicamente los modelos Summer-Air de DYS, el CAME Mini 3 Air de CAME Tv y el G3-5D de Align respectivamente, recopilaban de forma general las mejores características. Para explicar esta conclusión se puede observar los defectos de los demás modelos así como las conclusiones de algunas características:

- En general todos los gimbals poseen ángulos de trabajo sobre el eje alabeo que les permiten estabilizar adecuadamente la cámara.
- DS1 Beholder: la forma poco convencional implica una gran labor de acople; además el controlador simpleBGC no es de fácil acceso lo cual dificulta la modificación de los parámetros sin tener que destruir la carcasa; por último el desconocimiento sobre las dimensiones internas lo vuelven una opción riesgosa.

- Tarot 5D-TL5D001: un diseño específico para cámaras Canon 5D implica una labor riesgosamente compleja de acople con el hiperspectrómetro. Otras opciones presentan un peso y capacidades similares sin este defecto.
- Yelangu G2: el gimbal es incapaz de rotar 360° continuos en su eje guiñada, además la falta de respuesta del proveedor provoco el desconocimiento del controlador, así como la posibilidad de controlar el gimbal por control remoto, el desconocimiento de ambos factores críticos llevo a su eliminación.
- Defy G2: al dar las dimensiones del hiperspectrómetro al proveedor, este confirmo que no sería posible realizar el balanceo con el Defy G2. Por esta observación se desechó el modelo aun cuando este era el más liviano de todos.
- Zenmuse Z15-G: el precio del mismo se salió del presupuesto definitivo y sus características las presentan otros modelos como el 7 y 11. Por otro lado el peso de carga solo podría sujetar la UHD-185 y se prefiere la capacidad de poder equilibrar ambas cámaras.
- CAME-Argo: aun siendo el gimbal que posee las mejores dimensiones internas para balanceo, el peso del CAME-Argo es excesivo acercándose a los 3 kg. Además es el modelo con mayor altura llegando a los 452 mm, es decir casi medio metro, lo cual presenta el riesgo de estorbar al sistema de aterrizaje del dron.
- iSEE+: Descartado por ser sistema de dos ejes.
- Brushless Gimbal de Axis: falta de datos por parte del proveedor.

De los gimbals restantes se tiene tanto el Ronin-M como el Defy G5 los cuales no presentan problemas o razones por las cuales descartarlos a priori, en el caso de ambos modelos la decisión de no usarlos se basó en la superioridad de las tres otras opciones en ciertos aspectos:

1. Tanto el Ronin-M como el Defy G5 son más pesados que las tres opciones finales.
2. El Ronin-M asegura un control sobre el eje alabeo de  $\pm 25^\circ$ , que aunque sea suficiente se encuentra en el borde inferior de lo aceptable según este requisito.
3. El Defy G5 solo controla la posición de su eje cabeceo en  $\pm 45^\circ$ , si bien este problema no es crítico y se puede solucionar con un acople más complejo del hiperspectrómetro, las tres opciones finalistas presentan capacidades similares sin tener esta necesidad extra.
4. El Ronin-M es el más grande de los gimbals restantes y presenta complicaciones similares, más no tan drásticas como el CAME-Argo.
5. Tanto el Ronin-M como el Defy G5 fueron diseñados para el filme en tierra y la creciente preocupación sobre la adaptación entre el gimbal y el dron terminó de descartarlos como opciones.

Por otro lado las tres opciones finales: G3-5D, Summer-Air y CAME Mini 3 Air, cumplen las necesidades del perfil de búsqueda y los criterios del GII, además de forma particular cuentan con acople para drones, lo que facilita labores futuras.

En este punto fue necesario comparar la opción comercial con los modelos DIY. Dada la información presente en las tablas 3.2-4 y 3.2-5, el contraste se resumió en los siguientes puntos:

- El mayor punto positivo de la opción DIY se encuentra en el precio del mismo el cual es menor que las opciones finalistas de sistemas cerrados. El precio mínimo sería de €609.68 con la tercera estructura y el tercer controlador.

- El manejo de múltiples pedidos presenta el riesgo de atraso o complicaciones a la hora de la compra.
- La compra del controlador directo al proveedor implica que no existe ninguna configuración previa del PID sobre la cual partir. Aunque los valores originales de PID no sean los exactos para el hiperspectrómetro su existencia si permitiría reducir el tiempo de ajuste.
- La mayoría de las estructuras DIY fueron diseñadas para uso en tierra y no por dron, esto limita las opciones a únicamente la primera opción.
- El apoyo del proveedor sobre la puesta en marcha del equipo sería inexistente, solo ofrecerían soporte sobre su producto y no en el funcionamiento en conjunto de ambos.

Descartados los modelos DIY, se procedió a realizar la decisión entre los tres modelos finalistas.

Primeramente se eliminó el gimbal G3-5D por su diseño de operar en conjunto con el controlador del UAV propio de la empresa Align, y aunque sea un sistema independiente parte de sus capacidades de procesamiento se pierden al independizarlo del UAV.

Para la elección final entre el Summer-Air y el Came Mini 3 Air se definió que ambos gimbals cumplían todos los requisitos necesarios, y que su principal diferencia se centraba en que el Summer-Air era 600g más liviano y el Came Mini 3 posee encoders que aseguran una mejor estabilización. Tras deliberación se afirmó que el UAV podría asumir el exceso de peso en vista a una mayor seguridad y exactitud en la toma de las imágenes.

De esta forma se eligió el gimbal CAME Mini 3 Air para el desarrollo de las secciones posteriores del proyecto.

### 3.3. Configuración de Cámara Hiperspectral UI3360-CP para Operación en Trípode Nexstar

En visión a la futura puesta en marcha de la cámara hiperspectral tipo push-broom, se decidió unir la cámara UI3360 CP a un trípode para observación de constelaciones marca Nexstar. La unión se basó en generar una interfaz que pudiera controlar la cámara hiperspectral y el trípode de tal forma que se pudiera realizar la captura de un cubo de datos.

Entre otras razones por las cuales se toma la decisión de realizar este procedimiento se encuentra:

- El hiperspectrómetro con cámara UI3360-CP, tipo push-broom, carecía de sistema de barrido para su funcionamiento en tierra.
- Parte de los problemas a afrontar es la comunicación, control y adquisición de imágenes de las cámaras hiperspectrales.
- Como se menciona al final del marco teórico, la toma de imágenes con cámara tipo push-broom presenta el problema de requerir un sistema de barrido, la labor a realizar permite una primera aproximación a dicho problema.

Si bien el objetivo del proyecto busca la implementación de la cámara con un estabilizador inercial en aire, se consideró válido realizar una aproximación inicial en tierra para poder observar el

comportamiento y necesidades de la cámara. Se presenta a continuación una descripción más detallada de lo solicitado por el GII en conjunto con las distintas etapas del desarrollo de la solución.

### 3.3.1. Requisitos de Interfaz

El objetivo de la interfaz fue controlar la cámara y la labor de barrido como tal. A groso modo se busca crear una interfaz equivalente a la usada para controlar el barrido de la cámara Retiga, en la figura 3.3-1 se puede observar la misma

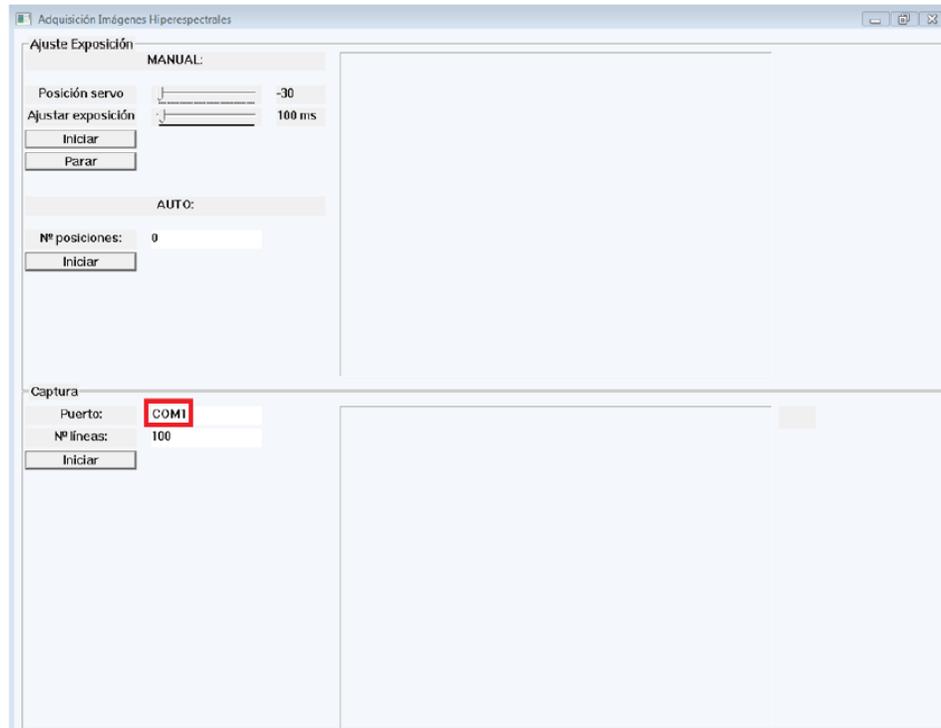


Figura 3.3-1 Interfaz para Control de Cámara Retiga. Extraído de (Priego Torres, 2011)

Específicamente la interfaz debe ser capaz de:

- Asignar un tiempo de exposición específico a la cámara
- Poder capturar y presentar una única imagen de la cámara para que el usuario sea capaz de verificar si el tiempo de exposición es el adecuado
- Asignar la cantidad de líneas específicas a tomar, esto se traduce directamente en la cantidad de imágenes a tomar
- Asignar el ángulo de barrido deseado
- Almacenar todas las imágenes del barrido en una carpeta que contenga la información de la fecha y hora en que se inició el barrido
- Usar formato de imagen .png o .tiff en la medida de lo posible
- Definir e implementar los mínimos y máximos de los dispositivos en cuanto al tiempo de exposición, la cantidad de líneas y el ángulo de barrido
- Almacenar varios datos del proceso en un archivo .txt al final de la operación.

Se da la libertad del ambiente sobre el cual se programará la interfaz. Además no hay limitación en la forma en que el usuario ingresa los datos y trabaja con la interfaz, ya sea el uso de cajas de texto, botones de elección de opciones, entre otros.

### 3.3.2. Comunicación y Operación de Cámara UI 3360-CP

Inicialmente se buscó comprender la forma en que se puede operar la cámara, donde operar se refiere al uso a través de algún lenguaje de programación y no mediante una interfaz propia del proveedor.

Los primeros intentos de comunicación y control de la cámara se realizaron mediante las distintas herramientas del software Matlab, específicamente la “*Image Acquisition Toolbox*” y la “*Webcam Toolbox*”. Con ambas librerías se obtuvieron resultados positivos para las necesidades 1, 2, 4 y 5; no obstante ambos métodos eran incapaces de asignar un tiempo de exposición específico a la cámara, cabe destacar que se podía asignar tiempo de exposición siempre y cuando se respetase la función:

$$T_{Expo} = \left(\frac{1}{2}\right)^n [s] \quad (3-1)$$

Donde n es un número entero menor a -4. La incapacidad de controlar adecuadamente el tiempo de exposición, eliminó la opción de utilizar Matlab como medio de programación de la interfaz y la cámara.

Seguidamente se consultó el manual en línea de la cámara dado por el proveedor de forma gratuita, en busca de opciones directas del proveedor. Mediante esta búsqueda se encontraron bibliotecas de C++, C# y Delphi que permitían el control de la cámara en aplicaciones de diseño propio, en conjunto con los drivers necesarios para controlar la cámara y la interfaz propia del proveedor.

Sin afán a ser exhaustivos, se explica la forma de operar de las bibliotecas. A ojos del programador la comunicación se realiza mediante el uso constante de un número de identificación propio de la cámara. Todos los métodos contienen dentro de sus propiedades dicho número de identificación de cámara lo que le permite al programador saber a cuál cámara le envía el comando, en caso de usar múltiples cámaras. Todas las funciones retornan un valor numérico que sirve a manera de código de error en caso de que el método no se ejecutara adecuadamente (iDS, 2016).

La biblioteca permite conocer y modificar todas las propiedades de la cámara, de la imagen a capturar y de los posibles formatos de la misma. Se presenta a continuación una lista con algunas de las labores que permite realizar la biblioteca:

- Inicializar la cámara
- Seleccionar el estilo de captura: esto se refiere a captura por trigger en programación, video continuo y demás.
- Asignar y liberar memoria: la cámara requiere un espacio de la RAM de la computadora en la cual almacena la imagen en el momento de captura, la biblioteca presenta métodos para asignar y retornar dicho espacio de memoria.
- Selección de formato: la biblioteca permite definir si la captura se realiza en escala de grises o escala RGB, además define la cantidad de bits por píxel entre 8, 10, 12 y 16.
- Conocer los valores máximos y mínimos asignables de framerate y tiempo de exposición.

- Asignar framerate
- Asignar tiempo de exposición
- Captura de imágenes o video
- Guardado de imagen: se puede definir el formato entre png, jpg y bitmap; además el nombre con el que se guarda la imagen es configurable así como la dirección en la que se guarda, partiendo del hecho que la dirección por defecto es la misma en la que se encuentre el ejecutable de la aplicación
- Desconexión de la cámara

Cabe destacar que la biblioteca da la versatilidad entre detener el programa hasta que se termine la captura o permitir que el mismo siga su ejecución en lo que se realiza la captura, lo que asegura adecuadamente la necesidad cinco.

Por otro lado existe una relación inversa entre los valores que se le pueden asignar al tiempo de exposición y al framerate, a menor framerate mayor es el tiempo de exposición máximo que la cámara puede adquirir, el tiempo mínimo es constante en todo momento. Por esta razón, el usar la biblioteca implica que el valor de framerate debe ser alterable en la interfaz.

Dada la gran versatilidad y el alto nivel de control de parámetros, se decidió utilizar la biblioteca del proveedor para la comunicación con la cámara, específicamente se utiliza la versión de C++ por la familiaridad con el lenguaje sobre el C# y Delphi. Esto a su vez define que la interfaz se realizaría con Windows forms de C++.

### 3.3.3. Operación con Tripode Nexstar

Seguidamente se decidió entender la forma de operación del trípode nexstar. El trípode fue diseñado para la observación de estrellas y constelaciones, por lo que se desconocía si podría cumplir con los requisitos para la operación de barrido. Cabe aclarar que se sabía a priori que la comunicación era mediante puerto serial por lo que la comunicación con el trípode no limitaba el lenguaje en el cual se programaría la interfaz.

La lectura de los manuales del trípode así como un período de pruebas permitieron reconocer e implementar una variedad de comandos, entre estos se menciona los que serían útiles para la lógica de barrido:

1. Solicitar la posición del trípode
2. Desplazar el trípode a una posición exacta, definida por el ángulo de rotación de los dos ejes
3. Iniciar la rotación continua sobre un eje a una velocidad específica
4. Detener la rotación continua sobre un eje
5. Reconocer si el desplazamiento dado por el comando dos ha terminado

Como se mencionó la comunicación se da por medio serial; en los casos donde un comando admitía datos variables la serie de bytes contemplan un espacio designado. Como ejemplo se puede observar la figura 3.3-2, donde se presenta una parte del manual de trípode nexstar y la sintaxis del comando "Slew" (Celestron, 2008), tercer comando según la enumeración previa.

Command Function	PC Command	Hand Control Response	Applies to Versions
Variable rate Azm (or RA) slew in positive direction	"P" & chr(3) & chr(16) & chr(6) & chr(trackRateHigh) & chr(trackRateLow) & chr(0) & chr(0)	"#"	1.6+
Variable rate Azm (or RA) slew in negative direction	"P" & chr(3) & chr(16) & chr(7) & chr(trackRateHigh) & chr(trackRateLow) & chr(0) & chr(0)	"#"	1.6+

Figura 3.3-2 Ejemplos de comandos de Trípode Nexstar. Extraído de (Celestron, 2008)

Para ambos casos la secuencia implica enviar el carácter "P", o bien el byte 80 dado el código ASCII del carácter P, luego se envían los bytes 3, 16 y 6 o 7, este último siendo la diferencia entre los dos comandos presentes; los bytes de trackRateHigh y trackRateLow almacenan la información de la velocidad a la cual se desea que se mueva el trípode. Finalmente se envían dos bytes de valor cero y un byte de cierre con el valor 35 o "#" según el código ASCII. El byte de cierre es una especificación que se da al inicio del manual.

En conjunto con los comandos y sus formas de representación serial, se descubrieron una variedad de particularidades y posibles problemas en cuanto a la operación del trípode:

- Las operaciones de rotación del trípode parten de la existencia de un ángulo 0° inicial. Es decir si en el comando de rotación se le provee por ángulo 50° este valor depende de la orientación que considere el 0° inicial.
- El ángulo 0° inicial se puede definir de dos formas: Mediante una operación de alineación o bien el valor defecto que provee el trípode la cual se basa en que "La rotación en la que se encuentran los ejes al momento de encender el trípode marca el 0° inicial".
- La operación de alineación no se puede realizar mediante el comando serial, esto porque requiere el uso del control manual que trae el trípode; además es necesario realizar ese proceso cada vez que se conecta el trípode a la alimentación.
- El desplazamiento del trípode mediante él envió de una posición exacta le toma al trípode como mínimo seis segundos. Esto ya que el trípode no realiza un único movimiento rápido sino que realiza un movimiento brusco que corrige y afina en el lapso dado, visualmente su comportamiento es similar a un control PID.
- Todo comando que envíe información del trípode al computador, como los comandos uno y cinco necesitan como máximo 3.5s para asegurar él envió de la información.
- Según el manual de usuario, si se solicita información múltiples veces a trípode, por ejemplo utilizando los comandos uno y cinco sin respetar el tiempo de 3.5s, existe la posibilidad de que el trípode no retorne ninguna información.

- No se encontraron velocidades máximas y mínimas oficiales para el comando tres. En su defecto se conoce un rango de velocidades válidas dado que se utilizan como ejemplos a lo largo de manual, con seguridad se sabe que el trípode puede rotar a velocidades entre 0.042 y 3 °/s

Todo en conjunto deja claro que es posible controlar el movimiento del trípode pero que se debe aceptar una variedad de limitantes en su operación, limitaciones que tendrían repercusión sobre las posibles lógicas de barrido.

### 3.3.4. Lógica de Barrido

Al comentar las experiencias con la cámara Retiga Exi se conocieron los dos métodos de lógica de barrido: movimiento de a pasos o movimiento continuo.

Movimiento de a pasos significa que el trípode se mueve a la primera posición para que luego la cámara tome una foto, seguidamente el trípode se mueve a la segunda posición para que la cámara tome la segunda foto y así sucesivamente hasta obtener la cantidad de imágenes necesarias; esta opción requiere conocer la cantidad de imágenes y el ángulo de barrido para definir así la distancia angular entre imagen e imagen. Movimiento continuo implica que el trípode empieza una rotación continua a una velocidad tal que la cámara sea capaz de capturar todas las imágenes necesarias en el ángulo de barrido solicitado; esta opción debe contemplar el tiempo de captura, así como el ángulo y cantidad de capturas para definir la velocidad a utilizar. En la figura 3.3-3 se puede observar un diagrama de flujo de ambos procesos.

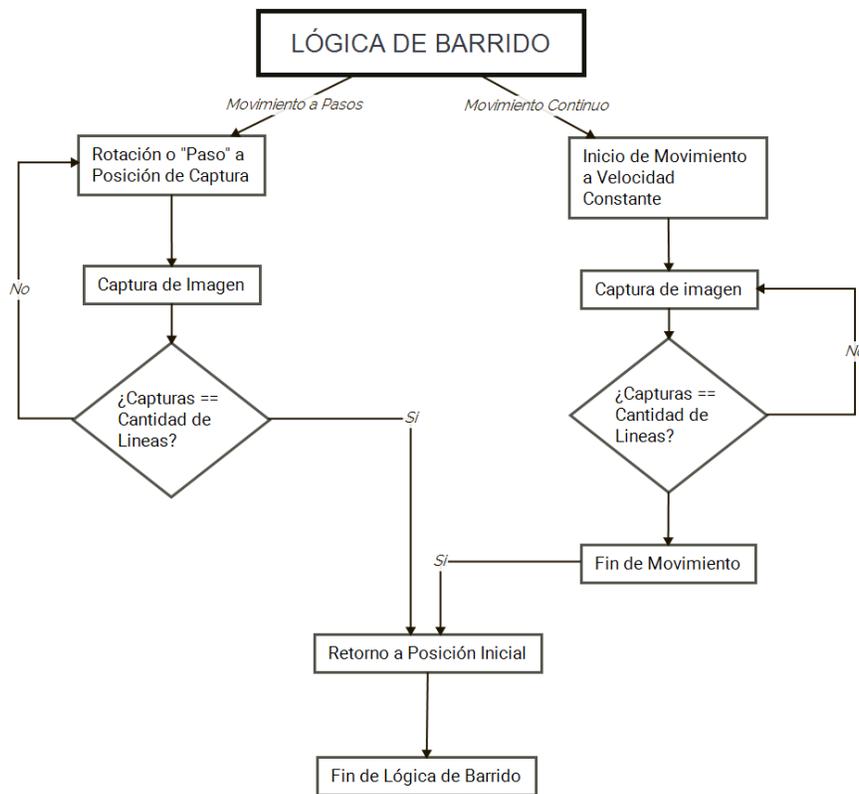


Figura 3.3-3 Diagrama de Flujo para la Lógica de Barrido. Desarrollado en X Mind 7

Ante las opciones se consideró las limitantes del trípode nexstar.

1. Cada paso en el trípode se traduce en un movimiento de seis segundos y no un movimiento rápido, considerando que un cubo de datos podría constar de 300 líneas esto se traduce a media hora solo para el paso sin contar el tiempo de captura y almacenado.
2. Debe de existir alguna forma de detectar el final del movimiento del trípode, inicialmente se podría utilizar el comando cinco; no obstante la limitante de espera de 3.5s vuelve el proceso de comunicación muy lento.

Por otro lado el trípode presenta una función de uso simple para realizar un movimiento de rotación constante y la única limitante, el rango de velocidades, no se consideró crítica ya que la experiencia con la cámara Retiga daba a entender que las velocidades no se saldrían de los valores mencionados.

Seleccionado el barrido por movimiento continuo, se definió una ecuación para el cálculo de la velocidad de barrido:

$$V[^\circ/s] = \frac{\text{angulo } [^\circ]}{T_{\text{captura}}[s] * C_{\text{Lineas}}} \quad (3-2)$$

Donde  $C_{\text{Lineas}}$  es la cantidad de líneas a capturar y  $T_{\text{captura}}$  es el tiempo de captura de una imagen individual.

### 3.3.5. Interfaz Implementada

Con la lógica de barrido seleccionada, conocido el método de operación de la cámara UI3360 CP y comprendido el método de operación del trípode se diseñó la interfaz gráfica que se aprecia en la figura 3.3-4.

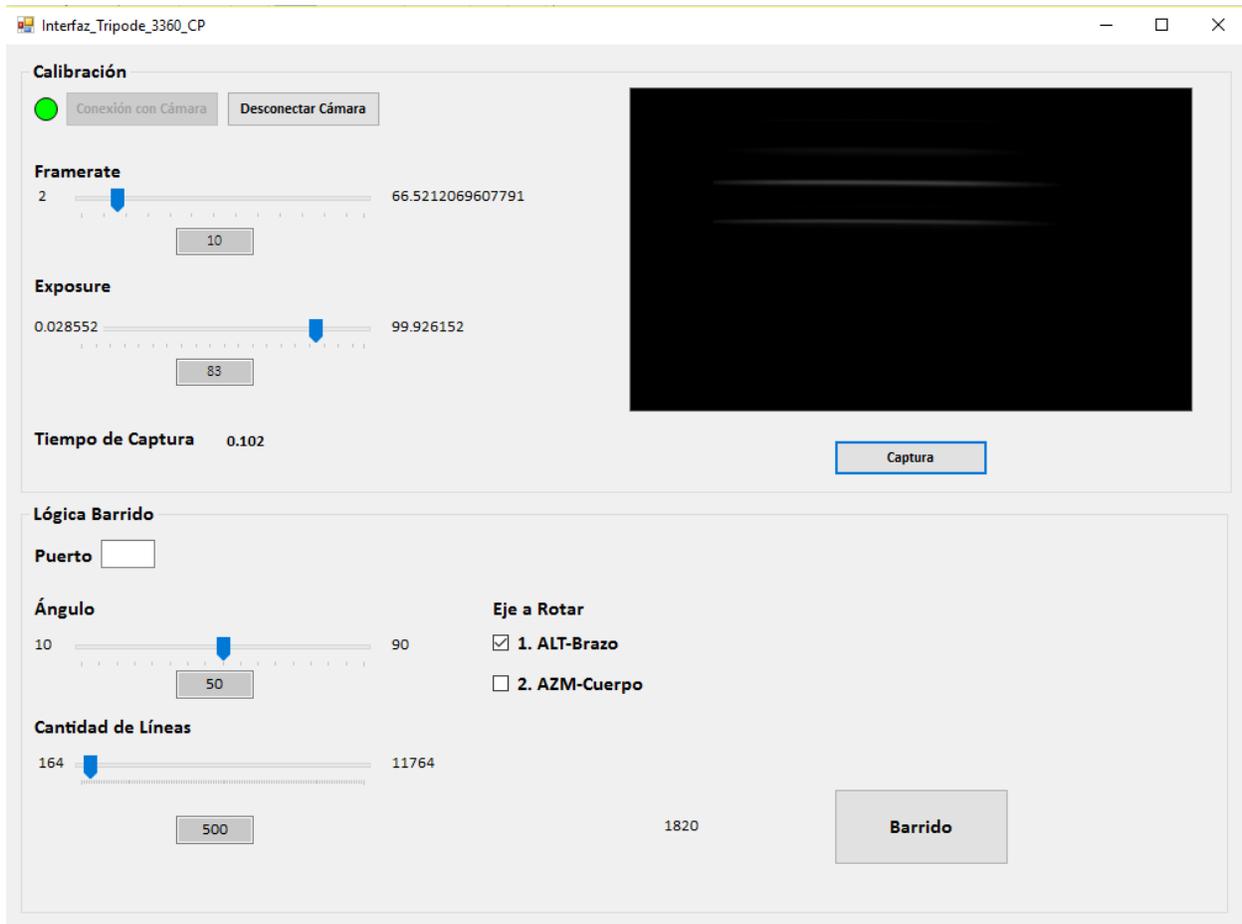


Figura 3.3-4 Interfaz de Control Hiperespectrometro UI3360 CP con Tripode Nexstar. Desarrollado en Microsoft Visual Studio Express 2012

Inicialmente se destaca que la interfaz se divide en dos secciones, la primera se denomina “Calibración” y le provee al usuario todos los valores y medios con los cuales calibrar la captura de imagen de la cámara; por otro lado la segunda sección se denomina “Lógica Barrido” y le provee al usuario las opciones necesarias para poder realizar el barrido. Se explican los distintos elementos de la interfaz:

- **Conexión con Cámara:** El botón se encarga de inicializar la cámara. La labor de inicializar consiste no solo en establecer comunicación con la cámara sino también en asignar el modo de captura al estilo por software, asignar un espacio de memoria RAM para las imágenes de prueba y definir que las imágenes a capturar serán en escala de grises con profundidad de píxel de 16 bits. Por otro lado el botón de conexión también habilita los elementos de modificación de framerate, tiempo de exposición y los botones de captura y desconexión.
- **Desconexión Cámara:** Botón que realiza la operación inversa al de conexión con cámara. También deshabilita el botón de barrido.
- **Framerate:** La sección constan de una caja de texto y un slider que permiten elegir el valor del framerate a utilizar, los valores mínimos y máximos del framerate se toman directamente de la cámara al inicializarla. La elección se puede realizar con la caja de texto o el slider, el elemento que no se utilice se actualiza al momento de cambio para mantener coherencia en la interfaz.

- **Exposure:** Consta de los mismos elementos y lógica de coherencia que la sección de framerate, los valores de mínimo y máximo iniciales se toman directamente de la cámara al inicializarla; además, dado que el framerate afecta directamente el tiempo de exposición máximo, este valor se actualiza cada vez que se da un cambio en los elementos del framerate.
- **Tiempo de Captura:** Elemento de interfaz informativo, presenta el tiempo que le toma a la cámara capturar y almacenar en memoria RAM una imagen. Su valor se actualiza cada vez que se presiona el botón captura.
- **Captura:** Botón que permite tomar una única imagen de prueba la cual se guarda con el nombre "test1.png". Mide el tiempo de captura para presentarlo en la interfaz y una vez tomada la imagen se la presenta al usuario para su revisión. Finalmente habilita el botón de barrido, dado que ya se tiene un valor de tiempo de captura con el cual medir la velocidad de barrido.
- **Puerto:** Espacio para la escritura del puerto COM al cual se ha conectado el trípode. Se debe escribir toda la sentencia es decir "COM #" en vez de solo "#".
- **Ángulo:** Sección para elegir el ángulo de barrido a realizar, cumple la misma coherencia de interfaz que los elementos de elección de framerate y tiempo de exposición. Por defecto empieza en un valor de 50 grados
- **Eje a Rotar:** Permite al usuario elegir sobre cuál de los dos ejes del trípode se va a realizar el barrido.
- **Cantidad de Líneas:** Permite al usuario elegir la cantidad de líneas que quiera tomar; los límites se actualizan cada vez que se cambia el valor del ángulo de barrido o cuando se define el tiempo de captura, es decir cuando se presiona el botón de captura. Los valores de máximo y mínimo se ven limitados por el valor de velocidad mínima y máxima del trípode según la ecuación:

$$CLineas_{max/min} = \frac{\text{ángulo}[^{\circ}]}{T_{Captura}[s] * Velocidad_{max/min}[^{\circ}/s]} \quad (3-3)$$

- **Barrido:** Botón que da inicio a la secuencia de barrido, al final de la operación del mismo se almacena la información de la sesión de barrido en un archivo de tipo txt de nombre "DATOS". La información almacenada abarca: Fecha, ángulo barrido, cantidad de líneas, tiempo de captura, velocidad de barrido, tiempo de exposición, ganancia y las imágenes que fallaron. Se adjuntó un archivo .txt de nombre "README" al ejecutable final que explica la sintaxis del archivo DATOS en caso de que la información sea necesaria en futuras aplicaciones.

Cabe destacar que los botones de conexión con cámara, captura y barrido contemplan una variedad de mensajes de error en caso de que suceda algún fallo. Por ejemplo si el barrido consta de 50 líneas de las cuales tres no se capturan adecuadamente, la interfaz notificará al usuario al finalizar el barrido advirtiéndolo o los números de imágenes que no salieron adecuadamente; otro ejemplo es el que se aprecia en la figura 3.3-5 donde se despliega el mensaje de error "Error al Iniciar la Cámara" cuando falla dicha operación, sea porque la cámara no se encuentra conectada o bien cualquier otra razón.

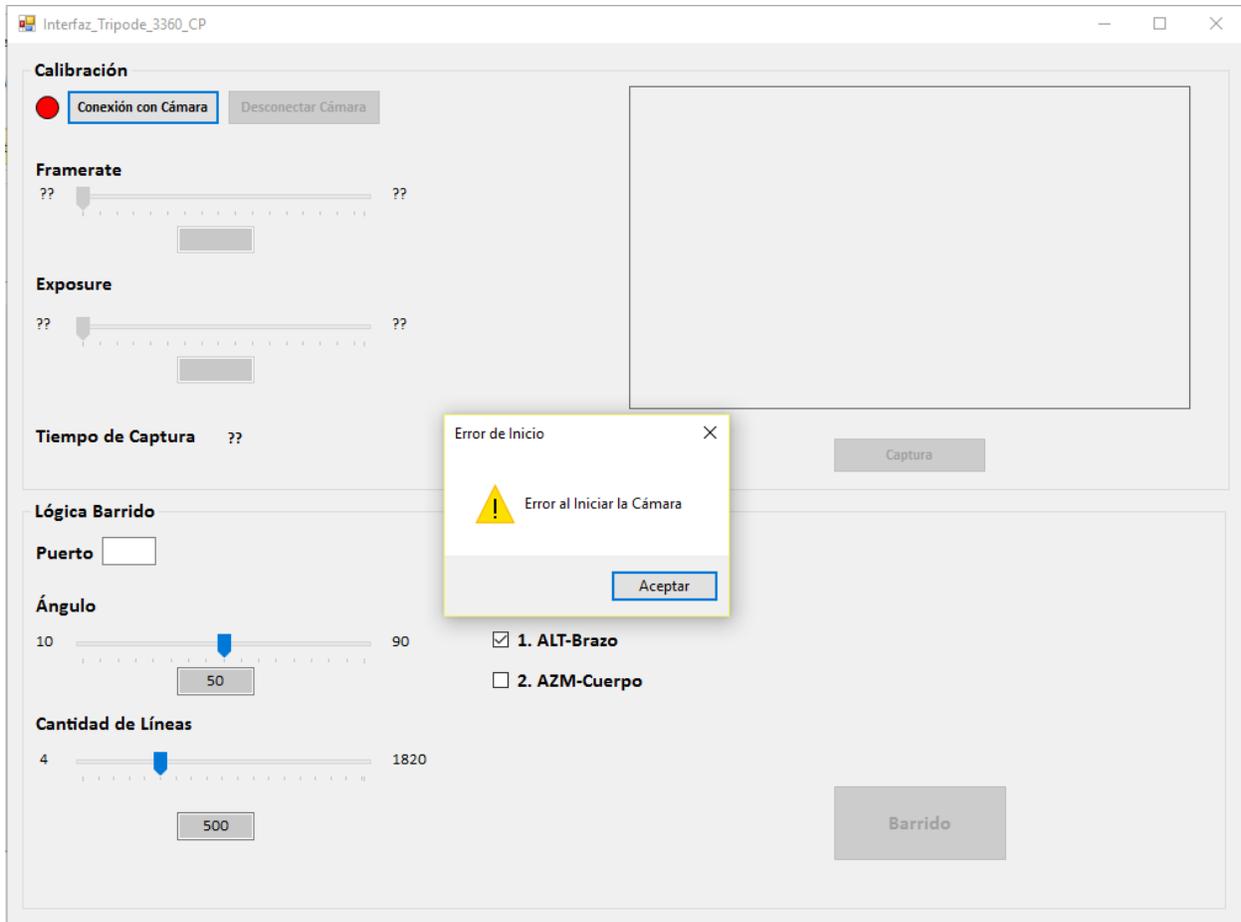


Figura 3.3-5 Interfaz con Error al Inicializar la Cámara. Desarrollado en Microsoft Visual Studio Express 2012

Las imágenes del barrido se guardan en una carpeta compuesta de la forma “Día\_Mes\_Hora\_Minutos\_Segundos\_Año” y bajo el nombre “Image#” donde # se sustituye por el número de imagen respectiva.

### 3.3.6. Resultados de Interfaz

Para poder reconstruir el cubo de datos, y observar así el resultado de la interfaz, se utilizó un script .m de Matlab provisto por el GII que se encarga de la reconstrucción de la imagen y la organización de la misma en una matriz de tres dimensiones.

Haciendo uso de la operación de barrido de la interfaz desarrollada y el script .m mencionado se obtiene como resultado la figura 3.3-6, que presenta una banda espectral determinada de todo el cubo de datos.

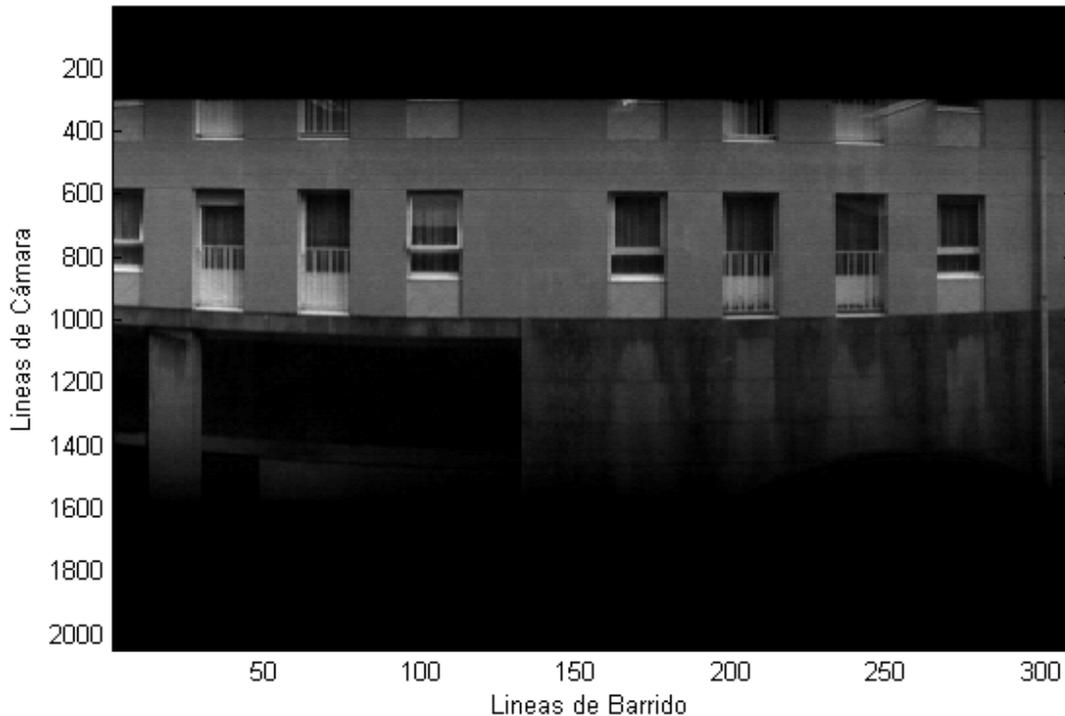


Figura 3.3-6 Resultado de Barrido de Interfaz. Obtenido de Script de Matlab proveído por el GII

La imagen consta de 310 líneas de las cuales ninguna falló, barre un ángulo de  $50^\circ$ , se capturo el nueve de noviembre a las 12:52 pm en un día soleado; el tiempo de captura de cada línea fue de 0.1 s lo que provoca que la velocidad de barrido fuera de  $1.61^\circ/\text{s}$ .

Los distintos encargados del GII dieron el visto bueno de la captura realizada confirmando así el éxito tanto del control de la cámara pushbroom como de la lógica de barrido implementada y la interfaz de control.

### 3.4. Control de Cámara Hiperespectral UHD-185 Tipo Snapshot

En conjunto con la cámara tipo push-broom, el GII cuenta con una cámara hiperespectral del tipo snapshot la cual puede llegar a utilizarse en el proyecto SIMMAP; por este motivo se decidió entender la forma en que opera dicha cámara así como programar una aplicación capaz de controlar la misma, permitiendo así mayor flexibilidad en su uso.

Cabe destacar que en esta sección se busca destacar las particularidades de cómo opera la cámara y no las características generales o validez de los datos capturados.

### 3.4.1. Características de la Cámara y Requisitos de su Control

La cámara UHD-185 es un hiperspectrómetro tipo snapshot fabricado por la empresa Cubert, para su operación los fabricantes enviaron no solo la cámara sino una variedad de equipo y elementos para su operación, entre estos destacan:

- Mini-PC Fitlet-iA10: Computador con sistema operativo Windows 7 que contiene el software necesario para utilizar la cámara, sirve para realizar labores a distancia y cuenta con la capacidad de crear su propia señal wifi para comunicación con computadores personales.
- Elementos de conexión: Cables tipo Ethernet, conversores Ethernet a USB y similares, necesarios para comunicar la cámara con un laptop o la Mini-PC
- Laptop Lenovo Thinkpad: Contiene el software de control a distancia necesario para conectar con la cámara.
- Manual de Operación: Explica tanto la interfaz y software propios del proveedor como los pasos para establecer comunicación mediante aplicaciones propias.

En lo que se refiere a la interfaz de operación, y sin afán a ser exhaustivos en su explicación, la cámara utiliza un software denominado C-Link para establecer comunicaciones con la cámara y una interfaz de nombre Cube-Pilot que permite la captura, calibración y visualización de las imágenes así como su información hiperspectral de forma gráfica. En conjunto los programas C-Link y Cube-Pilot pueden ser controlados a distancia mediante comunicación TCP/IP, razón por la cual la mini PC genera su propia señal wifi. En la figura 3.4-1 se puede observar una captura de la interfaz Cube-Pilot.

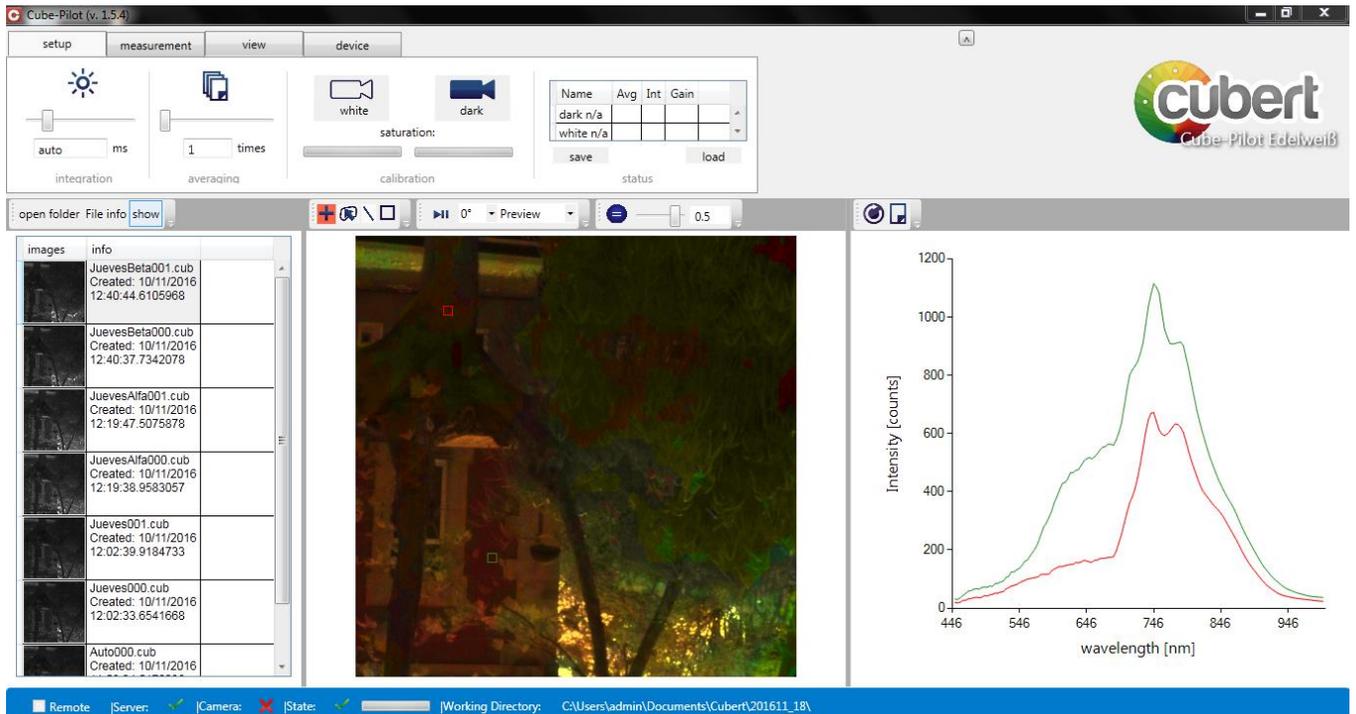


Figura 3.4-1 Interfaz Cube-Pilot. Extraído del Programa Cube Pilot

Las distintas funcionalidades mencionadas se pueden apreciar en la figura 3.4-1, por excepción de la captura de imágenes que pertenece a la ventanilla denominada “measurements”. Finalmente cabe destacar que la información hiperspectral se almacena en un documento de extensión “.cub” propio de

la empresa, pero se puede exportar a una serie de imágenes .tiff, una por banda espectral, para su análisis en programas externos.

Dado que la interfaz es funcionalmente adecuada para la captura de imágenes en tierra, el enfoque del control requerido por el GII no se basa en una aplicación con interfaz de usuario formal sino en controlar la cámara sin usar la interfaz; es decir mediante alguna aplicación propia; esto en miras de poder tener un programa de captura que se pueda utilizar a distancia para al momento de vuelo. Cabe destacar que acceder a la interfaz mediante el Mini-PC si bien viable, no es práctico porque requiere de la presencia de un usuario que dé la orden de captura constantemente, se busca autonomía en la realización de capturas constantes.

El control de la cámara no implica traducir todas las operaciones de la interfaz pero si se busca poder:

- Tomar imágenes de calibración
- Captura de cubos de datos
- Asignar tiempo de exposición y ganancia
- Exportar información a imágenes tipo Tiff

Cabe destacar que el lenguaje de programación bajo el cual se realiza la comunicación y pruebas quedo a libertad; por temas de coherencia interna del proyecto, familiaridad y uso eficiente del tiempo se decidió utilizar el lenguaje C++.

### 3.4.2. Método de Comunicación y Control Mediante Comandos

Para poder llevar a cabo el control solicitado se tornó a consultar el manual “Cubert Util Edel Weiss Software and Installation Guide” incluido con la cámara. En el mismo se encontró no solo el cómo comunicar una aplicación propia con la cámara sino también una lista de comandos básicos para la operación de la cámara.

Primero se explica lo referente al método de comunicación con la cámara. Como se mencionó previamente el programa C-Link, encargado de establecer una comunicación con la cámara, así como el programa Cube-Pilot se pueden controlar mediante protocolo TCP/IP; no obstante la lectura del manual dejo claro que dicha conexión TCP/IP puede provenir tanto de una instancia cliente del programa Cube-Pilot como de cualquier otro programa que conozca el protocolo (Cubert). Por extensión esto implica que el reto pasa a ser el protocolo TCP/IP.

Para poder implementar el protocolo se utilizó la biblioteca winsock2, diseñada por y para Microsoft Visual Studio, la biblioteca contempla funciones para buscar posibles comunicaciones según una IP dada, generar un socket para el protocolo, definir un puerto de comunicación y operaciones de envío y recepción de caracteres. La IP a la cual contactar y el puerto específico por utilizar se detallan en el manual del UHD-185 con 192.168.10.100 y 3000 respectivamente. Esta biblioteca presentó la solución al tema de comunicaciones.

Resuelto el tema de las comunicaciones con la aplicación C-Link y Cube-Pilot, se explica la forma en que se controla propiamente la cámara. El manual de usuario explica que el control de la cámara se

basa en enviar mensajes de caracteres a través de la comunicación TCP/IP. Específicamente el formato es similar a la estructura XML y responden a la forma:

$$\langle Cmd \rangle Comando = "Parámetro" \langle \backslash Cmd \rangle$$

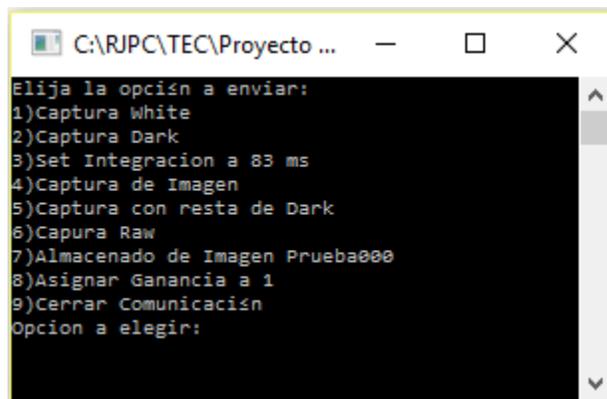
Comando se reemplaza por la función dada y el valor de “parámetro” se puede o no obviar según la función específica, los demás elementos son necesarios por temas de formato para que el programa y el hiperspectrómetro puedan realizar la interpretación y ejecución de los mismo. Entre los comandos implementados se encuentra:

- CapDark: Comando para calibración de valor 0.
- CapRef: Comando para calibración de imagen blanca.
- SetInt: Comando para asignar el tiempo de exposición, el valor se especifica en el campo de parámetro.
- CapCube/CapDS/CapRaw: Comandos de captura de cubo espectral, el estilo “Cube” aplica dos correcciones de las calibraciones de valor 0 y calibración de imagen blanca, el estilo “DS” solo aplica la corrección por calibración de valor 0 y el estilo “Raw” no aplica correcciones a los datos.
- SetGan: Comando para asignar ganancias.
- ExportTiff: Comando para exportar el cubo de datos “.cub” a las imágenes .tiff para su análisis por software externo.

Los comandos implementados permiten cumplir adecuadamente los requisitos del control establecidos.

### 3.4.3. Interfaz de Pruebas para Control de Hiperspectrómetro UHD-185

En la figura 3.4-2, se presenta la interfaz diseñada para las pruebas del control propio.



```
C:\RIPC\TEC\Proyecto ... - □ X
Elija la opción a enviar:
1)Captura White
2)Captura Dark
3)Set Integración a 83 ms
4)Captura de Imagen
5)Captura con resta de Dark
6)Capura Raw
7)Almacenado de Imagen Prueba000
8)Asignar Ganancia a 1
9)Cerrar Comunicación
Opcion a elegir:
```

Figura 3.4-2 Interfaz de pruebas de control de Hiperspectrómetro UHD-185. Desarrollado en Microsoft Visual Studio Express 2012

Dado que el objetivo constaba únicamente en realizar pruebas del control por aplicaciones propias, se optó por utilizar una interfaz simple de aplicación de consola; además, los comandos que asignan tiempo de integración o ganancia asignan valores fijos y no ingresados por el usuario ya que esta segunda opción no añade problemas al control de la cámara sino al manejo de strings en c++.

La aplicación de pruebas se utilizó tanto en computadores ajenas al equipo como en el mini-pc propio de la cámara, en el primer caso es necesario conectar el computador a la red wifi del mini-pc, nombrada "Cubert", en el segundo caso solo se debe ejecutar la aplicación. La interfaz dio resultados positivos donde todos los comandos mencionados respondían igual a como lo desarrolla la interfaz Cube-Pilot del proveedor. Es necesario aclarar que para que el programa funcione se requiere que el programa C-Link del proveedor esté en funcionamiento.

### 3.5. Programación en Matlab para Manejo de Información Hiperespectral

Al estar en capacidad de controlar ambas cámaras, se definió en una reunión con los distintos encargados del proyecto la necesidad de poder comparar la información de las cámaras push-broom y snapshot; esto por temas de elección de cámara, verificación de las imágenes capturadas, comprensión de la información entregada y posibles quejas o consultas que se deberían realizar a los proveedores de la cámara snapshot.

Para cumplir esta labor se programaron una serie de scripts .m de Matlab, se describe a continuación la labor de los scripts realizados así como los resultados provenientes de estos. Cabe destacar que la labor a realizar se basa en el manejo de la información y su representación para análisis, más no el análisis de la información como tal.

#### 3.5.1. Generación de Cubo de Datos del Hiperespectrómetro Snapshot

Para poder comparar la información hiperespectral de ambas cámaras se debe tener acceso al cubo de datos de la cámara snapshot en la misma forma que se posee el cubo de la cámara push-broom. Como se ha mencionado en secciones previas, el GII cuenta con un script .m el cual genera un cubo de datos a partir de las imágenes de una cámara tipo push-broom.

Como primera labor se replicó este procedimiento pero para las imágenes exportadas de la cámara snapshot. Para esta modificación se debe considerar solo las diferencias entre las imágenes que ambas cámaras entregan, esto ya que el procedimiento de ordenar y generar el archivo .mat es el mismo.

Las imágenes tipo push-broom contemplan una dimensión espacial y una espectral donde cada imagen representa un elemento de la dimensión espacial faltante; por otro lado, las imágenes snapshot contemplan las dos dimensiones espaciales y cada imagen representa un elemento de la dimensión espectral faltante. A nivel matricial se traduce a que cada imagen contiene información en dos dimensiones pero debe almacenarse de forma ordenada en la dimensión faltante, espacial para la pushbroom y espectral para la snapshot.

Para aclarar posibles dudas del proceso, en la figura 3.5-1 se presenta un diagrama de flujo del script desarrollado:

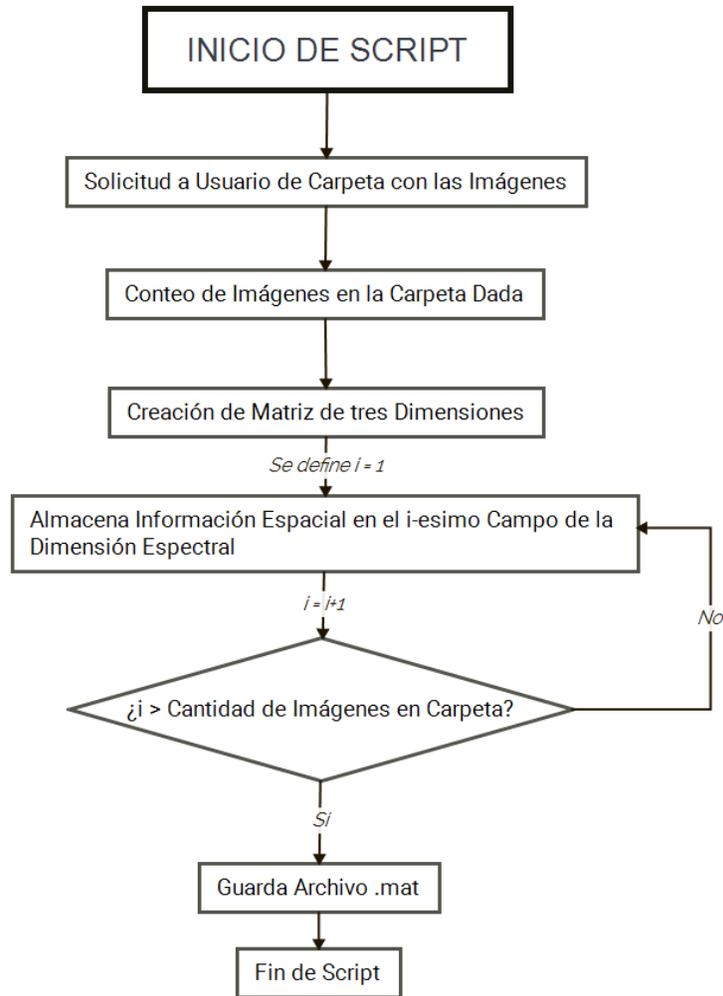


Figura 3.5-1 Diagrama de Flujo de Script para Generar Cubo de Datos. Desarrollado en X Mind 7

El script se denomina “leerCarpetaYGuardarCuboSnapshot” y como se aprecia en la figura realiza ciertas operaciones adicionales a las explicadas previamente; no obstante, estas se reciclan del script original del GII. El resultado del script es un fichero .mat para ser cargado y utilizado en matlab.

A manera de prueba se puede usar el comando `imagesc(cuboSnapshot(:, :, i))` de Matlab para observar la imagen de la i-esima banda, se presenta en la figura 3.5-2 una imagen de la banda 998 nm proveniente de un archivo .mat generado con el script implementado.

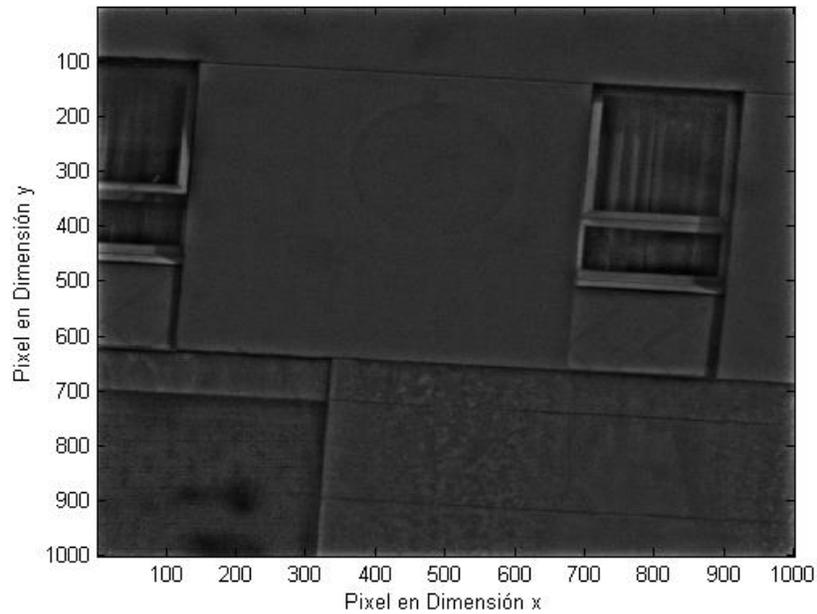


Figura 3.5-2 Banda 998nm capturada por Snapshot UHD-185. Extraída de script de Matlab desarrollado.

### 3.5.2. Comparación Gráfica de Información Hiperespectral entre Cámara Push-broom y Snapshot

La forma básica en que se puede comparar la información es con una gráfica que presenta la información espectral de un área espacial específica. En general la lógica de graficar responde a los siguientes pasos:

1. Selección de un área común en las imágenes push-broom y snapshot: Esto queda a cargo del usuario el cual se asegura de elegir zonas similares en ambas imágenes.
2. Normalizar la Información para su representación: Se busca comparar los perfiles de las señales y no su magnitud, ya que esta depende más de como cuantifica la intensidad lumínica cada cámara.
3. Grafica con los datos espectrales del área común: Con esto se puede observar si los perfiles son similares, así como problemas y similares.

En la figura 3.5-3 se presenta un diagrama de flujo donde se detalla el proceso llevado a cabo por el script programado, seguidamente se explican las secciones.

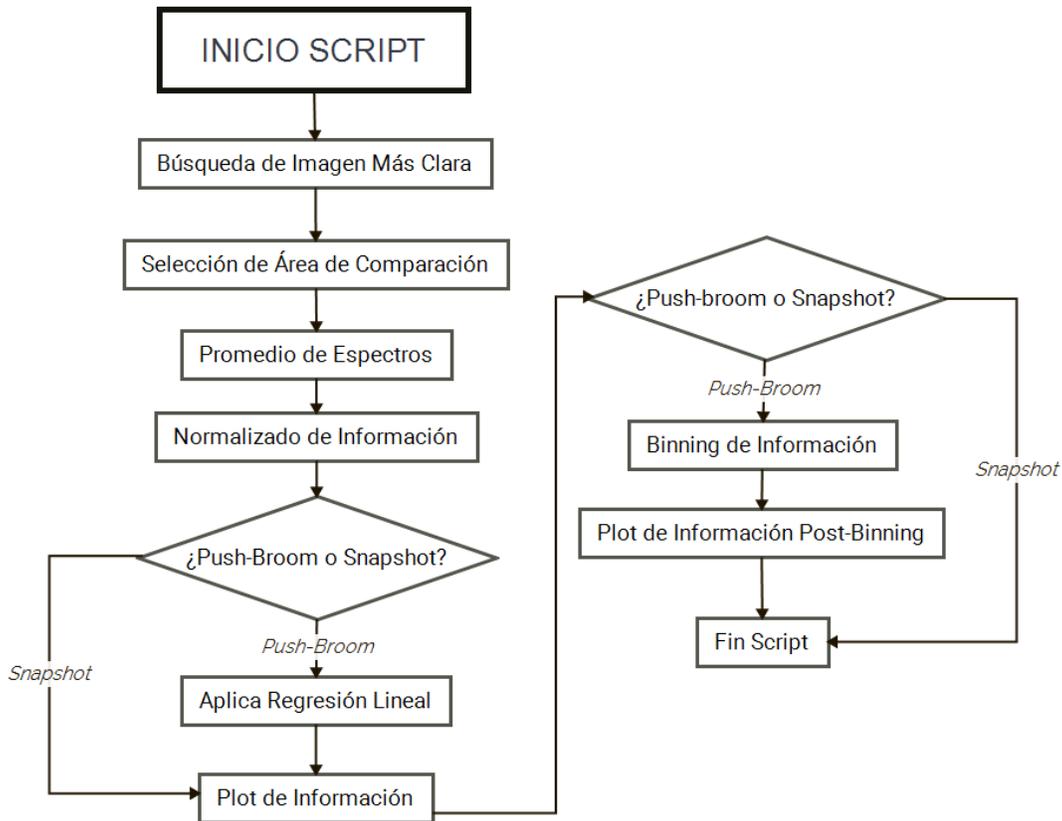


Figura 3.5-3 Diagrama de Flujo Completo de Script de Matlab para Comparación de Información Hiperespectral. Desarrollado en X Mind 7

Cabe destacar que el script asume que los archivos .mat que contienen los cubos de datos de la cámara push-broom y snapshot se encuentran cargados en el espacio de trabajo de Matlab. Se procede a explicar las distintas partes del script.

### Búsqueda de Imagen Más Clara

Los cubos de datos se pueden dividir en “capas espaciales”, es decir imágenes en escala de gris que constan de dos dimensiones espaciales pero donde la magnitud de cada píxel responde a un espectro de onda específico; dichas capas se pueden visualizar en pantalla como es el caso de la figura 3.5-2. La cantidad de capas espaciales es igual a la cantidad de bandas espectrales en el cubo de datos.

Dado que el usuario debe elegir la zona donde se comparará la información hiperespectral, se debe proveer una imagen sobre la cual el usuario realice la selección; para este efecto el primer paso del script es buscar la capa espacial más adecuada para esta selección. La lógica de dicho proceso se puede observar en el diagrama de flujo de la figura 3.5-4.

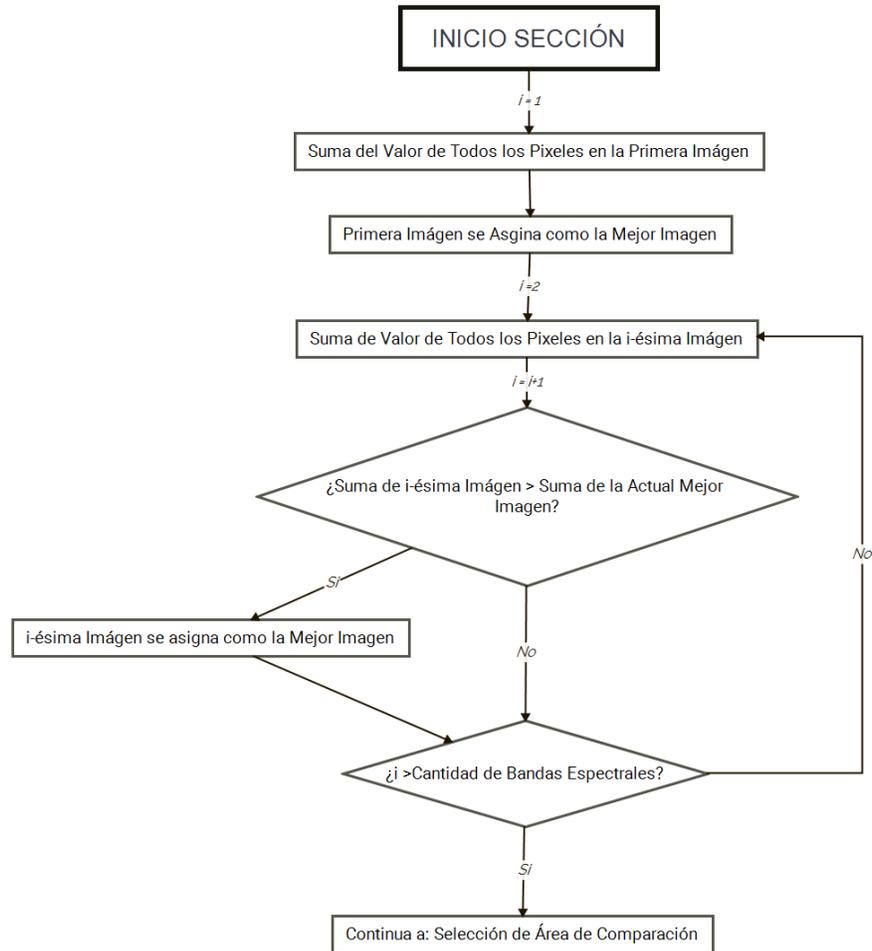


Figura 3.5-4 Diagrama de Flujo de Sección "Busqueda de Imagen Más Clara". Desarrollado en X Mind 7

Como se observa la búsqueda de la capa espacial más adecuada se basa en la suma de todos los valores de los píxeles; si bien este proceso no es perfecto, ya que puede elegir imágenes de poco contraste con tonos muy blancos, se puede aceptar como una aproximación adecuada en luz de los resultados finales del script.

### Selección de Área de Comparación

La selección del área de comparación se realiza mediante el método imcrop de Matlab.

### Promedio de Espectros

El promedio de espectros se basa en utilizar el método mean de Matlab, junto con los resultados de la función imcrop. Esto permite obtener un único valor por espectro, lo que se traduce en una matriz 1xn donde n es la cantidad de espectros.

### Normalizado de Información

Previo a la presentación de la gráfica se normaliza la información, esto en visión de que la escala o sensibilidad de los hiperspectrómetros es diferente por lo cual las magnitudes recibidas podrían variar

entre uno y otro. Además de que para el análisis que se desea hacer en el GII lo que importa es poder comparar el perfil de la gráfica. El método utilizado para normalizar los datos es:

$$Dato_{Normalizado} = \frac{Dato_{original} - \text{Mínimo}}{\text{Máximo} - \text{Mínimo}} \quad (3-4)$$

### Regresión Lineal

El proceso de la regresión lineal se basó en pasar todos los datos del eje horizontal X de las imágenes push-broom, es decir los valores de 385 a 950 nm, a través de una función que ajusta los valores para que se adecuen a los de la cámara snapshot.

Para poder obtener la función de regresión se utilizó el método fit de Matlab. El método fit permite generar una ecuación que se ajuste a la relación dada entre dos matrices de puntos, en este caso la primera matriz responde a los puntos x de la cámara snapshot y la segunda matriz responde a los puntos x de la cámara pushbroom.

Para definir las parejas de puntos a comparar se busca que los valores x describan ciertos valores “y” característicos de sus gráficas respectivas. Por ejemplo la cúspide donde  $y = 1$  en la figura 8.5-7 para la gráfica de push-broom se alcanza en el  $x = 762.9$ , mientras que en la snapshot se alcanza en  $x = 742$ . La tabla 3.5-1 presenta la información de varios puntos utilizados en la generación de la recta de mejor ajuste.

Tabla 3.5-1 Datos para Regresión Lineal

Zona de Gráfica	Valor X Snapshot	Valor X Pushbroom
Cúspide más alta, Máximo Y = 1	742	762.9
Segunda Cúspide Más Alta	778	788.9
Valle entre Cúspides	766	774.3
Inicio de Incremento hacia el Máximo Y = 1	562	579.4
Punto Medio Entre Bajada de la Segunda Cúspide al Final	846	861

Conocidos los puntos se obtuvo la ecuación mediante la ecuación fit:

$$X_{Snapshot} = 1.015 * X_{PushBroom} - 25.67 \quad (3-5)$$

Se seleccionó un método lineal ya que las pruebas con ecuaciones de mayor grado se obtuvieron ecuaciones con igual o menor correspondencia. La ecuación se aplica a todos los valores entre 385 y 950 nm previo al plot de la información.

### Binning de Información

El proceso de binning consiste en reducir la información del push-broom a la misma cantidad de la cámara snapshot. Esto quiere decir reducir los 1088 valores de la push-broom a los 138 que se obtienen en la snapshot, para esto se aplica un promedio convirtiendo cada ocho elementos de la matriz normalizada  $1 \times n$  en un único valor. Se elige específicamente cada ocho elementos ya que es el entero más cercano entre la relación de los 1088 valores a los 138:

$$\frac{\text{Cantidad de Valores PushBroom}}{\text{Cantidad de Valores Snapshot}} = \frac{1088}{138} = 7.88$$

Por temas de coherencia se aplica el mismo proceso de promedio al rango de longitudes de onda, de 385 a 950nm, contra las que se grafica el valor de la matriz normalizada. Cabe destacar que el binning se aplica a las longitudes que pasaron por la regresión lineal manteniendo así la coherencia dentro de la gráfica.

### **Plot de Información**

Finalmente se gráfica los valores de la matriz normalizada contra el rango de espectros que manejan cada hiperespectrómetro. En el caso de la cámara push-broom se grafica la matriz normalizada contra el rango de 385-950 nm y en el caso de la cámara snapshot se grafica contra el rango de 450-998nm.

Cabe destacar que el proceso descrito por el diagrama de flujo de la figura 8.5-3 se realiza dos veces a lo largo del script .m. La primera vez para el cubo de datos de la cámara push-broom y la segunda para el cubo de datos de la cámara snapshot.

Se presenta en las figuras 3.5-5, 3.5-6 y 3.5-7 los resultados del script con cubos de datos capturados el día jueves 10 de noviembre en condiciones de luz solar abundante.



*Figura 3.5-5 Imagen de Cámara Pushbroom Para Selección de Área. Extraído de script de Matlab desarrollado*



Figura 3.5-6 Imagen de Cámara Snapshot Para Selección de Área. Extraído de script de Matlab desarrollado

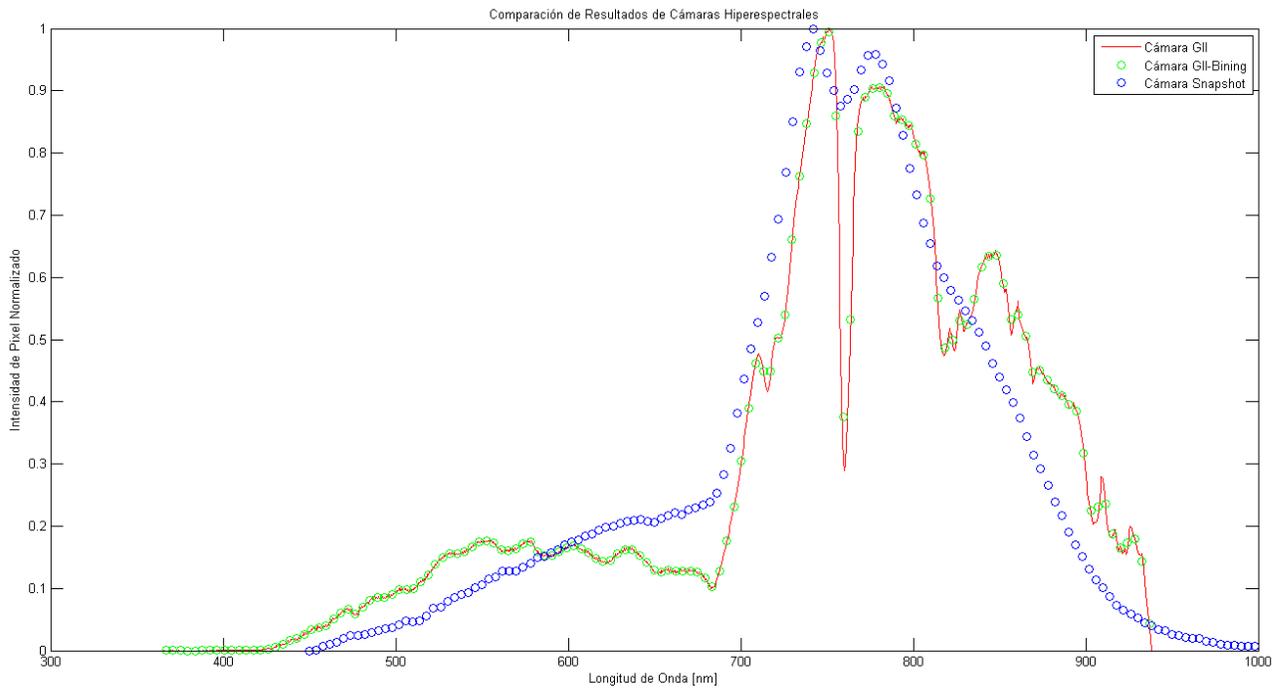


Figura 3.5-7 Gráficas Finales de Comparación Snapshot vs Pushbroom. Extraído de script de Matlab desarrollado

En la gráfica se puede apreciar como la cámara pushbroom tiene una caída entre los espectros 700 y 800 nm, esto se debe a la absorción de ciertas longitudes de onda por la atmósfera. Por su cuenta la cámara snapshot presenta este fenómeno pero con un mayor suavizado de los datos. Esta diferencia fue crítica para el uso del hiperespectrómetro en el proyecto SIMMAP por lo cual los encargados

decidieron no utilizar el modelo UHD-185 hasta que se pudiera mejorar la información u obtener respuestas del proveedor a los problemas dados.

En este sentido las gráficas y scripts de Matlab lograron no solo verificar los datos del barrido de cámara pushbroom sino también establecer dudas sobre la cámara tipo snapshot y su utilidad en el proyecto SIMMAP.

### 3.6. Microcontrolador para Control y Operación de Trípode y Cámara Hiperespectral

Parte importante del desarrollo del control, tanto de la cámara hiperespectral como del gimbal, se centra en cual microcontrolador o empujado se utilizará en el momento de vuelo.

Aunque importante, el proceso de selección se dio de forma sencilla con criterios concisos; se definió utilizar el Mini- PC Fitlet-iA10 con sistema operativo Windows 7 para el control de la cámara hiperespectral y el Gimbal, las razones para tal decisión fueron:

1. Ahorro en Costo: Se ahorra el costo de comprar y traer más equipo, se consideró adecuado aprovechar los sistemas que se poseían actualmente.
2. Acoplado a Cámara Snapshot: El Mini-PC ya se podía utilizar adecuadamente con el hiperespectrómetro snapshot lo que reduce los posibles problemas, se debería asegurar que puede operar el hiperespectrómetro push-broom.
3. Comunicación por Wifi: La capacidad de generar una conexión wifi con la cual comunicarse abre opciones a la forma de operar con el equipo de control, previniendo futuros problemas y generando más opciones de mejora o de desarrollo.

Como única limitante a esta elección se debía probar que las bibliotecas de control del hiperespectrómetro tipo pushbroom, así como los programas básicos de prueba, servían adecuadamente. Esta labor se desarrolló brevemente siendo los únicos problemas la búsqueda de drivers y el hecho de que el fitlet-iA10 no permitía utilizar la interfaz de Windows forms.

Dado que el hiperespectrómetro pushbroom si se podía controlar debidamente con una aplicación win32, se definió que el Mini-PC era adecuado para el proyecto.

Finalmente en la figura 3.6-1 se puede observar el Mini-PC a utilizar así como algunas de las especificaciones más relevantes:

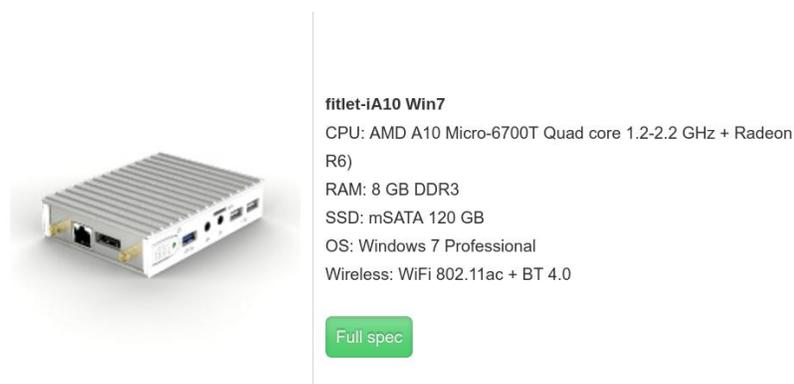


Figura 3.6-1 Mini-PC Fitlet-iA10. Extraído de (fitPC, 2016)

### 3.7. Mejora de Tiempo de Operación en Cámara Pushbroom

Como parte del proceso de control e implementación de la cámara UI-3360 CP, se decidió analizar y optimizar el tiempo de operación, tanto captura como guardado de imagen. Como se mencionó al final de la sección *Configuración de Cámara Hiperespectral UI3360-CP para Operación de Trípode Nexstar*, el tiempo de captura de cada línea ronda los 100 ms, mientras que el tiempo de guardado ronda los 500 ms; si bien los tiempos son aceptables para una aplicación en tierra, las implementaciones en aire se ven limitadas por el tiempo de vuelo del UAV razón por la cual se deseó una mayor frecuencia de trabajo.

La necesidad de optimizar el tiempo de operación se vuelve más crítica al considerar que el tiempo de guardado en la Mini-PC fitlet-iA10 es mucho mayor llegando como máximo a 950 ms por línea capturada y en promedio a los 800 ms. Considerando que un barrido puede requerir entre 100 y 400 líneas esto implica que se pueden utilizar de 95 a 380 segundos de vuelo guardando imágenes.

#### 3.7.1. Método de Análisis de Optimización

Para poder analizar las opciones para la optimización de tiempo se plantea un método de análisis que permita estructurar y encaminar el proceso de búsqueda de solución al problema. El método deja claro los posibles componentes que puedan ser el cuello de botella del proceso y permite una noción clara sobre el camino a seguir. Cabe destacar que para efectos de orden el tiempo de operación se divide en dos componentes: El tiempo de captura y el tiempo de guardado en memoria.

En la figura 3.7-1 se presenta el esquema que describe el método u orden tomado en el análisis de la optimización del tiempo de operación.

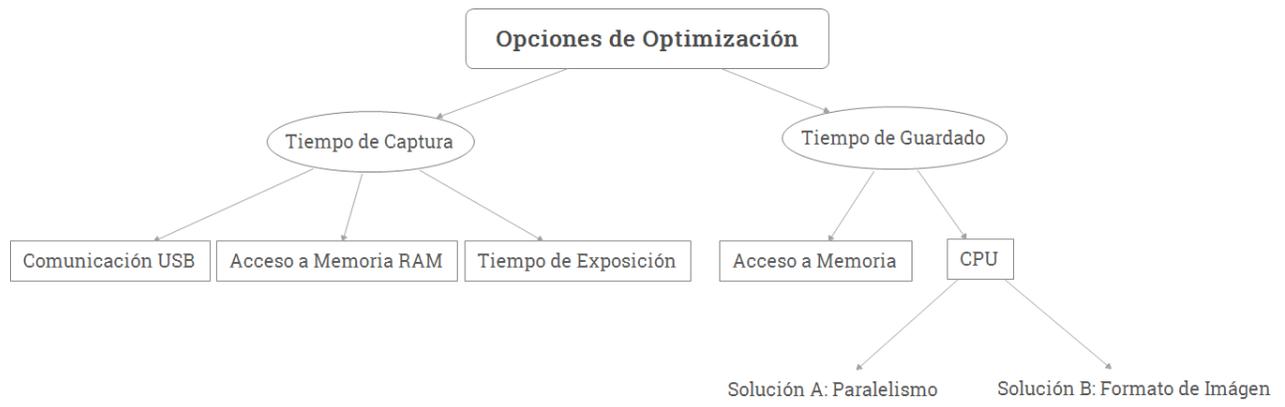


Figura 3.7-1 Mapa Conceptual de Método de Análisis de Opciones de Optimización. Desarrollado en X Mind 7

#### 3.7.2. Tiempo de Captura

El tiempo de captura al momento de buscar la optimización es de 100 ms por línea, como se muestra en la figura 3.8-1, los factores que afectan dicho valor son el acceso a la memoria RAM, el tiempo de exposición y la comunicación USB.

El tiempo de exposición es el tiempo en el cual el sensor de la cámara UI3360-CP se ve expuesto a la luz solar; efectivamente es el tiempo que permite la captura adecuada de una imagen, en la sección *Comprensión de Cámara Hiperepsectral* específicamente en la tabla 3.1-1 se especificó que el hiperespectrómetro con cámara UI3360-CP requiere de 83 ms en condiciones de luz solar abundante, dado que el tiempo depende de la naturaleza, es decir un factor no controlable, no fue posible reducir el tiempo de captura por este camino.

En lo que se refiere al flujo de información a través del cable y puerto USB; la cámara UI3360-CP utiliza un puerto USB 3.0, teóricamente estos son capaces de transmitir información a un ritmo de 5 gigabit por segundo (gbps), o bien 640 MBps; por otro lado cada imagen tomada por la cámara tiene un peso final de hasta 1.5 MB, asumiendo un formato png y resolución de 16 bits por píxel. Al considerar ambos valores se obtiene que cada imagen puede tomar aproximadamente 2.34 ms en transmitirse, al comparar dicho tiempo con el tiempo global de captura de una línea se puede concluir que la transmisión USB no es el cuello de botella en el proceso de guardado.

Finalmente el tiempo restante de captura, 14.66 ms, se considera muy bajo como para tener mejoras apreciables al trabajar sobre el acceso a la memoria RAM.

De esta forma se determinó que los factores del tiempo de captura poseían valores aceptables o bien no era posible incidir en mayor forma sobre los mismo; en este sentido la única solución sería asegurar que la captura de imágenes se realizaría en condiciones de luz solar abundante para asegurar y no empeorar el tiempo de exposición de 83 ms.

### 3.7.3. Tiempo de Guardado

El tiempo de guardado al momento de buscar la optimización es de 800-950 ms por línea, como se muestra en la figura 3.7-1, los factores que afectan dicho valor son el acceso y escritura en memoria y la capacidad de procesamiento de la CPU.

Se debe aclarar que el proceso de análisis es solamente una aproximación con medios básicos pero incompletos y sirve solo para dar una visión inicial del problema de optimización, es necesario en etapas futuras del proyecto SIMMAP el aplicar métodos estándares como lo son los benchmarks o similares para poder apreciar el fenómeno real.

Inicialmente se analizó el tiempo de acceso y escritura en disco; para este análisis se utilizó la herramienta de análisis de rendimiento en el administrador de tareas de Windows, se analizó el porcentaje de tiempo de actividad así como la existencia de cuellos de botella o picos apreciables; cabe destacar que las pruebas se realizaron con ciclos de captura de 100 imágenes, la cantidad suficiente para poder apreciar los datos sin gastar un mayor tiempo entre prueba y prueba. En la figura 3.7-2 se aprecia una captura de pantalla del análisis de rendimiento.

Se reconoce en este punto que el administrador de tareas presenta el acceso a disco de todo proceso en el computador, inclusive el sistema operativo, por lo que la imagen 3.7-2 es solo una referencia inicial y no una figura para toma de decisión final.

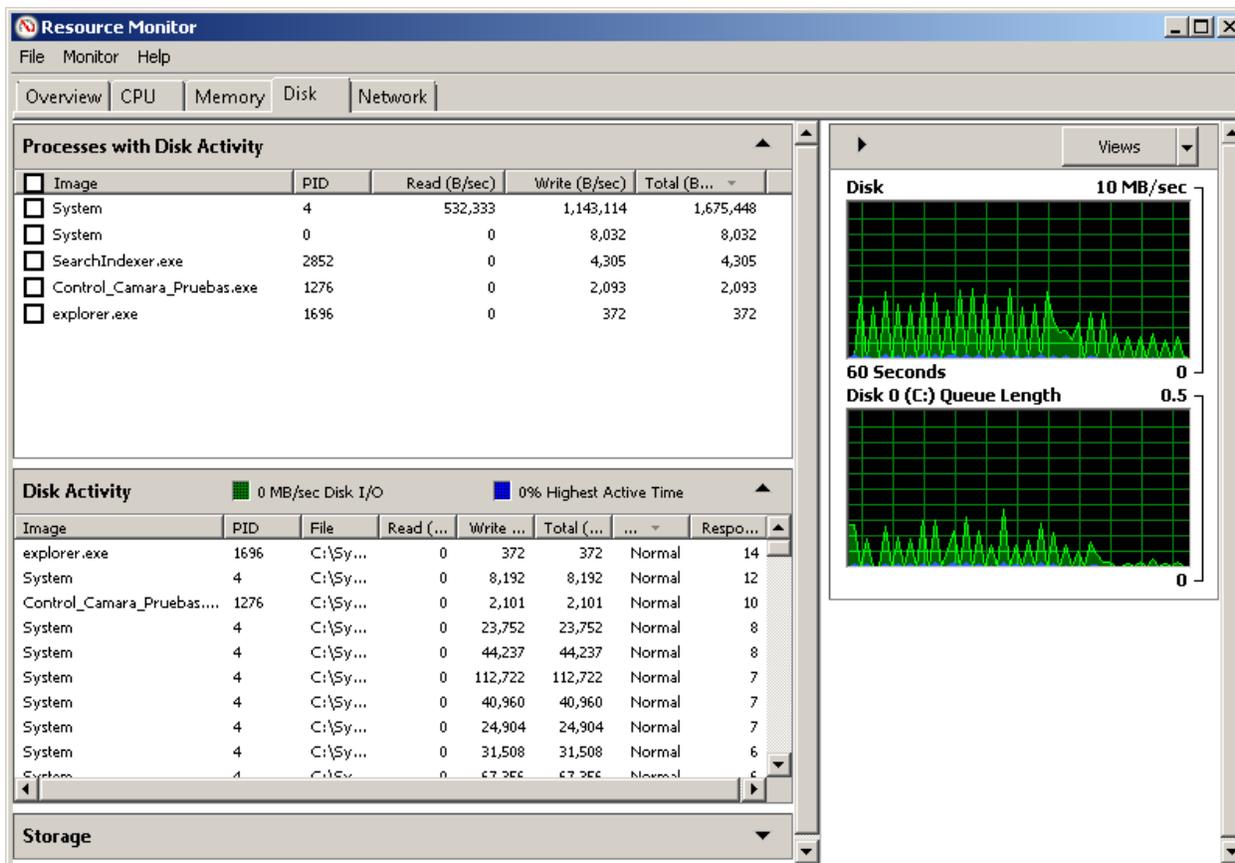


Figura 3.7-2 Análisis de Disco en Mini-PC Fitlet-IA10. Extraída del Administrador de Tareas de Windows

Como se observa en la figura la escritura alcanza valores cercanos a los 2000 Byte/s, aunque bajo no implican que su efecto sobre el acceso a disco sea mínimo, es necesario recordar que el acceso a disco siempre es el mayor cuello de botella para el sistema. En este punto las posibles soluciones rondan la mejora del disco; no obstante, el disco duro del Fitlet-IA10 es un ssd mSATA 3.0 de estado sólido, formalmente de las mejores opciones a nivel comercial, por esta razón se considera que las posibles mejoras deberían ir en línea de un mejor aprovechamiento de los recursos.

Finalmente se realizó un diagnostico superficial del CPU al momento de almacenar imágenes. Nuevamente se utilizó el administrador de tareas de Windows en conjunto con un programa de captura de 100 imágenes para poder llevar a cabo el análisis. En la figura 3.7-3 se aprecia una captura de pantalla del administrador de tareas. Cabe destacar que la imagen 3.7-3 no es prueba o diagnóstico definitivo sino solamente una primera aproximación, a futuro se debe buscar utilizar un benchmark para analizar y calificar adecuadamente el rendimiento del CPU.

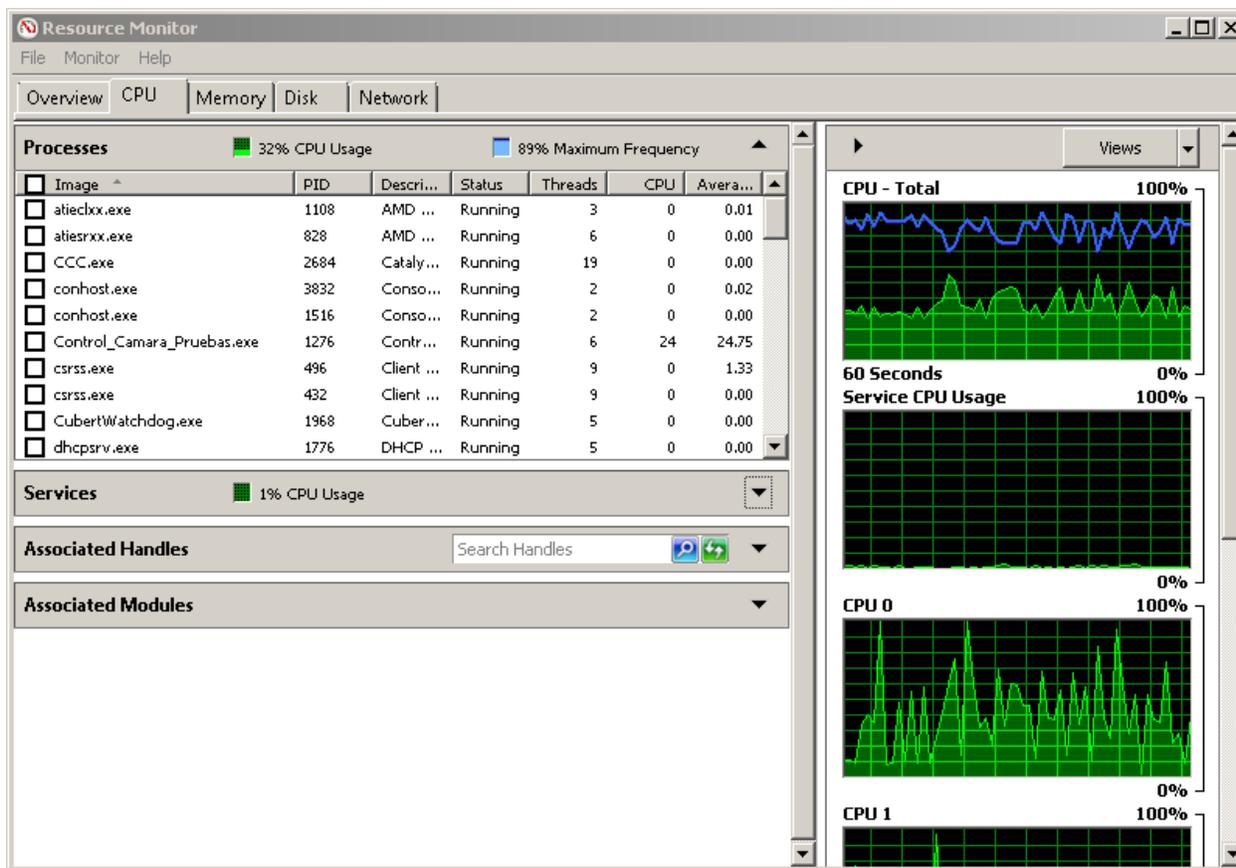


Figura 3.7-3 Análisis de CPU en Mini-PC Fitlet-iA10. Extraída del Administrador de Tareas de Windows

Como se observa en la figura bajo la categoría del programa “Control\_Cámara\_Pruebas”, el programa de pruebas para el estudio de tiempo de ejecución, se puede observar cómo se consume un 24% del CPU. Al considerar que el procesador del fitlet-iA10 es un quad-core se puede comprender que el proceso de guardado satura, en recursos el equivalente a un núcleo; aunque formalmente no satura un núcleo propiamente. Este hecho demuestra como el uso del CPU presenta una limitante en el proceso de trabajo, cabe aclarar que se atribuye el problema no tanto al CPU como tal ya que opera de forma esperada sino a como el programa diseñado utiliza los recursos del CPU.

De esta forma se define que la lógica de control implementada no aprovecha totalmente los recursos del sistema empotrado, y aunque el cuello de botella se encuentra en el acceso a disco, una forma aceptable de mejorar el tiempo de guardado es al variar o mejorar el programa de control diseñado. Ante esto se presentan dos opciones la distribución del proceso mediante programación por hilos y la reducción del tamaño de imágenes al variar el formato de las mismas. Se desarrolla a continuación las pruebas y análisis realizados con las dos últimas opciones.

### 3.7.3.1. Paralelismo: Programación por Hilos

Como opción a distribuir el proceso, y por ende el consumo de CPU, se buscó la posibilidad de implementar paralelismo mediante la programación por hilos, inicialmente se buscó poder hacer una

ejecución paralela que permitiera tener múltiples hilos guardando grupos distintos de imágenes; para poder comprender el propósito inicial se puede observar la figura 3.7-4.

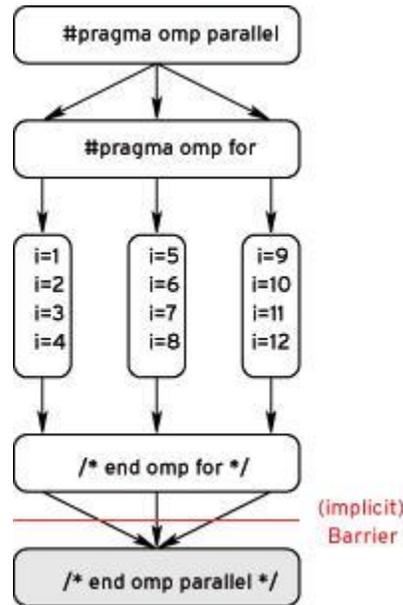


Figura 3.7-4 Secuencia de Trabajo de Programación por Hilos. Extraído de (Wolfgang, 2016)

En la figura se aprecia como un ciclo for se divide entre tres hilos para que cada uno ejecute una sección del mismo, en el caso desarrollado se sustituye cada ejecución  $i=\#$  por cada llamada a el método de guardado `is_ImageFile` de la cámara UI3360-CP. Previo a implementar la lógica se revisó el manual de operación de programación de la cámara en caso de que existiera algún apoyo para programación por hilos, esto permitió encontrar una aclaración del proveedor donde deja claro que la biblioteca de programación es compatible con el uso de hilos siempre y cuando no se trate de acceder a la misma función con dos hilos distintos; inclusive la biblioteca fue diseñada para evitar que dos hilos puedan ejecutar la misma función al mismo tiempo.

Esta decisión de diseño en la biblioteca provoca que la lógica a implementar no fuese viable, efectivamente eliminando la principal razón por la cual se buscaba la programación por hilos, distribuir el proceso de guardado en más de un núcleo.

En este punto se decidió probar una segunda alternativa que podría reducir el tiempo de operación total aun cuando no podría trabajar directamente sobre el tiempo de guardado, esto es paralelizar las operaciones de captura y guardado de imágenes. Dicha labor tendría por objetivo el ahorrar tiempo de ejecución al hacer un uso eficiente del mismo; propiamente no se busca reducir el tiempo de guardado o el tiempo de captura sino traslapar los lapsos de captura y guardado, en vez de capturar 200 imágenes y luego guardar las 200 se podría capturar 100, empezar el guardado de dichas 100 imágenes y realizar la segunda captura de 100 imágenes. Para entender el proceso realizado así como las ventajas temporales que este implica se pueden observar las figuras 3.7-5 y 3.7-6.

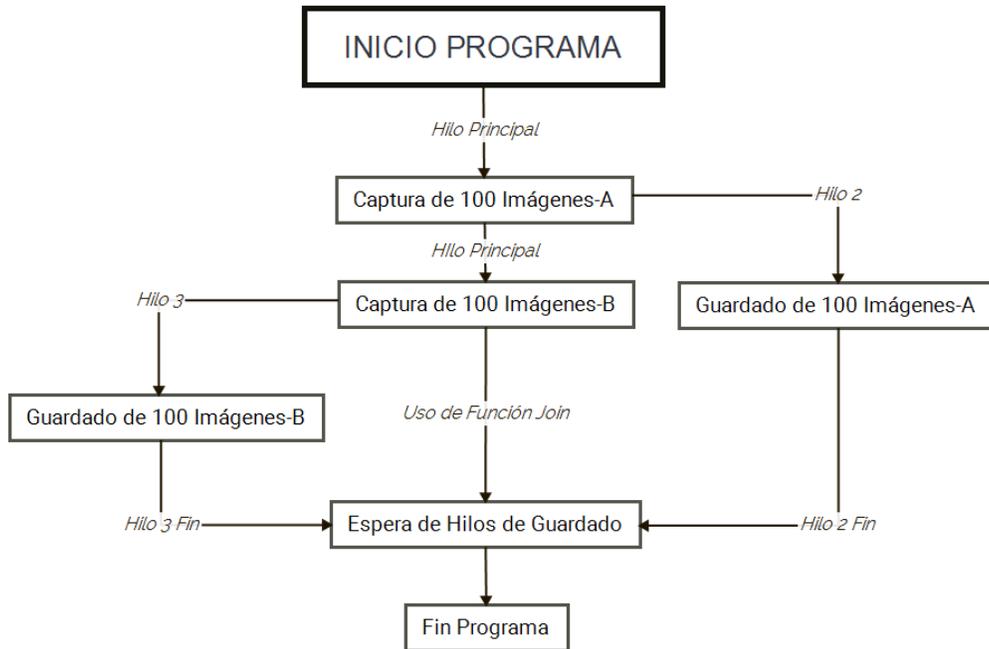


Figura 3.7-5 Lógica de Hilos en Captura y Guardado de Imágenes. Desarrollado en X Mind 7

Ejecución Actual Tguardado>Tcaptura			
Captura	100A	100B	
Guardado			100A 100B
Ejecución Lineal si Tcaptura>Tguardado			
Captura	100A	100B	
Guardado			100A 100B
Hilos Si Tcaptura>Tguardado			
Captura	100A	100B	
Guardado		100A	100B
Hilos Si Tguardado>Tcaptura			
Captura	100A	100B	
Guardado		100A	100B

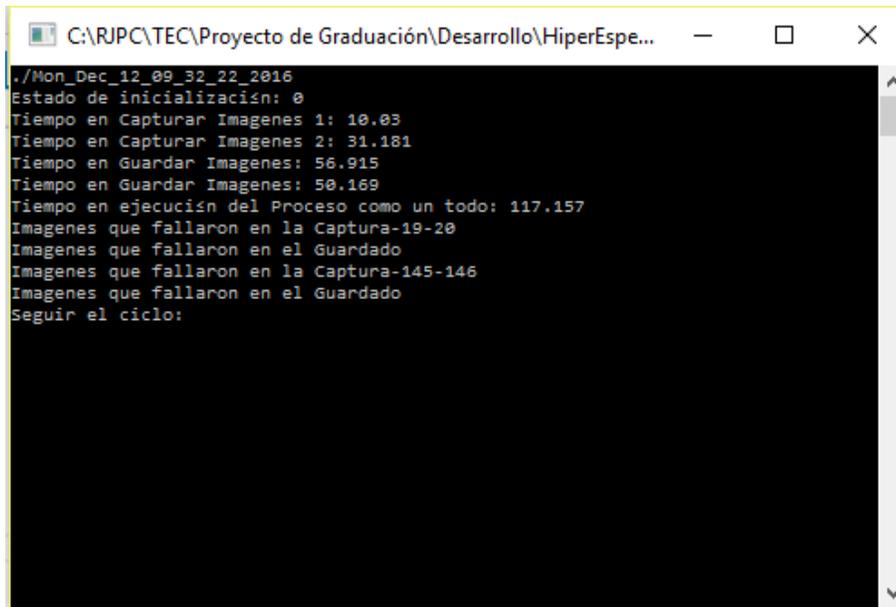
Figura 3.7-6 Comparación de Duración entre Lógicas con y sin Hilos. Desarrollado en Microsoft Excel 2013

La figura 3.7-5 demuestra la lógica de hilos a implementar donde después de cada sesión de captura se genera un hilo para el almacenado de la información. Por otro lado la figura 3.7-6 demuestra una comparación del tiempo de ejecución entre los distintos casos; el primer caso muestra el estado al tiempo de optimización y el cuarto caso presenta la opción viable a desarrollar con la lógica de hilos, se puede apreciar que el tiempo invertido en la segunda captura “100B” se puede ahorrar al guardar en paralelo las imágenes “100A”, si bien esta lógica no reduce propiamente el tiempo de guardado es un método secundario que puede reducir tiempo de operación como tal. Cabe aclarar que el segundo y tercer

casos se presentan para contemplar la utilidad de la programación por hilos en caso de reducir el tiempo de guardado a menos de 100 ms, aplicando otros métodos.

Para poder implementar el código de pruebas se utilizó la biblioteca pthreads de C++ y se modificaron las distintas funciones de control de cámara para acoplarse a su llamada por hilos. Se realizaron dos tipos de pruebas: En un laptop aparte para probar la lógica y la ejecución adecuada del código y seguidamente en el Mini-PC para comprobar los resultados de la primera prueba.

En la figura 3.7-7 se puede observar una captura del programa de pruebas con el cual se tomaron los tiempos, este caso representa las pruebas sobre la laptop de pruebas.



```
C:\RIPC\TEC\Proyecto de Graduación\Desarrollo\HiperEspe...
./Mon_Dec_12_09_32_22_2016
Estado de inicialización: 0
Tiempo en Capturar Imagenes 1: 10.03
Tiempo en Capturar Imagenes 2: 31.181
Tiempo en Guardar Imagenes: 56.915
Tiempo en Guardar Imagenes: 50.169
Tiempo en ejecución del Proceso como un todo: 117.157
Imagenes que fallaron en la Captura-19-20
Imagenes que fallaron en el Guardado
Imagenes que fallaron en la Captura-145-146
Imagenes que fallaron en el Guardado
Seguir el ciclo:
```

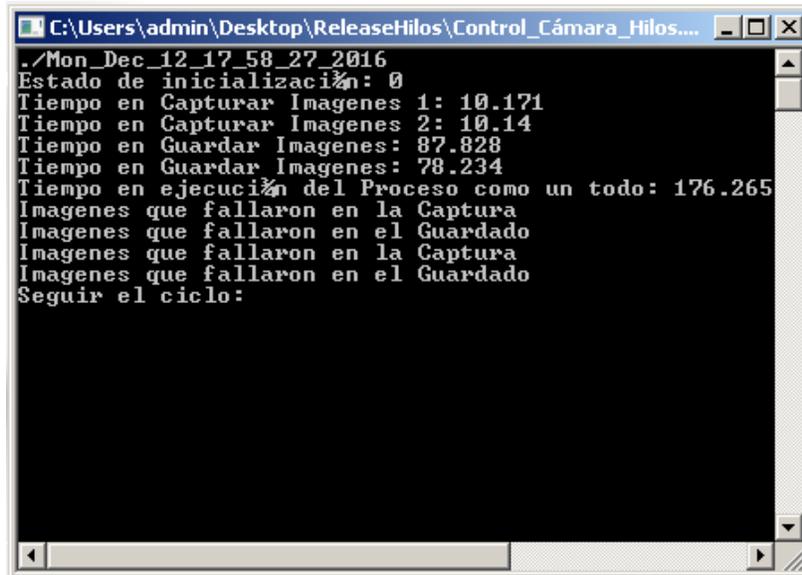
Figura 3.7-7 Captura de Pantalla de Ejecución de Programa de Pruebas con Hilos. Desarrollado en Microsoft Visual Studio Express 2012

Realizando la suma de todos los tiempos independientes se obtiene un valor de 148.295 s, valor menor al tiempo de ejecución real de 117.157 s; esta diferencia demuestra la existencia real del paralelismo en el programa de pruebas, por otro lado la diferencia entre ambos tiempos es de 31.138 s un valor cercano a la duración de la segunda captura que prueba la teoría presentada en la figura 3.7-6 es decir que el paralelismo permitió ahorrar el tiempo de captura “100B”. No obstante contrario a la lógica esperada los tiempos de captura no fueron constantes.

Analizando la biblioteca y sus limitantes se deduce que una posible razón de este fenómeno es que el acceso a memoria RAM en la biblioteca no se puede paralelizar, es decir el método de captura de imágenes accede a la memoria RAM para poder guardar las imágenes y al mismo tiempo el método `is_ImageFile` accede a la memoria RAM para poder almacenar la información en memoria fija; se presume que la biblioteca maneja un único camino o método de acceso a RAM y aunque se trabaje en espacios de memoria diferente esta limitante provoca que se deban tomar turnos para la captura o guardado. La variación entre los tiempos de captura soporta esta teoría. Este defecto solo implica que realmente se ahorran 10 segundos; no obstante también da a entender que el ritmo al cual se capturan imágenes varía

de una tanda a otra de captura, según el estilo de barrido a utilizar esto puede introducir problemas de distribución no equidistante entre las líneas de la captura.

Por otro lado la figura 3.7-8 presenta el resultado de utilizar el mismo programa de pruebas en la Mini PC fitlet-iA10.



```
C:\Users\admin\Desktop\ReleaseHilos\Control_Cámara_Hilos...
./Mon_Dec_12_17_58_27_2016
Estado de inicialización: 0
Tiempo en Capturar Imagenes 1: 10.171
Tiempo en Capturar Imagenes 2: 10.14
Tiempo en Guardar Imagenes: 87.828
Tiempo en Guardar Imagenes: 78.234
Tiempo en ejecución del Proceso como un todo: 176.265
Imagenes que fallaron en la Captura
Imagenes que fallaron en el Guardado
Imagenes que fallaron en la Captura
Imagenes que fallaron en el Guardado
Seguir el ciclo:
```

Figura 3.7-8 Captura de Pantalla de Ejecución de Programa de Pruebas con Hilos en Mini-PC. Desarrollado en Microsoft Visual Studio Express 2012

Realizando la suma de todos los tiempos, de cada operación individual, se obtiene un valor de 186.373, que es aproximadamente 10 segundos mayor al tiempo real de ejecución de 176.265; si bien a primera entrada esto da a entender que el paralelismo se realizó adecuadamente, e inclusive sin el problema presente en la laptop, al realizar futuras pruebas se percató un problema mayor y es que el computador fitlet-iA10 no realiza un verdadero paralelismo.

Para poder notar esto se observó la carpeta de almacenamiento de imágenes en lo que se ejecutaba el programa, en caso de haber paralelismo se supone que imágenes se guardarían en la carpeta correspondiente desde el momento en que se presenta el mensaje “Tiempo de Capturar Imágenes 1: #”; no obstante se observó que la carpeta correspondiente empezó a almacenar archivos .png hasta después de que se presentara el segundo mensaje de captura de imágenes; es decir, el guardado empezaba hasta después de terminar toda la captura. Para una segunda revisión se utilizó el monitor de recursos del administrador de tareas de Windows, en esta aplicación se puede observar la cantidad de hilos que maneja cada programa, se notó que efectivamente después del primer mensaje de captura la cantidad de hilos aumentaba en uno, el de guardado de imágenes, lo cual aseguraba la existencia del hilo de guardado y confirmaba que a nivel de sistema operativo o de biblioteca no se daba acceso a RAM a dicho hilo. La figura 3.7-9 muestra un ejemplo del fenómeno sucedido.

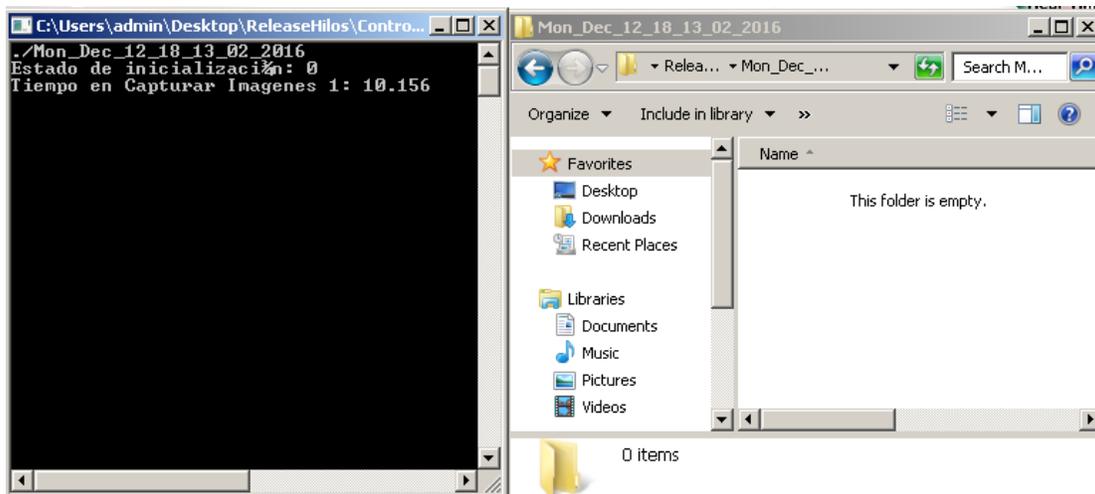


Figura 3.7-9 Captura de Pantalla de Falta de Paralelismo en Pruebas de Programa con Lógica de Hilos en Mini-PC. Extraída con la herramienta Recortes de Windows.

De esta forma la segunda prueba de uso de hilos no alcanzó resultados del todo positivos, si bien en la laptop se pudo ahorrar algunos segundos en el tiempo de captura, esto no se logró en el sistema empotrado principalmente por limitantes de la biblioteca. Si bien se definió desde el inicio de que la lógica de hilos no lograría reducir el tiempo de guardado como tal, dado los bloqueos de la biblioteca de control de cámara, las lógicas secundarias resultaron no ser del todo útiles para ahorrar tiempo de operación como tal.

En este punto es necesario aclarar que la lógica realizada se diseñó con la biblioteca pthreads, sería necesario realizar pruebas a futuro con la biblioteca Open MP buscando asignar núcleos específicos a labores específicas.

### 3.7.3.2. Variaciones del Formato de Guardado

Como última opción a la reducción del tiempo de guardado de una imagen se decidió probar otros formatos de almacenado. Al momento de estudiar el proceso de optimización se trabajaba con formato png con una profundidad de píxel de 16 bits por píxel, este formato se seleccionó ya que proveía la información más detallada y de mayor calidad lo cual favorecía el futuro procesado que se le debería hacer.

Como variantes se decidió probar tres formatos distintos, todos a una profundidad de 8 bits por píxel, específicamente png, bmp y jpg. En este punto se reconoció que la reducción de bits por píxel así como el cambio de formato podría implicar una distorsión de la información; por este motivo se dividió la prueba de variación de formato en dos partes: En primer lugar se tomaría nota del tiempo de guardado variando los formatos y como segunda labor se utilizaría los .m diseñados previamente para comparar la información entre el formato png a 16 bit contra los otros tres formatos, el resultado de la segunda labor se le entregaría a la encargada respectiva del análisis de información hiperespectral para recibir una valoración adecuada del formato.

Variar los formatos de captura se basó en leves ajustes de variables en los programas diseñados, la lógica de captura no se vio afectada por este detalle. En las tablas 3.7-1 y 3.7-2 se presenta la

información recaudada del proceso de pruebas y comparación, cabe destacar que los valores presentes son el producto de un promedio de 25 mediciones individuales y el orden en que se presentan los formatos va de mayor a menor calidad.

Tabla 3.7-1 Mediciones de Tiempo de Guardado en Laptop de Pruebas

Líneas	1	10	100
Tiempo PNG 16 bit (s)	0.50798	4.7558	46.8086
Tiempo PNG 8 bit (s)	0.17804	1.7844	18.3742
Tiempo BMP (s)	0.01524	0.15069	3.8647
Tiempo JPG (s)	0.01436	0.13688	1.4182

Tabla 3.7-2 Mediciones de Tiempo de Guardado en Mini-PC Fitlet-iA10

Líneas	1	10	100
Tiempo PNG 16 bit (s)	0.82	7.67948	61.0719
Tiempo PNG 8 bit (s)	0.423	4.28196	41.7123
Tiempo BMP (s)	0.02091	0.29452	3.34268
Tiempo JPG (s)	0.0288	0.3894	4.6676

Se aprecia en ambas tablas un cambio drástico en el tiempo de almacenado, especialmente al comparar un formato de alta compresión como el JPG contra el PNG a 16 bits. Específicamente el formato JPG posee, en casi todos los casos, la ventaja en tiempo de guardado, la única excepción es el caso particular del almacenado JPG contra BMP en el fitlet-iA10, aun realizando múltiples tandas de medición los resultados siempre mostraron que el formato BMP tomaba menos tiempo que el JPG. No obstante al ser el BMP un mejor formato que el JPG, y al presentar ambos un tiempo entre 13 y 18 veces menor al del formato original, se decidió ignorar la particularidad.

Para comparar los formatos se utilizó una modificación del programa diseñado en la subsección 8.5 *Comparación Gráfica de Información Hiperespectral entre Cámara Push-broom y Snapshot*, la diferencia siendo que no se utilizan barridos sino una única captura y no se selecciona un área sino que se utiliza un único píxel espacial predeterminado. Para capturar todas las imágenes, asegurando una misma escena y un tiempo corto, se utilizó otra variante del programa de pruebas diseñado para la cámara pushbroom en la cual se variaban los valores de formato y ajustes de cámara entre cada captura permitiendo así obtener las cuatro líneas a comparar en un tiempo menor a los dos segundos. En las figuras 3.7-10, 8.8-11, 3.7-12, 3.7-13 y 3.7-14 se observan todas las gráficas comparativas generadas para juzgar los formatos de captura.

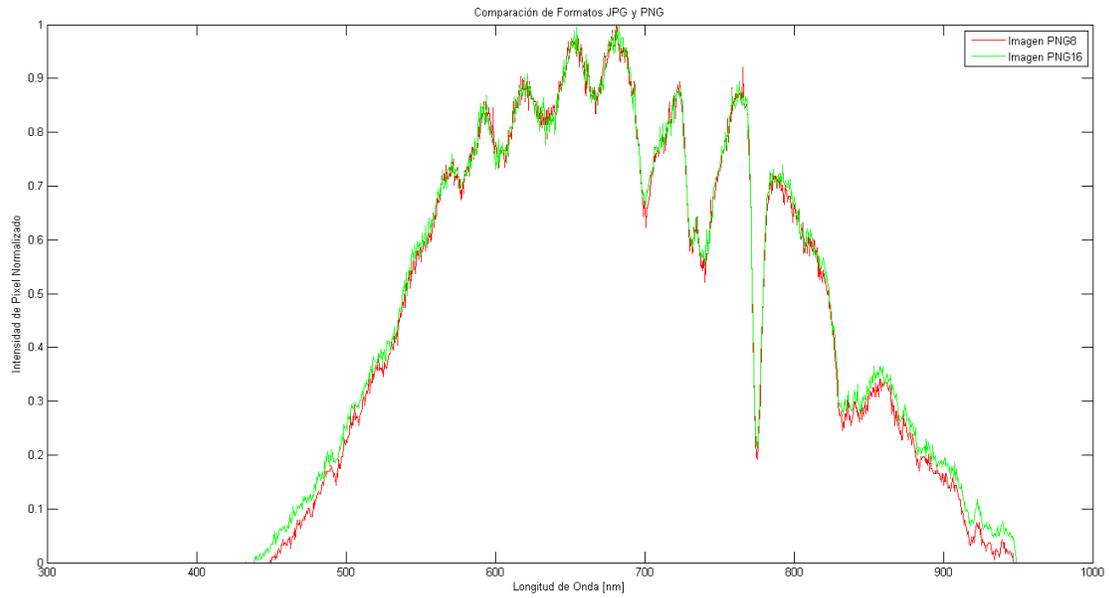


Figura 3.7-10 Gráfica de Comparación entre Formatos PNG 16 bit y PNG 8 bit. Extraído de script de Matlab desarrollado

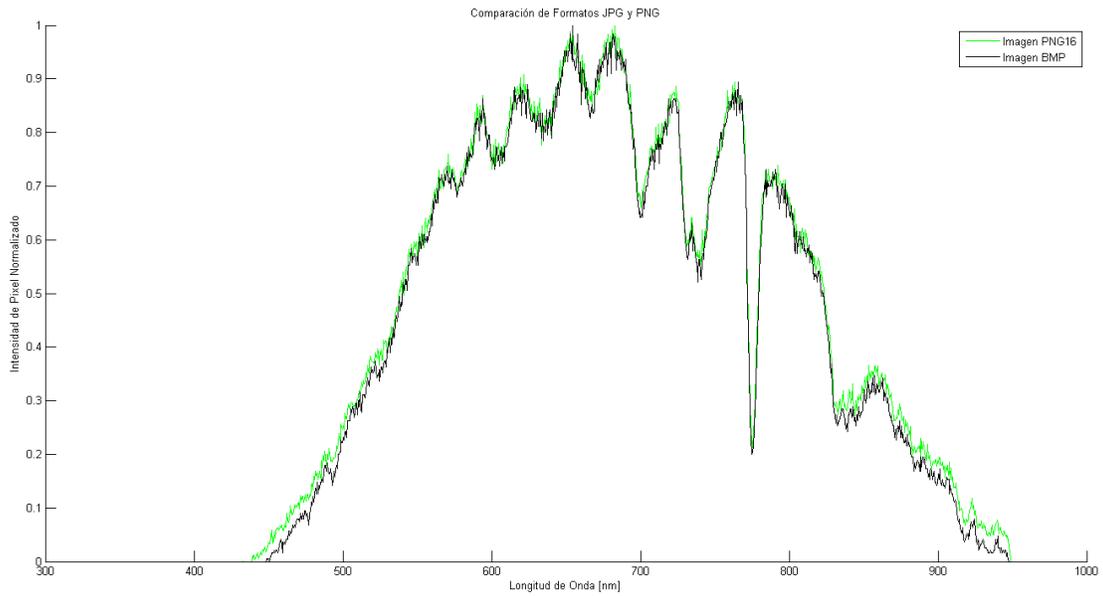


Figura 3.7-11 Gráfica de Comparación entre Formatos PNG 16 bit y BMP. Extraído de script de Matlab desarrollado

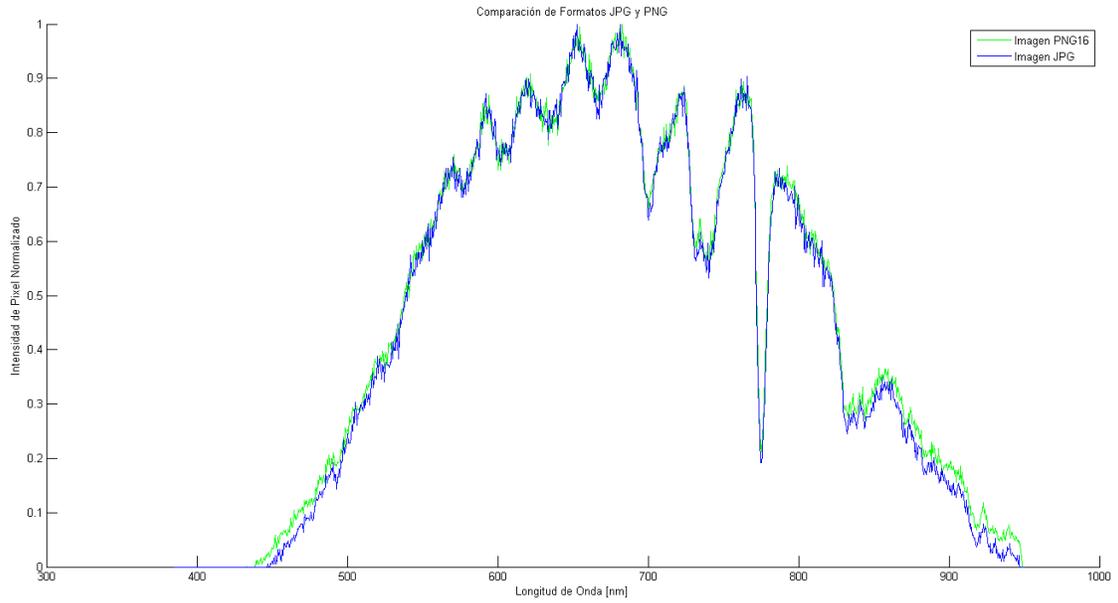


Figura 3.7-12 Gráfica de Comparación entre Formatos PNG 16 bit y JPG. Extraído de script de Matlab desarrollado

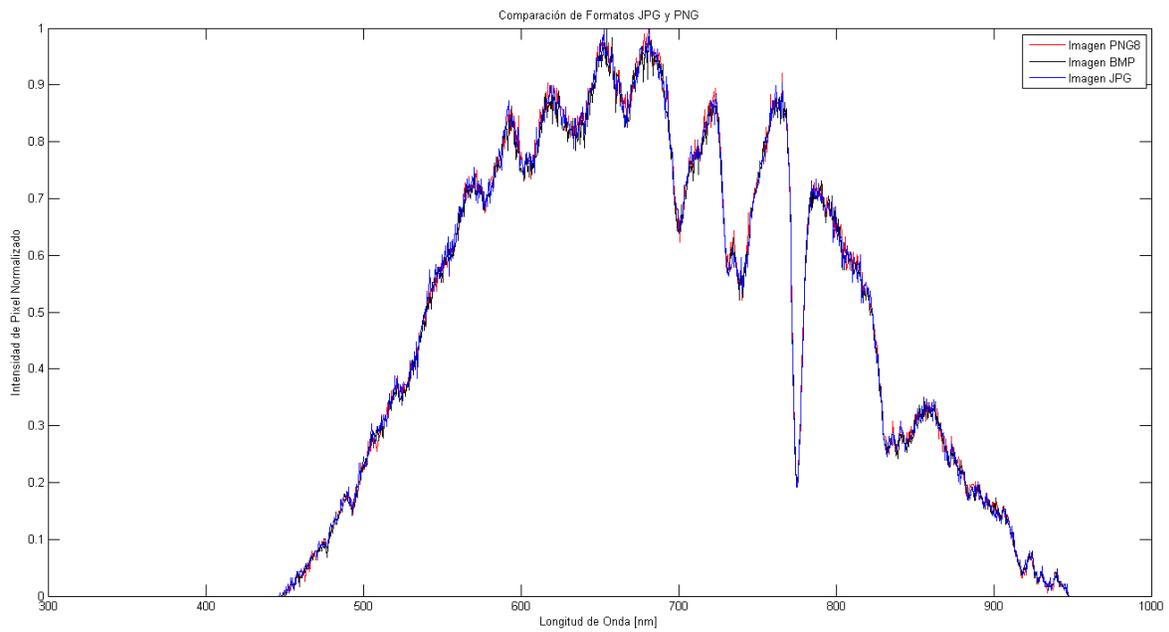


Figura 3.7-13 Gráfica de Comparación entre Formatos PNG 8 bit, BMP y JPG. Extraído de script de Matlab desarrollado

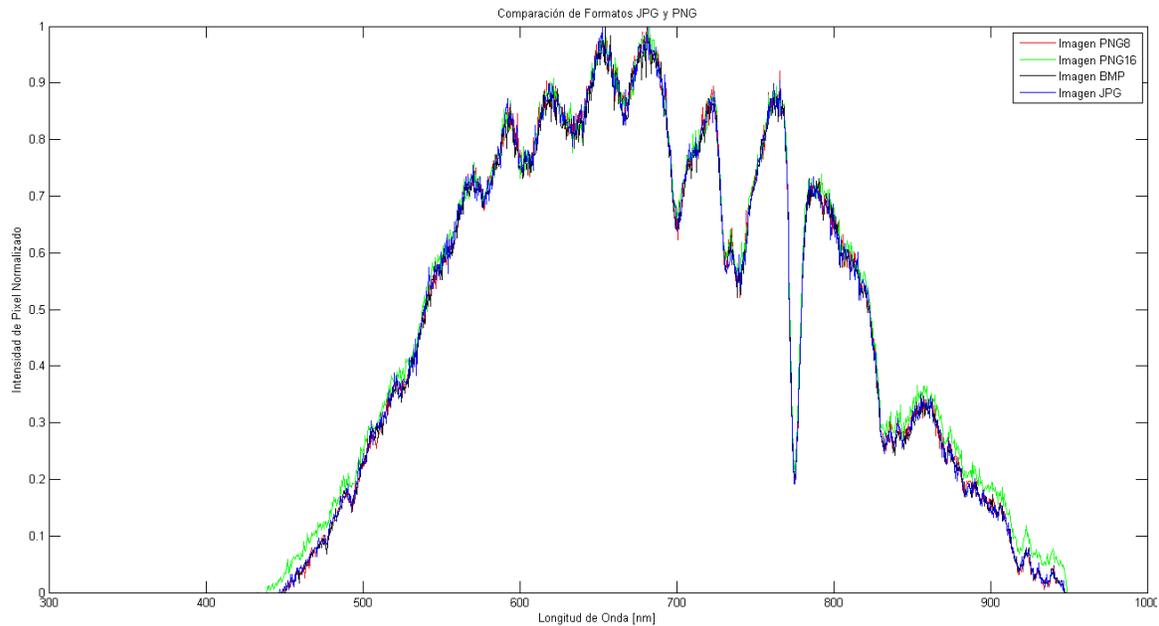


Figura 3.7-14 Gráfica de Comparación entre todos los Formatos. Extraído de script de Matlab desarrollado

A primera impresión se observa que las mayores diferencias se encuentran en los extremos de la gráfica; no obstante, los perfiles son muy similares entre sí. Se entregaron las figuras presentes a la encargada del análisis de la información hiperespectral y en consultas posteriores se definió que el parecido entre los formatos de bajo detalle y el PNG de 16 bit era aceptable para los requisitos de las futuras etapas de procesamiento.

Como conclusión al proceso de optimización se definió asegurar que el control de la cámara hiperespectral pudiese optar libremente por los formatos de 8 bit, aprovechando así su tiempo de almacenado rápido, no se utilizaría a priori la lógica de hilos; no obstante, se dejó el código y las explicaciones para implementar dicha lógica en caso de que futuras iteraciones del proyecto lo consideren necesario, además se recomienda trabajar con la biblioteca Open MP sobre pthreads.

### 3.8. Control de Gimbal CAME Mini 3-Air

Conocida la forma de controlar ambas cámara hiperespectrales se debe asegurar la posibilidad y el método con el cual se puede operar el gimbal seleccionado, CAME Mini 3-Air, mediante una aplicación propia sin dependencia de las interfaces de usuario provistas por el proveedor. Las exigencias de dicha operación varían según se utilice la cámara snapshot o push-broom; entre las cuales la cámara push-broom plantea la mayor cantidad de exigencias:

- Encender o apagar el gimbal o sus motores a conveniencia: En caso de desestabilización o similares se debe tener un método de emergencia con el cual se apague el gimbal.

- Operaciones de barrido: La capacidad de poder realizar las operaciones de barrido “Movimiento a Pasos” o “Movimiento Continuo” según se describieron en la sección 3.3.4 *Lógica de Barrido*. Esto aplica únicamente para la cámara push-broom.

Otras capacidades como la modificación de parámetros PID para el control se pueden realizar previo a la puesta en marcha del equipo razón por la cual no es necesario que el control propio del gimbal contemple estas opciones. Se describe a continuación el método con el cual se puede operar el gimbal, así como los programas de prueba implementados y los resultados al aplicarlo al gimbal.

### 3.8.1. Protocolo de Comunicación Serial Simple-BGC

El gimbal CAME Mini 3-air utiliza el controlador simple BGC de la compañía BaseCam Electronics, consultando los manuales respectivos se descubrió que el controlador posee un API, o interfaz para programación de aplicaciones por sus siglas en inglés, basada en comunicación serial tipo UART. Entre otros detalles la comunicación se da a 115200 baudios, con un bit de parada y sin paridad o bien paridad par (BaseCam Electronics, 2013). En este sentido la labor se reduce a entender el protocolo de comunicación serial así como encontrar el formato de los comandos deseados.

La comunicación serial del simple BGC consta de paquetes de bytes divididos en dos partes, encabezado y cuerpo, a su vez en conjunto ambas secciones abarcan seis subsecciones:

1. Byte de Inicio “>”: Parte del encabezado es un byte constante “>” el cual es interpretado como la entrada de un comando.
2. Identificador del Comando: Byte sin signo que almacena el valor numérico del comando a utilizar. Cada comando existente tiene asociado un único identificador.
3. Tamaño de Data: Byte sin signo que almacena la cantidad de bytes a ser enviados en la sección de Data. Como máximo se pueden enviar 255 bytes y puede tener un valor de 0 en caso de que el comando no requiera de datos.
4. Suma de encabezado: Ultimo elemento del encabezado, byte sin signo que debe responder a la fórmula *ID de Comando + Tamaño de Data*
5. Data: Serie de bytes con la información necesaria para que el comando ejecute su operación.
6. Suma de Data: Un byte sin signo que debe responder a la ecuación  $\sum Valor\ de\ Bytes \% 256$ .

Como ejemplo de este formato se puede observar la figura 3.8-1, tomada del manual propio de la API serial donde se muestra la forma de un comando genera (BaseCam Electronics, 2016).

0x3E (>)	0x52 (R)	0x01	0x53	0x01	0x01
	command id	data size	header checksum	data	body checksum
header				body	

Figura 3.8-1 Ejemplo de Comando de la Serial API del Controlador Simple BGC. Extraído de (BaseCam Electronics, 2016)

Cabe destacar que el manual de operación aconseja esperara entre 10-20 ms entre él envió de un comando a otro. De la misma forma cuando un dato requiere de más de un byte para almacenar su información la API asume que se enviara primero la parte baja del byte y luego la parte alta.

Comprendido el formato general bajo el cual se pueden enviar comandos al controlador del gimbal se debe analizar las particularidades de los comandos específicos a implementar. Según la lectura del manual los comandos deseados son “CMD\_MOTORS\_ON”, “CMD\_MOTORS\_OFF” y “CMD\_CONTROL”.

#### 3.8.1.1. CMD\_MOTORS\_ON y CMD\_MOTORS\_OFF

Aunque posean ID distintas, los comandos de encendido y apagado de motores realizan una operación similar y su sintaxis es igual. Como su nombre lo indican los comandos encienden o apagan los motores del gimbal según sea necesario, no requieren de datos para su operación razón por la cual el campo de Data no se utiliza y el byte de “Tamaño de Data” debe ser de 0. Específicamente MOTORS\_ON responde a la ID 77 y MOTORS\_OFF responde a la ID 109.

#### 3.8.1.2. CMD\_CONTROL

El comando “CONTROL”, permite desplazar el gimbal a una posición específica o hacer que el mismo se mueva de forma continua sobre un eje, esto especificando valores de ángulo y velocidad en los ejes cabeceo, alabeo y guiñada según corresponda. Efectivamente el comando permite realizar operaciones de barrido tanto del tipo “Movimiento a Pasos” como “Movimiento Continuo”. Responde a la ID 67 y requiere de 13 bytes en su campo de Data, la información que requiere para operar es:

- Modo: Un byte sin signo que especifica el estilo de control a implementar, entre las opciones se encuentra sin control (0), velocidad (1), ángulo (2), velocidad con ángulo (3) y control remoto (4). El barrido de movimiento a pasos responde al estilo por ángulo y el barrido de movimiento continuo se implementa en el modo de velocidad.
- Velocidad Alabeo: Dos bytes con signo que determinan la velocidad a usar en el alabeo, si el modo de control es de ángulo y el valor dado es de 0, se utiliza la velocidad por defecto de fábrica.
- Ángulo Alabeo: Dos bytes con signo que representan el ángulo de alabeo objetivo para el modo de control por ángulo, el valor se ignora en el control por velocidad.
- Velocidad Cabeceo: Igual a velocidad alabeo pero para el cabeceo.
- Ángulo Cabeceo: Igual al ángulo alabeo pero para el cabeceo.
- Velocidad Guiñada
- Ángulo Guiñada

Cabe destacar que la referencia para los valores de ángulo se define en la interfaz de usuario según la orientación de la IMU; es decir la orientación definida en la IMU marca el punto 0° para el control. Por otro lado el orden en que se envían los bytes es el mismo en el cual se presentaron, respetando el orden entre byte bajo y byte alto en los datos que requieren más de un byte.

Finalmente para comprender la sintaxis del comando se debe aclarar el formato en que se calculan y presentan los datos de velocidad y ángulo; es decir la forma en que se asigna el valor de los datos velocidad y ángulo no es directo como asignar “Ángulo Alabeo = 30°”, sino que se da en valores definidos por las siguientes fórmulas:

$$V (\text{Comando}) = \frac{V(^{\circ}/s)}{0.1220740379} \quad (3-9)$$

$$\alpha (\text{Comando}) = \frac{\alpha (^{\circ}) * 16384}{360} \quad (3-10)$$

Siguiendo estas ecuaciones se asignan los dos bytes con signo que almacenan la información de velocidad y ángulo. Como ejemplo de esto se puede observar el caso donde se quiere asignar un ángulo de 30. Siguiendo la fórmula anterior se obtiene el valor 1365,33, a su vez este valor en sus dos bytes se representaría de la forma: 00000101\_01010110 lo cual se traduce a que el byte alto a enviar debe ser 5 y el byte bajo debe ser 86; enviando estos valores en el orden bajo->alto se logra asignar un ángulo de 30°.

En el caso de un número negativo se sigue el protocolo de complemento a dos, por ejemplo para el valor de ángulo de -40°, la fórmula retorna el valor de -1820,44 lo que se representaría de la forma: 11111000\_11100100 es decir los valores 248 y 228 respectivamente. En este caso se debe enviar los bytes 228 y 248 en ese orden para asignar un ángulo de -40°.

### 3.8.2. Interfaz de Pruebas en C++

Para poder probar el serial API se decidió implementar una interfaz simple de win32 en C++, el lenguaje se debe a la elección de utilizar el Mini-PC y la coherencia con los demás programas desarrollados hasta la fecha. La interfaz permite al usuario elegir entre los comandos de MOTORS\_ON, MOTORS\_OFF y CONTROL, en el último caso solicita al usuario todos los datos necesarios y envía un mensaje a través de puerto serie. En la figura 3.8-2 se puede apreciar la interfaz implementada.

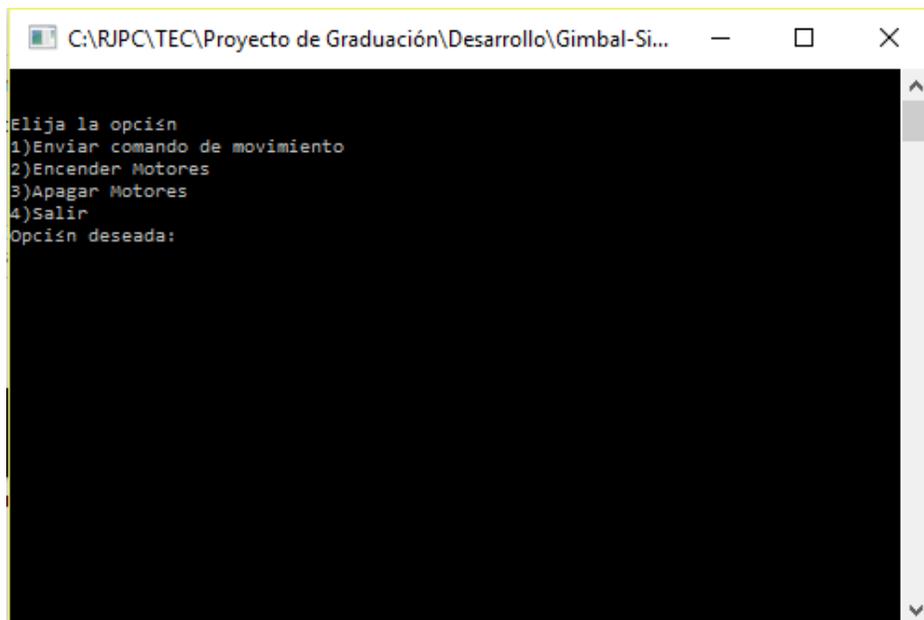


Figura 3.8-2 Interfaz de Pruebas para Comunicación Serial con Controlador SimpleBGC. Desarrollado en Microsoft Visual Studio Express 2012

Cabe destacar que parte de la comprensión del API serial del simple BGC, así como parte del código implementado se tomó de la biblioteca diseñada por el proveedor. Entre otras cosas la biblioteca

permitió comprender el orden en que se debe enviar la información conformada por más de un byte, proveyó estructuras de objetos útiles para el manejo y orden de la información de los datos del comando CMD\_CONTROL y una variedad de definiciones para el programa de prueba desarrollado. No se utilizó directamente la biblioteca ya que la misma fue diseñada para aplicaciones en arduino, por esta razón se optó por la adaptación de la misma.

Previo a su implementación con el gimbal se debió probar que la interfaz enviaba la información debida, para esto se conectó la salida serial del computador de pruebas al osciloscopio Tektronix MSO 4054, se diseñaron y calcularon los valores de varios mensajes específicos para así verificar que el programa enviaba correctamente la información por el puerto serie.

En el caso de los comandos de encendido y apagado de motores la sintaxis de los mismos es sencilla ya que carecen de datos. Específicamente se deben enviar los bytes en el orden presente en la tabla 3.8-1:

Tabla 3.8-1 Información a Enviar para Encendido o Apagado de Motores de Controlador Simple BGC

Sección	Inicio ">"	ID Comando	Tamaño de Data	Suma de Encabezado	Data	Suma de Data
Decimal	62	77/109	0	77/109	N/A	0
Hexadecimal	3E	4D/6D	0	4D/6D	N/A	0

En la figura 3.8-3, se puede observar el resultado de enviar dichos comandos mediante la interfaz implementada.

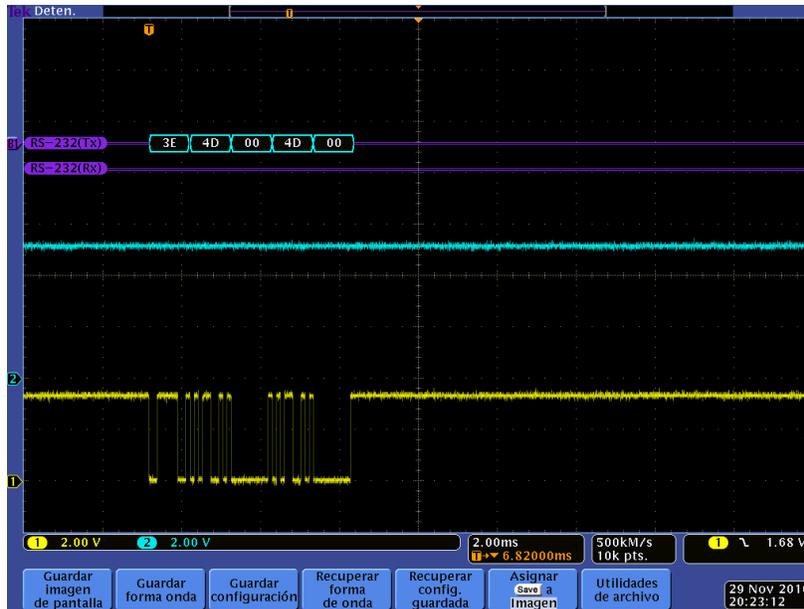


Figura 3.8-3 Captura de Osciloscopio con Mensaje Serial de Encender Motores. Extraído de Osciloscopio Tektronix MSO 4054

Por otro lado para probar el comando de control se utilizaron los siguientes valores:

- Modo: 3
- Velocidad Alabeo: 10°/s

- Velocidad Cabeceo:  $-50^\circ/s$
- Velocidad Guiñada:  $90^\circ/s$
- Ángulo Alabeo:  $20^\circ$
- Ángulo Cabeceo:  $-40^\circ$
- Ángulo Guiñada:  $-100^\circ$

Aplicando las fórmulas de la sección *CMD\_CONTROL*, así como el procedimiento para definir los bytes explicado en la misma sección se obtiene el mensaje descrito en las tablas 3.8-2 y 3.8-3.

Tabla 3.8-2 Información a Enviar para Comando Control de Controlador Simple BGC

Sección	Inicio ">"	ID Comando	Tamaño de Data	Suma de Encabezado	Data	Suma de Data
Decimal	62	67	13	80	Ver Tabla 8.9-3	48
Hexadecimal	3E	43	0D	50	Ver Tabla 8.9-3	30

Tabla 3.8-3 Datos para Comando Control de Prueba

Dato	Modo	V <sub>A-</sub>	V <sub>A+</sub>	α <sub>A-</sub>	α <sub>A+</sub>	V <sub>C-</sub>	V <sub>C+</sub>	α <sub>C-</sub>	α <sub>C+</sub>	V <sub>G-</sub>	V <sub>G+</sub>	α <sub>G-</sub>	α <sub>G+</sub>
		Bajo	Alto										
Dec	3	81	0	142	3	103	254	228	248	225	2	57	238
Hex	03	51	00	8E	03	67	FE	E4	F8	E1	02	39	EE

En la figura 3.8-4 se puede apreciar el resultado de la interfaz.

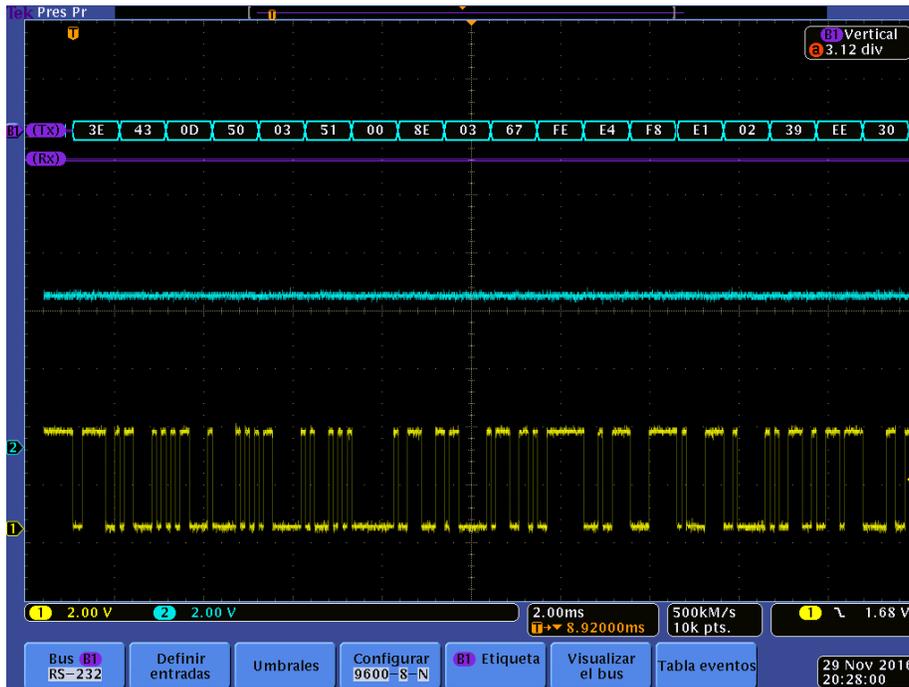


Figura 3.8-4 Captura de Osciloscopio con Mensaje Serial de Comando de Control. Extraído de Osciloscopio Tektronix MSO 4054

Como se observa en las figuras 3.8-3 y 3.8-4 la interfaz diseñada envía correctamente los bytes necesarios para operar el controlador simple BGC y por ende el Gimbal CAME Mini 3-air. Demostrado el operar adecuado de la aplicación se puede proceder a las pruebas sobre el gimbal.

### 3.8.3. Control SBGC en Gimbal CAME Mini 3-air

Como último punto previo a las pruebas se debe definir el método de conexión al controlador SBGC en el Gimbal, la figura 3.8-5 presenta un diagrama del controlador que permite observar las distintas opciones de conexión.

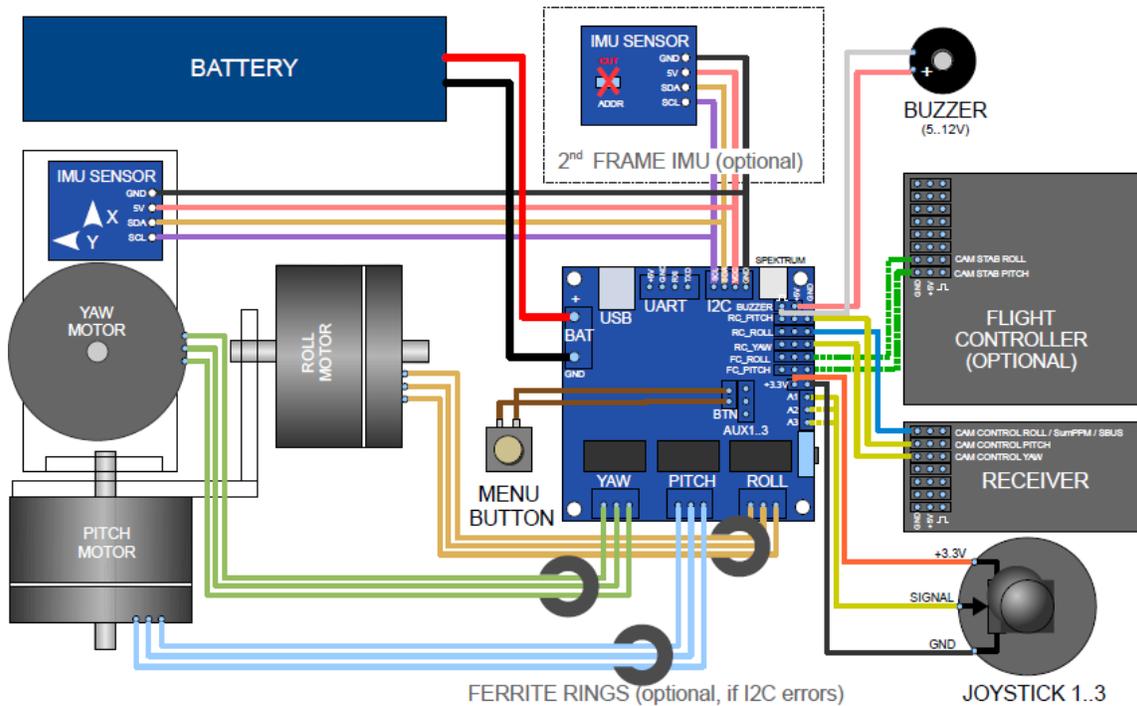
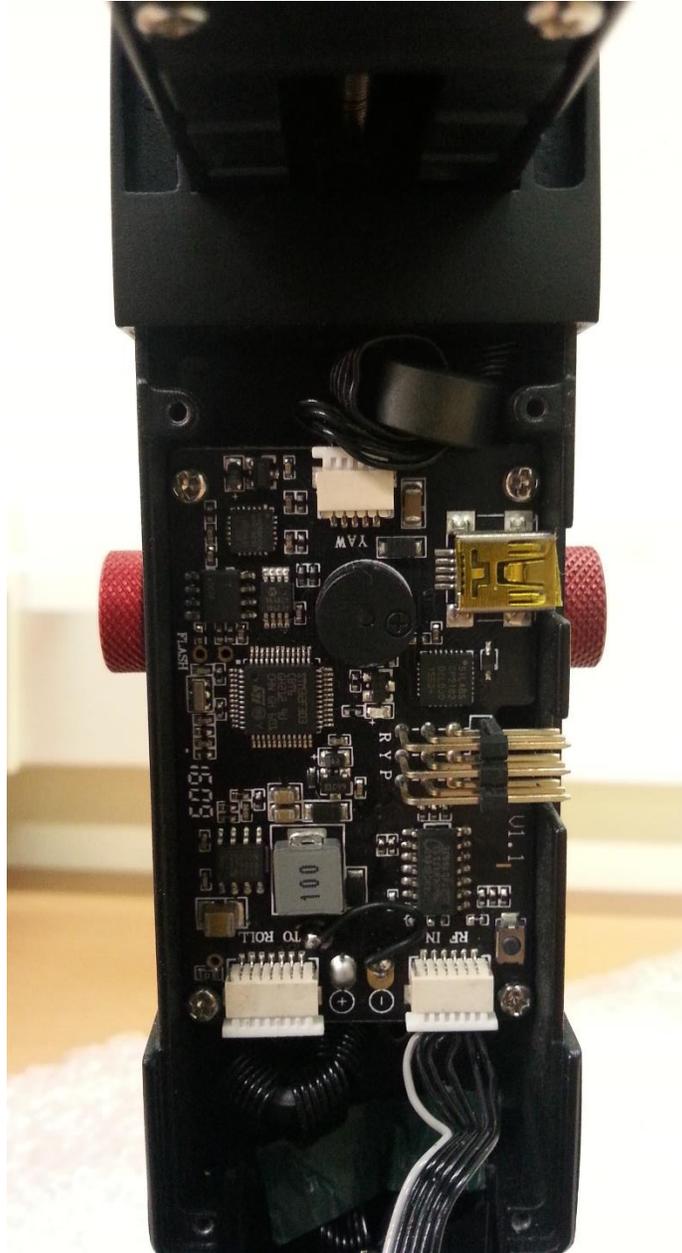


Figura 3.8-5 Diagrama de Conexión de Controlador SBGC. Extraído de (BaseCam Electronics, 2013)

La figura así como el propio manual del SBGC denotan dos opciones principales para la comunicación UART: El puerto respectivo o bien la configuración de los pines RC\_ROLL y RC\_YAW para la recepción de señal serial. Para el segundo caso es necesario realizar una configuración de propiedades en la GUI propia del SBGC, finalmente existe una tercera opción no definida propiamente en el manual pero si válida.

La interfaz del proveedor se vale del puerto USB para la comunicación y configuración del controlador, dicha conexión USB simplemente es un medio general para un tercer canal UART el cual se puede utilizar gracias a los drivers de puente USB-UART CP210 de Silicon Labs. Aprovechando este medio se puede facilitar la conexión entre el fitlet iA10 y el controlador del Gimbal.

La figura 3.8-6 muestra una fotografía del controlador SBGC contenido en el Gimbal CAME Mini 3-air.



*Figura 3.8-6 Placa SBGC en Gimbal CAME Mini 3-air. Tomado con cámara de Samsung S3*

Como se observa el modelo del controlador no es similar al presente en el manual, tampoco así a los distintos modelos comerciales del productor BaseCam, particularmente la placa de la figura 3.8-6 no presenta el puerto dedicado UART y la configuración de los pines R-Y y P no dio resultados positivos; considerando este factor y la presencia del puerto USB se decidió utilizar el tercer medio para la comunicación UART.

Propiamente el proceso de pruebas se basó en la prueba de cada comando de la interfaz programada, en orden se realizaron las siguientes labores.

1. Apagado de motores

2. Encendido de motores
3. Comando de control en modo de ángulo al eje cabeceo
4. Comando de control en modo velocidad al eje cabeceo
5. Repetición de pasos 3 y 4 para eje alabeo
6. Repetición de pasos 3 y 4 para eje guiñada
7. Comando de control en modo ángulo a los ejes cabeceo y alabeo

Las pruebas se realizaron de tal forma que se pudieran comprobar todas las posibles configuraciones en las que se podría utilizar el gimbal, cabe destacar que se dio prioridad en las pruebas a los pasos 1-4 ya que conforman las operaciones principales a realizar al momento de captura de imágenes, 1 y 2 por opciones de encendido y apagado de emergencia de motores, 3 y 4 por ser el eje encargado de la operación de barrido del pushbroom. Se rescatan de esta sesión de pruebas dos apreciaciones importantes:

- El control en modo ángulo permite variar la orientación con rapidez y sin esperas extras, esto a diferencia del trípode nexstar en el cual cada comando de movimiento implica una espera mínima de seis segundos en lo que el trípode asegura la posición; este detalle implica que el barrido de a pasos es viable en el gimbal.
- Similar al caso del trípode el controlador SBGC no menciona de forma directa cuando ha llegado a su destino, lo más similar a esto sería el uso de comandos que pidan la información en tiempo real; no obstante aún este proceso sufre de limitantes similares a las del trípode ya que el controlador SBGC requiere de 10 a 20 ms entre comando y comando.

Finalmente el proceso de pruebas fue exitoso, ninguna de las siete pruebas presento problema alguno y se definió que el programa, y por ende la biblioteca diseñada en el mismo, controlan correctamente el gimbal. El proceso de pruebas se repitió en el fitlet-iA10 obteniendo los mismos resultados.

### 3.9. Acople Físico de Gimbal a UAV y Cámara Hiperespectral

Crítico para la implementación en conjunto del gimbal, el UAV y las cámaras hiperespectrales, es el método de acople físico entre las mismas. Si bien este tema se tomó en cuenta en otros momentos, por ejemplo la selección del gimbal se vio fuertemente afectada por este factor, se presenta un desarrollo más a profundidad de este segmento del proyecto.

Por temas de orden y coherencia se puede separar el proceso de acople en dos partes: Unión Gimbal-UAV y Unión Gimbal-Cámara Hiperespectral.

#### 3.9.1. Unión Gimbal-UAV

La unión entre el gimbal y el UAV fue uno de los criterios de selección principal que permitió definir el Gimbal CAME Mini 3-air como el estabilizador a utilizar; propiamente los estabilizadores comerciales diseñados para uso en drones optocontrolados o similares cuentan con un acople moderadamente estándar. Entre otras cosas los acoples estándar cuentan con sistemas de reducción de

vibraciones así como una metodología de acople y distancias definidas. La figura 3.9-1 extraída del manual de usuario del gimbal CAME mini 3-air demuestra estas características.

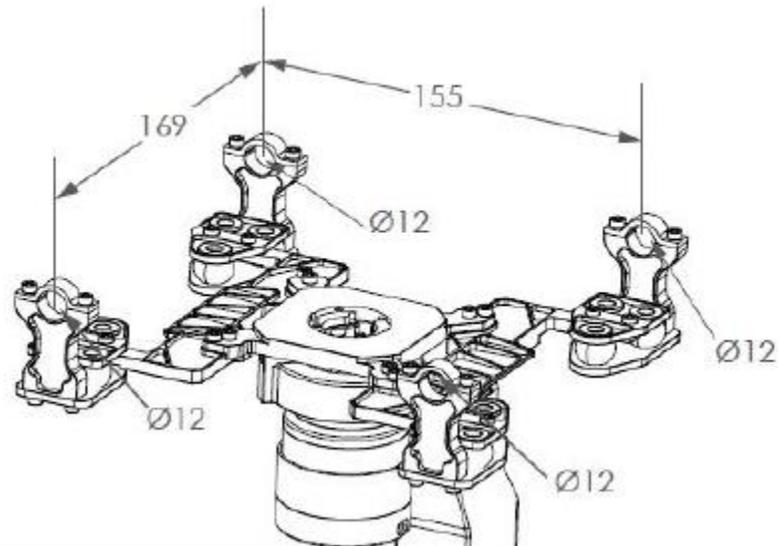


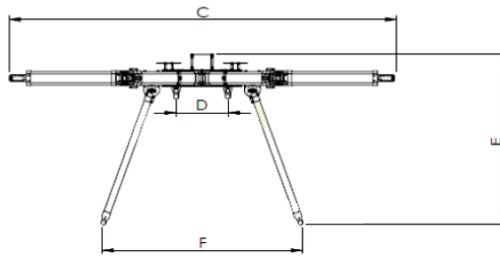
Figura 3.9-1 Acople Estándar Gimbal-UAV del CAME Mini 3-air. Extraído de (CAME-Tv, 2016)

La metodología de acople se vale de las cuatro sujeciones esquineras y de dos barras de diámetro definido, las cuales deben aportarse por el UAV en cuestión, para unir ambos elementos mediante presión de agarre entre sujeciones y barras. Importante de destacar es el diámetro de 12 mm y la separación centro a centro de 155 mm entre ambos espacios, estos dos valores conforman la parte central del estándar.

Por su cuenta el dron adquirido por el GII, marca Ixion, contempla la contraparte de dicha unión estándar, en las figuras 3.9-2 y 3.9-3 se presenta una imagen del dron a utilizar así como de la contraparte del acople.



Figura 3.9-2 Octocóptero marca Ixion. Extraído de (Ixion, 2016)



Modelo	Dimensión (mm)						Peso (g)
	A	B	C	D	E	F	
A-TEMPO	1200	459	1140	155	508	592	1820

Figura 3.9-3 Dimensiones de Octocóptero. Extraído de (Ixion, 2016)

La dimensión denominada por la letra D se refiere a la distancia centro a centro en las barras de acople del octocóptero al gimbal, se observa por manual que la dimensión cumple con el estándar facilitando la unión gimbal-UAV.

Finalmente como prueba del acople se presenta en la figura 3.9-4 una fotografía tomada en los laboratorios del GII donde se muestra el gimbal CAME mini 3-air unido al octocóptero de Ixion, por otro lado la figura 3.9-5 presenta con mayor detalle el punto de unión.



Figura 3.9-4 Octocóptero marca Ixion y Gimbal CAME Mini 3-air Acoplados. Tomado con Cámara Canon EOS 400D



Figura 3.9-5 Detalle de Acople Octocóptero marca Ixion y Gimbal CAME Mini 3-air. Tomado con Cámara Canon EOS 400D

### 3.9.2. Unión Gimbal-Cámara Hiperespectral

Para la unión entre el estabilizador y la cámara hiperespectral se dividió el proceso en dos partes: Análisis de la sujeción original y diseño de una pieza para solventar las limitantes de la sujeción original.

La figura 3.9-6 presenta el método original con el cual el estabilizador sujeta las cámaras tipo DSLR.

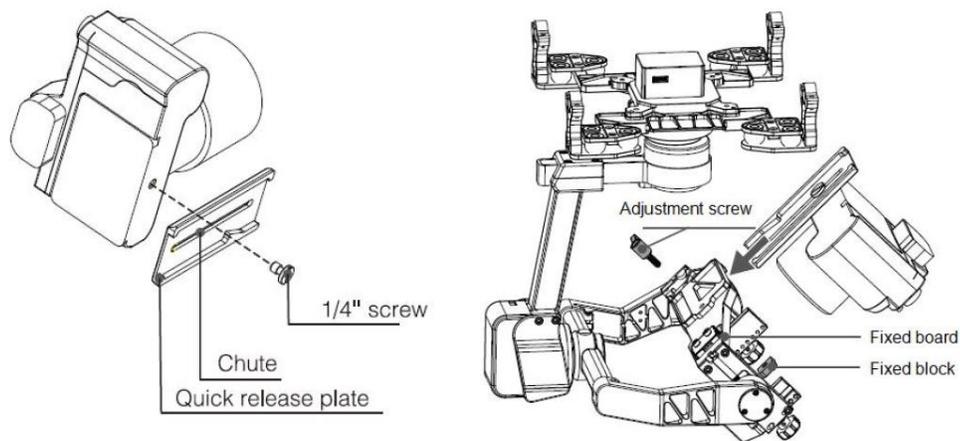
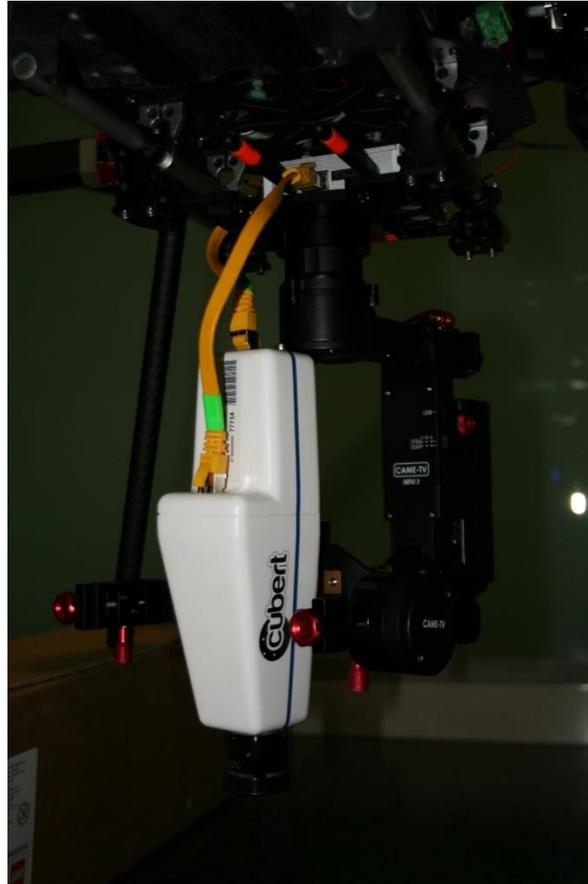


Figura 3.9-6 Diagrama de Unión CAME Mini 3-air con Cámaras tipo DSLR. Extraído de (CAME-Tv, 2016)

Se puede observar que la unión se basa en la placa de acople rápido y el uso de un tornillo de un cuarto de pulgada, inicialmente se propone utilizar el mismo método sin agregar ningún elemento extra, dado que los tornillos M5 pueden pasar la ranura de la placa de acople rápido no es necesario realizar modificaciones en este aspecto. La figura 3.9-7 presenta todo el conjunto octocóptero-gimbal-cámara unido de esta forma.



*Figura 3.9-7 Primera Iteración del Acople Gimbal-Cámara Hipereespectral. Tomado con Cámara Canon EOS 400D*

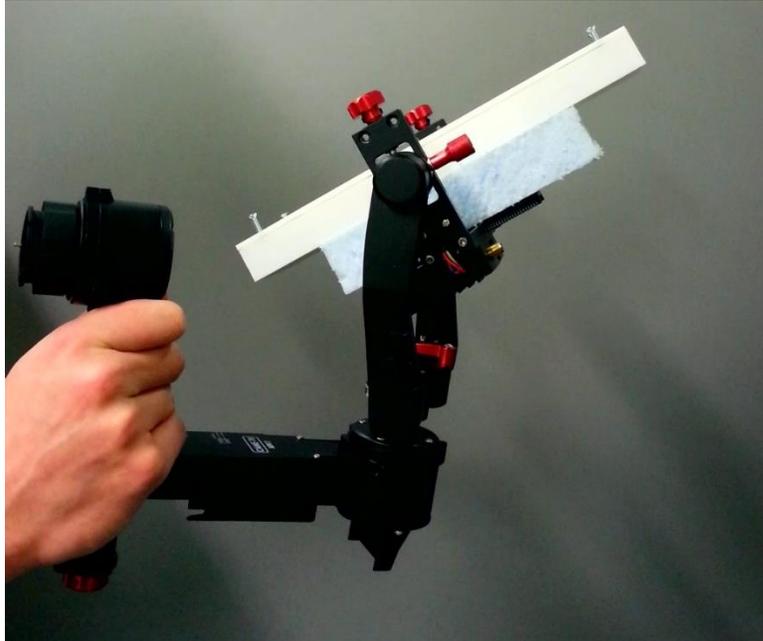
Se observa en la figura como el acople, sin ningún elemento extra, es una opción viable para poder unir ambos equipos; se debe destacar dos aspectos de la fotografía presente: Se agregaron los cables necesarios así como el fitlet-IA10 para representar de forma más exacta la unión al tiempo de operar y la imagen se tomó con el gimbal en funcionamiento, es decir orientando la cámara hacia el suelo.

Tal y como se presenta en la figura 3.9-7, es imposible balancear adecuadamente el hiperspectrómetro, dada la posición del motor del gimbal y las dimensiones del hiperspectrómetro. Si bien el gimbal es capaz de orientar la cámara aún con este problema su operación no es óptima y el peso no balanceado de la cámara genera una exigencia de torque extra sobre los motores que podría perjudicar los mismos. Por esto se decidió diseñar una pieza capaz de resolver este problema, los requisitos propios de la pieza serían:

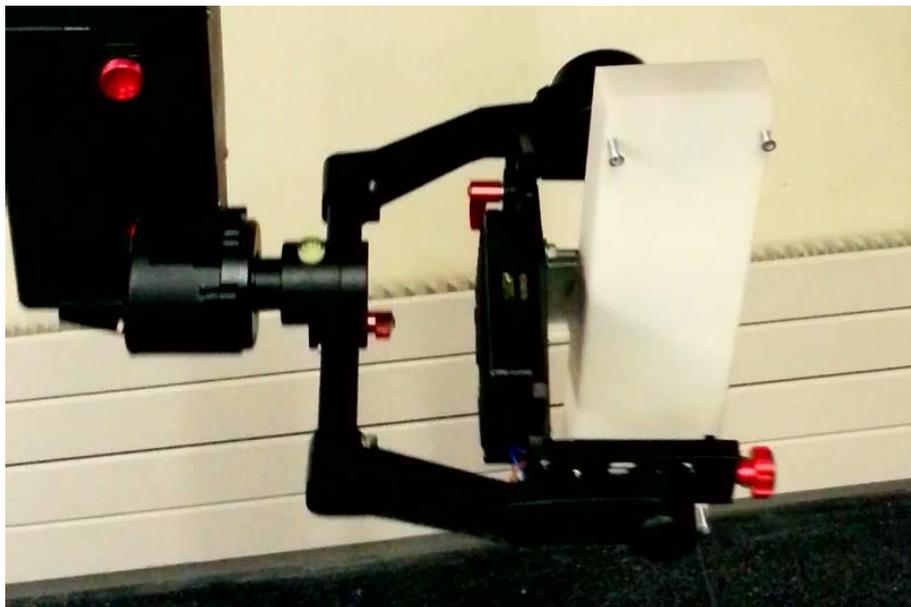
- Ser capaz de unirse a la placa de acople rápido

- Elevar la altura del hiperspectrómetro, permitiendo así el balanceo en el eje de cabeceo
- Desplazar el hiperspectrómetro de forma lateral, permitiendo así el balanceo del eje alabeo.

Para poder aproximar las dimensiones del acople se realizaron prototipos rápidos con materiales dispersos localizado en el laboratorio del GII, las figuras 3.9-8 y 3.9-9 presentan los dos prototipos.



*Figura 3.9-8 Prototipo para Balanceo de Eje Cabeceo. Tomado con cámara de Samsung S3*



*Figura 3.9-9 Prototipo para Balanceo de Ejes Cabeceo y Alabeo. Tomado con cámara de Samsung S3*

Se debe aclarar, que por temas de protección de las cámaras, los prototipos y pruebas se trabajaron con una réplica diseñada por estudiantes de la UDC, la réplica imita las dimensiones y peso de la cámara UI3360-CP tipo pushbroom al ser la más pesada y grande. En el caso del primer prototipo se

logró definir que un cambio en la altura de 3 cm permitía balancear el eje pitch, por otro lado el segundo prototipo demuestra como un desplazamiento lateral entre 2.5 y 3 cm permite el balanceo sobre el eje alabeo.

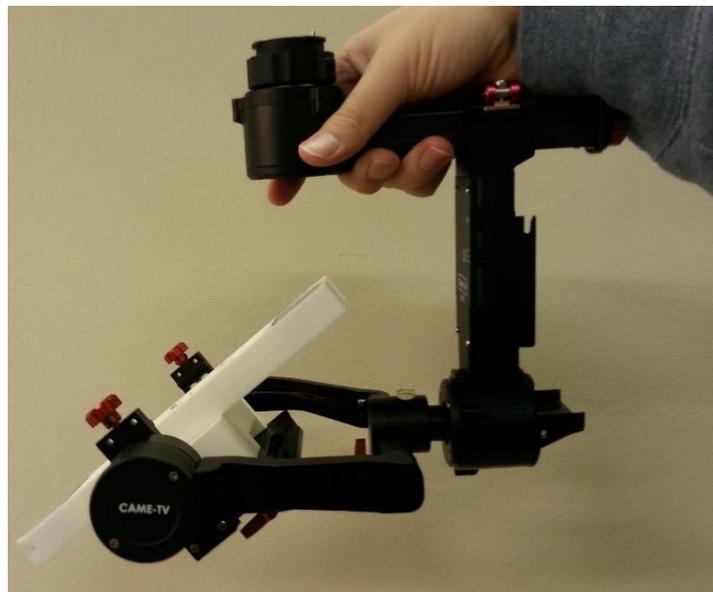
En consciencia de la información de los prototipos se diseñó una pieza para impresión 3D que pudiera fijar la cámara de forma más estable, dicha pieza se diseñó en la versión gratuita del software libre Onshape, la figura 3.9-10 presenta el plano de la pieza impresa.



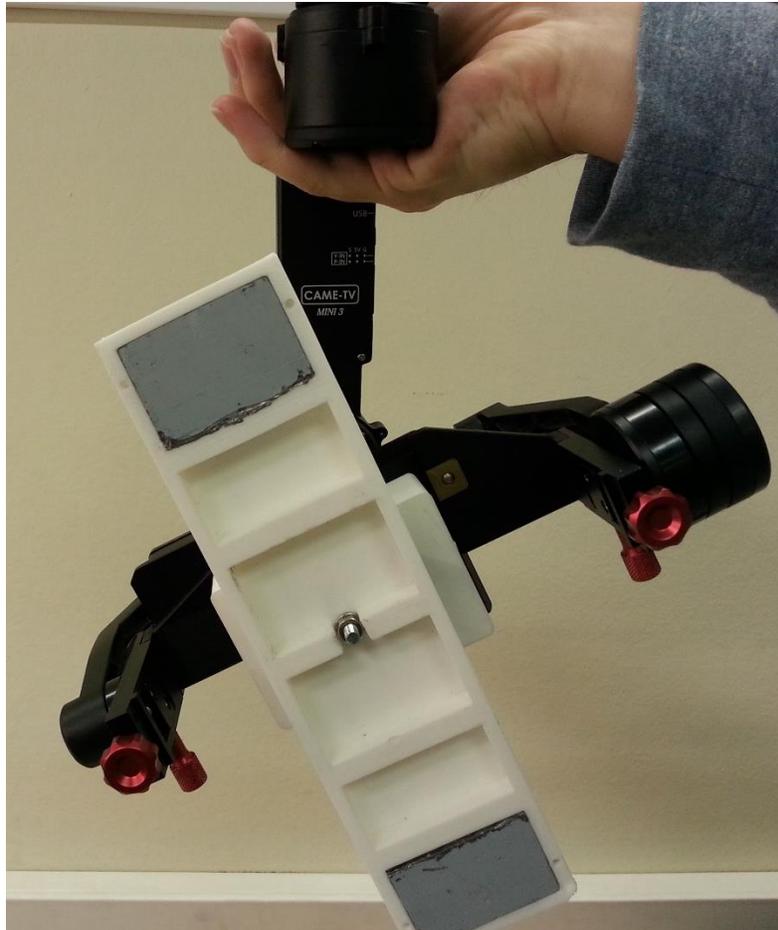
Las dimensiones de 30 mm de altura y 80 mm de largo se definieron gracias a los prototipos realizados. La unión a la placa de acople rápido se realiza mediante los dos agujeros de 4.7 mm de diámetro, esto en consideración de los tornillos M5 a utilizar. En la figura 3.9-11 se puede apreciar una fotografía de la pieza diseñada una vez impresa, seguidamente en las figuras 3.9-12 y 3.9-13 se presenta la réplica y el gimbal unidos por la pieza, las posiciones demuestran que el proceso de balanceo si se puede realizar adecuadamente.



*Figura 3.9-11 Acople de Gimbal y Cámara Impreso en 3D. Tomado con cámara de Samsung S3*



*Figura 3.9-12 Prueba de Balanceo en Eje Cabeceo con Placa de Ajuste. Tomado con cámara de Samsung S3*



*Figura 3.9-13 Prueba de Balanceo en Eje Alabeo con Placa de Ajuste. Tomado con cámara de Samsung S3*

En ambas imágenes se presentan posiciones tales que, en caso de haber un mal balance, el peso propio de la réplica provocaría que el gimbal cayera a una posición horizontal y vertical respectivamente. Al no darse este efecto se puede afirmar que el proceso de balanceo se logra satisfactoriamente y la placa impresa en 3D cumple su propósito.

### 3.10. Bibliotecas de Control Puhsbroom, Snapshot y Gimbal Came Mini 3-air

En vista de los trabajos futuros, así como las posibles necesidades de acceder, reutilizar o expandir las aplicaciones y programas realizados, se documentó y formalizó los programas realizados al punto de reducirlo en tres bibliotecas principales. Se buscó con esta labor dejar una base de la cual se podrían desarrollar futuras aplicaciones, o comprender el código de las aplicaciones existentes y facilitar el proceso de comprensión y puesta en marcha de los equipos.

Formalmente se redujo el trabajo a tres bibliotecas:

- **Control\_Snapshot\_GII.h:** Para el control de la cámara tipo snapshot, contempla métodos de comunicación TCP/IP y funciones para él envío de comandos, todo según lo demostrado en la sección 3.4.
- **Control\_SBGC\_GII.h:** Para la operación del gimbal, o más específicamente el controlador SBGC utilizado por el gimbal adquirido. Efectivamente la biblioteca se puede utilizar en otros modelos de gimbal distintos al CAME Mini 3-air. Todo según lo visto en la sección 3.9.
- **Control\_Pushbroom\_UI3360\_CP\_GII.h:** Biblioteca para el control de la cámara tipo pushbroom, contempla las lógicas de punteros utilizada para el manejo de múltiples espacios de memoria RAM, así como las modificaciones necesarias para su viabilidad en programación por hilos. Diseñada a partir de las labores descritas en las secciones 3.3 y 3.8.

Propiamente la documentación de cada biblioteca se divide entre tres o cuatro secciones principales:

- **Descripción:** Descripción general de la biblioteca, menciona las funcionalidades principales que aporta así como un breve resumen de lo que se puede lograr con la biblioteca.
- **Requisitos y Consideraciones:** Listado de todos los requisitos y cuidados que se deben tener al desear usar la biblioteca. Temas de drivers necesarios, dependencias, versión sobre la cual se diseñó la biblioteca y demás notas similares se pueden encontrar en esta sección.
- **Explicación de Estructuras:** En caso de que la biblioteca contenga estructuras se presentan cada una y se explican.
- **Explicación de Funciones:** Similar a la sección de explicación de estructuras pero para la descripción de las funciones.

Para su entrega, además del manual y la biblioteca, se adjuntaron todas las dependencias necesarias para poder utilizar la biblioteca, sea drivers, bibliotecas necesarias y otros manuales afines. En los apéndice 3, 4 y 5 se pueden observar los manuales de las bibliotecas para la cámara pushbroom, snapshot y el controlador SBGC respectivamente.

### 3.11. Programa Final de Pruebas

Se desarrolló un último programa que pudiera ser capaz de controlar tanto el movimiento del Gimbal como la captura de imágenes de la cámara tipo pushbroom. Efectivamente se programó el equivalente a la interfaz de la sección 3.3 *Configuración de Cámara Hiperespectral UI3360-CP para Operación en Trípode Nexstar*, pero utilizando el estabilizador en vez del trípode y con una interfaz de win32 y no Windows forms.

El programa utiliza las bibliotecas creadas a lo largo del proyecto para controlar tanto el hiperespectrometro tipo pushbroom como el estabilizador CAME Mini 3-air e implementa una secuencia de barrido basado en el estilo “Movimiento Continuo” descrito en la figura 3.3-3 como medio para la captura de imágenes. En la figura 3.11-1 se puede observar una imagen de la interfaz diseñada

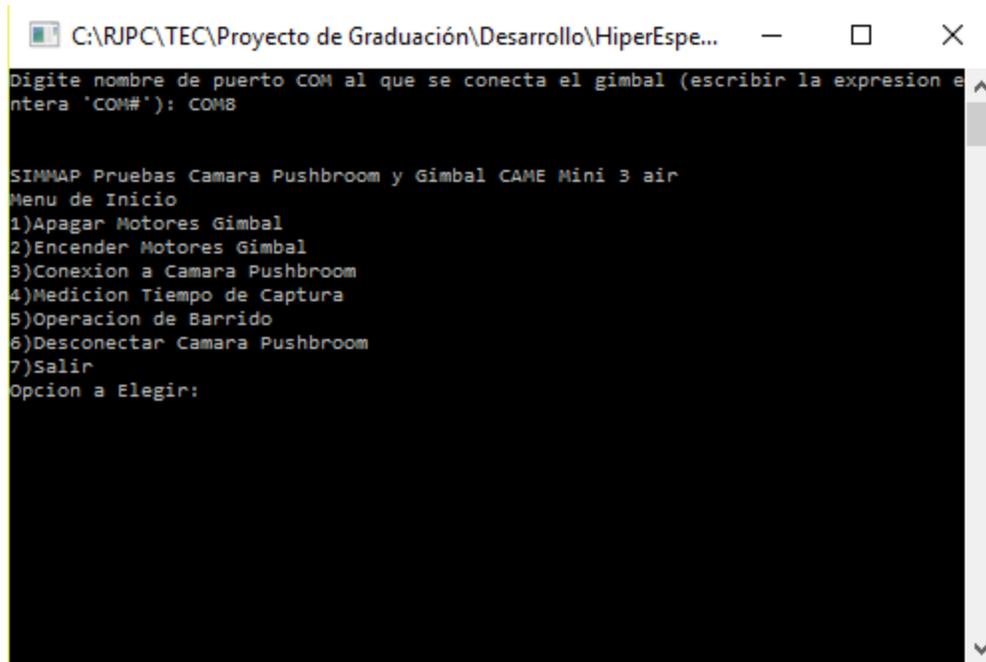


Figura 3.11-1 Interfaz Win32 de Programa Final para Control de Gimbal y Pushbroom

Es necesario destacar como la interfaz solicita inicialmente al usuario el nombre del puerto COM al cual se conecta el controlador SBGC del gimbal. Se describe a continuación la funcionalidad de cada opción del menú:

- Apagar/Encender Motores Gimbal: Apagan o encienden los motores del gimbal según corresponde
- Conexión a Cámara Pushbroom: Permite iniciar la cámara hiperespectral tipo pushbroom, asigna además el tiempo de exposición a 83 ms.
- Medición Tiempo de Captura: Realiza una captura de prueba para medir el tiempo real de adquisición de imagen, solo se puede utilizar esta opción si la cámara se encuentra conectada.
- Operación de Barrido: Solicita información de formato de imagen, cantidad de líneas y ángulo de barrido al usuario para luego realizar una operación de barrido por movimiento continuo. Esta labor solo se puede realizar si se ha calculado el tiempo de adquisición de imagen, es decir haber usado la opción cuatro del menú; esto ya que se utiliza la misma fórmula para calcular la velocidad de barrido que en el caso de la sección 3.3.

$$V [ \text{°/s} ] = \frac{\text{angulo} [ \text{°} ]}{T_{\text{captura}} [ \text{s} ] * C_{\text{Lineas}}} \quad (3-11)$$

- Desconectar Camara Pushbroom: Operación contraria a la opción tres del menú
- Salir: Fin de programa.

Para poder comprender a más detalle las operaciones realizadas en el programa, así como las dependencias entre opciones, se puede observar la figura 3.11-2 en el cual se presenta a modo de diagrama de flujo el programa diseñado.



Finalmente para poder apreciar con más detalle la opción “Operación del Barrido” del programa desarrollado, se presenta en la figura 3.11-3 un diagrama de flujo del proceso de captura de barrido, en el diagrama se hace mención de las funciones específicas de las bibliotecas Control\_SBGC\_GII.h y Control\_Pushbroom\_UI3360\_CP\_GII.h en caso de que se quiera seguir el proceso desde un punto de vista más detallado.

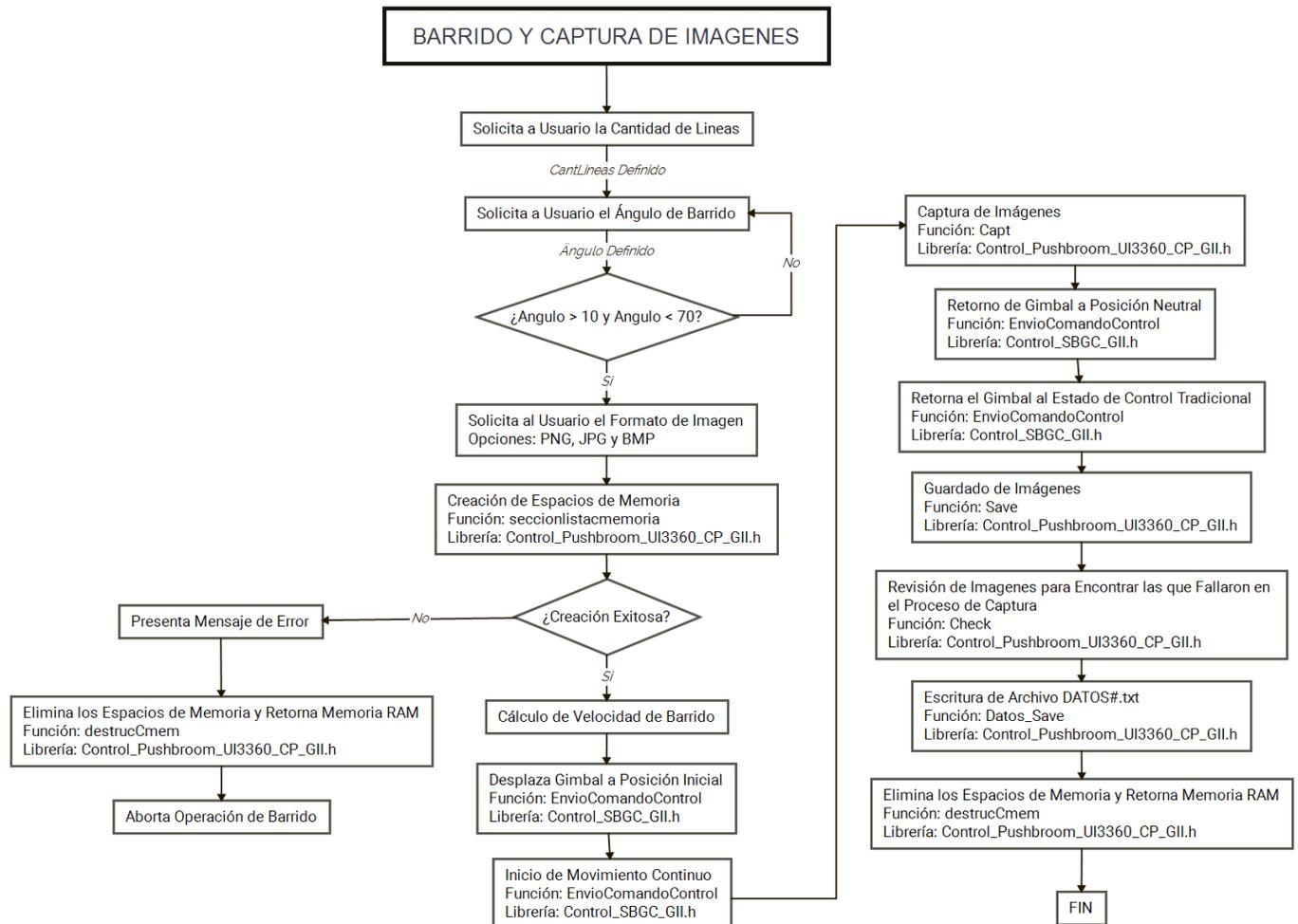


Figura 3.11-3 Secuencia de Operación de Barrido en Programa Final

Se debe aclarar que el valor de ángulo dado asume que la posición original, apuntar perpendicular al suelo, es el centro del barrido. Por esta razón el paso “Desplaza Gimbal a Posición Inicial” se encarga de acomodar la orientación de la cámara a una rotación de  $\text{Ángulo}/2$  previo al inicio del barrido. La limitante de ángulo entre 10 y 70 grados se pone solo por temas de cuidado del equipo, en la realidad el rango máximo puede llegar ligeramente menos de 160 grados, esto por temas de que la cámara no se puede colocar horizontalmente en el gimbal ya que chocaría con la estructura.

Por otro lado se destaca el cuidado en la generación adecuada de la cantidad de memorias, el fallo se da cuando no se logra generar todas las memorias deseadas; no obstante, si se asignan todas las

memorias previas a la que genera el fallo. Es por esta razón que en caso de fallar se eliminan las memorias que si se lograron generar para evitar el mal uso de la memoria RAM.

Queda explicado el programa de pruebas final para el control de la cámara pushbroom en conjunto con el gimbal CAME Mini 3-air. Las pruebas de barrido no se pudieron realizar para la fecha de finalización dado que la cámara tipo pushbroom no había sido modificada con la base para atornillar con métrica M5; no obstante, los éxitos de la lógica de barrido vistos en la sección 3.3 así como las pruebas parciales del control del gimbal y la cámara tipo pushbroom sirven de prueba para verificar funcionalmente el proceso de captura.

De esta forma se da por finalizando el desarrollo de las distintas partes del proyecto.

## 4. Verificación de Objetivos

En el presente capítulo se hace una recopilación sobre las labores y el estado de los objetivos desarrollado a lo largo del proyecto; buscando demostrar cuales objetivos se cumplieron y cuales objetivos no se cumplieron. En caso de ser necesario, si un objetivo no se cumplió se detalla el sentido en que este queda inconcluso y los motivos del mismo.

### 4.1. Selección de Estabilizador Comercial

En la sección 3.2 se detalló debidamente el proceso de selección del estabilizador comercial, empezando por una aclaración de los criterios de selección; seguidamente se diseñó un perfil inicial, más no excluyente, de búsqueda con el cual se perfiló el proceso de investigación. Como resultado se obtuvieron las tablas 3.2-1, 3.2-2, 3.2-3, 3.2-4 y 3.2-5 en las cuales se detallan la información de 13 modelos comerciales completas, cuatro estructuras DIY y tres controladores de gimbal.

Finalmente se seleccionó el modelo de gimbal, a lo largo de una reunión con los demás encargados del proyecto, en la cual se terminaron de perfilar los criterios más importantes de selección; entre estos algunos criterios excluyentes como la capacidad de controlar tres ejes de movimiento o la capacidad de realizar el control por orientación; también se definieron los criterios más relevantes de comparación como lo fueron la presencia de elementos físicos que faciliten el acople UAV con gimbal, la precisión del control y el peso.

Todo el trabajo dio con la selección del gimbal CAME-Mini 3-air el cual reúne de forma general las mejores características, entre los estabilizadores investigados, para el proyecto SIMMAP. Queda claro en este sentido que el objetivo se cumplió en su totalidad tanto el proceso de desarrollo como los entregables finales.

### 4.2. Sistema de Fijación entre Cámara Hiperespectral Gimbal y UAV

El sistema de fijación se desarrolló a lo largo de las secciones 3.2.5 y 3.9, abarcó no solo la unión estabilizador-cámara sino también la unión estabilizador-UAV. Propiamente en la sección 3.2.5, de selección de gimbal, uno de los criterios prioritarios de selección fue la presencia de elementos para facilitar la adaptación del gimbal al dron, mostrando así consideración del objetivo al momento de seleccionar el gimbal.

En la sección 3.9 se demuestra la viabilidad de los criterios de elección de gimbal al demostrar el acople estándar tanto con información de manual así como con fotos del resultado final. Por otro lado se presentó dos métodos de acopla la cámara hiperespectral al UAV, directamente por la placa de ajuste rápido y mediante una pieza impresa en 3D, diseñada a partir de pruebas con prototipos económicos del laboratorio. El segundo método logró asegurar no solo la sujeción sino también la posibilidad de balancear correctamente la cámara hiperespectral en el estabilizador.

El trabajo dio con planos únicamente del acople gimbal-cámara al ser los únicos necesarios y con un sistema de estabilizador-UAV-cámara integrado físicamente. De esta forma se considera que los entregables según los objetivos se cumplieron adecuadamente.

### 4.3. Control de Estabilizador Comercial y Cámara Hiperespectral

El control de las cámaras hiperespectrales se desarrolló a lo largo de distintas secciones del trabajo, iniciando con la sección 3.3 se comprende y opera la biblioteca original de la cámara UI3360-CP del hiperespectrómetro tipo pushbroom, llegando a implementar la misma en un trípode a manera de prueba de concepto del barrido; la sección 3.4 describe un proceso similar para el hiperespectrómetro UHD-185 tipo snapshot.

La sección 3.5 describe un proceso de análisis y verificación de las imágenes obtenidas con el control desarrollado, en la sección 3.6 se definió y justificó la elección del hardware a operar sobre el octocóptero definiendo el fitlet iA10 como controlador seleccionado. La sección 3.7 describe un proceso de mejora del trabajo realizado en la sección 3.3 donde se mejoró el tiempo y eficiencia de la captura de imágenes, la sección 3.8 elabora sobre el control y manejo del estabilizador comercial utilizado; finalmente la sección 3.10 detalla la documentación desarrollada para labores futuras.

El trabajo realizado permitió la programación de una variedad de aplicaciones para la captura de imágenes hiperespectrales, el almacenamiento de las mismas así como el control del estabilizador comercial; por otro lado se generaron varias bibliotecas en archivos .h, junto a sus manuales respectivos, de tal forma que el GII pudiese tener acceso al código de control y generar sus aplicaciones a futuro; finalmente se programaron scripts .m de Matlab para la presentación y generación de cubos de datos. De esta forma se considera que los entregables según los objetivos se cumplieron.

### 4.4. Verificación del Proyecto

El proceso de verificación se dio a lo largo del proyecto paralelo al desarrollo del mismo.

En cuanto a la elección del gimbal su verificación se da parcialmente en las secciones 3.8, 3.9. La sección 3.8 describe el proceso de control del gimbal, es bajo este proceso que se asegura que el gimbal es capaz de realizar las operaciones de barrido necesarias para la captura de cubo de datos con la cámara tipo pushbroom. Por otro lado la sección 3.9 demuestra como la elección del gimbal facilitó el acople Gimbal-UAV, además de que las dimensiones internas fueron las adecuadas para soportar las cámaras hiperespectrales. Finalmente la estabilidad del gimbal se observó contra las replica, en protección de las cámaras hiperespectrales.

En la sección 3.9 se presenta la verificación de los acoples Gimbal-Cámara y Gimbal-UAV. Específicamente en la sección 3.9.1 se demuestra, mediante las figuras 3.9-4 y 3.9-5, la unión exitosa entre UAV y Gimbal producto de la selección del estabilizador. Por otro lado se verifica el funcionamiento adecuado del acople Gimbal-Cámara a lo largo de la sección 3.9.2, se demostró que el acople no solo logra

sujetar físicamente la cámara sino también facilitar el proceso de balanceo. Las figuras 3.9-12 y 3.9-13 son pruebas de los resultados obtenidos.

El control de la cámara tipo pushbroom y la lógica de barrido se verifican en la sección 3.3 a través de la interfaz diseñada y los resultados de la misma, la figura 3.3-6 provee un resultado positivo de dicha labor. A su vez las sección 3.5 son en sí mismas verificaciones del control de la cámara tipo pushbroom y la lógica de barrido implementada; la posibilidad de comparar el modelo comercial tipo snapshot con el tipo pushbroom y el operar sobre la información provista por el barrido solo se podría hacer si en primer lugar la captura de cubos de datos se diera de una forma aceptable.

Finalmente se entregó el gimbal CAME Mini 3-air con los acoples necesarios para unir la cámara hiperespectral y un último programa desarrollado en la sección 3.11, en conjunto ambos conforman el prototipo final para pruebas. Cabe destacar que no se pudieron realizar pruebas con el UAV en aire ya que, dada la fecha de desarrollo, las condiciones climáticas de la zona impedían la realización de pruebas seguras.

Es necesario aclarar que no se pudo realizar pruebas de barrido directamente con la cámara hiperespectral, ya que las modificaciones propuestas para la misma no fueron realizadas para la fecha del fin del proyecto. Estas modificaciones implican el cambio de la base de aluminio a un plástico más liviano y la adición de una rosca métrica M5 en la base para su unión al acople diseñado. Formalmente la única forma en que se verificó el programa final fue con la revisión simultánea de operación de barrido y captura de imágenes.

Por otro lado no se realizaron pruebas en aire ya que las condiciones climáticas para el final del desarrollo no permitían volar de forma segura el octocóptero.

En este sentido se define que se cumplió el cuarto objetivo en la medida posible dado el tiempo de desarrollo del proyecto, dejando como entregables el prototipo final para pruebas y el informe como prueba de la funcionalidad de los distintos elementos del prototipo.

## 5. Conclusiones

- Se realizó una búsqueda de 13 modelos comerciales de gimbal, cuatro estructuras DIY y tres controladores para realizar la selección de un gimbal, la misma se desarrolló tomando en cuenta criterios dados por una investigación previa sobre gimbals y las necesidades del GII, entre los cuales destacan el peso, la calidad de estabilización y la facilidad de acople a un UAV.
- Se seleccionó el gimbal CAME Mini 3-air para el desarrollo futuro del proyecto SIMMAP, ya que contempla un medio adecuado entre peso, calidad de estabilización y tamaño; además contempla una unión estándar con UAV.
- Se seleccionó el Mini-PC fitlet iA10 para el manejo del estabilizador y las cámaras hiperspectrales, dada su capacidad de operar el hiperspectrometro snapshot, su compatibilidad con los programas C++ generados, la reducción de costos y la capacidad de conexión mediante wifi.
- Se desarrollaron aplicaciones y bibliotecas en el lenguaje C++ para el control y operación de cámaras hiperspectrales tipo pushbroom y snapshot así como para el estabilizador comercial controlado; entre otras labores las bibliotecas contemplan operaciones de manejo de características, captura de información y guardado estructurado de la misma.
- Se acoplo un estabilizador inercial comercial a un UAV mediante sujeciones estándar y se integró el mismo estabilizador a las cámaras hiperspectrales UI3360-CP y UHD-185 mediante el diseño de una pieza impresa en 3D, la pieza vela no solo por la sujeción mecánica sino también por el posicionamiento adecuado de la cámara para las labores de balanceo de gimbal.
- Se realizaron pruebas de forma individual de las secciones del proyecto, haciendo énfasis en la calidad del control y operación de barrido de la cámara hiperspectral, esto al analizar las capturas múltiples veces mediante scripts de Matlab encargados de presentar la información hiperspectral de un píxel o un área de la imagen.
- Se entregó un prototipo final de pruebas consistente en el gimbal CAME Mini 3-air con las adiciones necesarias para su acople y una aplicación general de pruebas desarrollada en C++. Además las bibliotecas y su documentación permiten el desarrollo a futuro de nuevas aplicaciones y modificaciones sin mayor complejidad.

## 6. Recomendaciones y Trabajos Futuros

- En caso de vibraciones indeseadas en el gimbal se recomienda variar los valores de “Power”, que definen la potencia alcanzable por los motores, de la interfaz del SBGC dada por el proveedor previo a alterar los valores PID.
- Se debe asegurar que iniciar el programa C-Link sea parte de la rutina de arranque del Fitlet-iA10 para que así la biblioteca Control\_Snapshot\_GII.h pueda trabajar inmediatamente después de encender el Mini-PC. Evitando así problemas de acceso a la interfaz del Mini-PC en tiempo de vuelo.
- Es recomendable sustituir las piezas más pesadas de la cámara tipo pushbroom, así como del gimbal CAME Mini 3-air, con materiales más liviano; esto mejoraría el tiempo de vuelo del UAV.
- La alimentación del gimbal CAME Mini 3-air depende de la batería provista por el proveedor, sustituir esta alimentación por una fuente más liviana, inclusive la misma fuente del UAV, reduciría parte del peso; no obstante esto implica que el balanceo del eje guiñada se debe corregir.
- Cuando se realizan capturas se debe tener cuidado de asignar adecuadamente la orientación inicial del estabilizador, caso contrario al centrar la cámara después de un barrido la orientación de la cámara será distinta a la inicial.
- Se recomienda analizar el rendimiento del sistema empotrado con métodos más complejos y apropiados como el uso de benchmarks o similares.
- Como parte del proceso de optimización se debe plantear el desarrollo de un sistema operativo de tamaño mínimo, buscando aprovechar al máximo los recursos del sistema empotrado
- Se debe buscar una segunda aproximación a la lógica por hilos, utilizando en este caso la biblioteca Open MP y no la biblioteca pthreads.

## 7. Referencias Bibliográficas

- Aasen, H., Burkart, A., Bolten, A., & Bareth, G. (2015). Generating 3D hyperspectral information with lightweight UAV snapshot cameras for vegetation monitoring: From camera calibration to quality assurance. *ISPRS Journal of Photogrammetry and Remote Sensing*, 245-259.
- Abdo, M., Toloei, A., Vali, A., & Arvan, M. (2014). Modeling, Control and Simulation of Cascade Control Servo System for One Axis Gimbal Mechanism. *International Journal of Engineering*, 157-169.
- Align. (18 de Abril de 2015). G3-5D Instruction Manual. *[En Línea]*. [fecha de consulta: 7 Setiembre - 15 Octubre 2016], Disponible en: <<http://www.align.com.tw/ManuaLib/Multi/Gimbal/RGG302.pdf>>.
- Align. (2015). *G3-5D Gimbal*. Obtenido de Align Conquer Your Heart: <http://www.align.com.tw/aerial-equip-en/multicopter/g35d/>
- Axis. (s.f.). *Brushless Gimbal + Joystick*. Obtenido de Axis: <http://axisgimbal.com/3-axis-gimbal/brushless-gimbal/>
- Axis. (s.f.). *Brushless Gimbal Startup Bundle*. Obtenido de Axis: <http://axisgimbal.com/3-axis-gimbal/dslr-3-axis-brushless-gimbal-startup-bundle/>
- B&H Photo-Video-ProAudio. (2016). *ikan DS1 Beholder Gimbal for Select DSLRs and Mirrorless*. Obtenido de B&H: [https://www.bhphotovideo.com/c/product/1200662-REG/ikan\\_ds1\\_beholder\\_gimbal\\_for.html](https://www.bhphotovideo.com/c/product/1200662-REG/ikan_ds1_beholder_gimbal_for.html)
- Barton, J. (2012). Fundamentals of Small Unmanned Aircraft Flight. *Johns Hopkins APL Technical Digest*, 132-149.
- BaseCam Electronics. (2013). SimpleBGC 32bit 3-Axis Software User Manual. *[En Línea]*. [fecha de consulta: 14-17 noviembre 2016], Disponible en: <<http://www.basecamelectronics.com/>>.
- BaseCam Electronics. (2016). *BaseCam SimpleBGC 32-bit*. Obtenido de BaseCam Electronics: <http://www.basecamelectronics.com/simplebgc32bit/>
- BaseCam Electronics. (9 de Setiembre de 2016). SimpleBGC 2.5 Serial Protocol Specification. *[En Línea]*. [fecha de consulta: 14 - 17 Noviembre 2016], Disponible en: <<http://www.basecamelectronics.com/>>.
- BaseCam Electronics. (9 de Agosto de 2016). *SimpleBGC Serial API Examples and Libraries*. Obtenido de Github: <https://github.com/alexmos/sbgc-api-examples>
- Brake, N. (Agosto de 2012). Control System Development for Small UAV Gimbal. *(Tesis de Grado)*. Faculty of California Polytechnic State University, California, USA.
- Budimir, M. (10 de Enero de 2012). *What is a Servomotor?* Obtenido de <https://www.youtube.com/watch?v=2t8UnHy1buQ>
- CAME-Tv. (2016). *CAME-Argo*. Obtenido de CAME-TV: <http://www.came-tv.com/cameargo-3-axis-gimbal-camera-32bit-boards-with-encoders-p-778.html?zenid=r93le2ejf21gapn988smj7feh4>

- CAME-Tv. (2016). CAME-Argo User Manual 1. *[En Linea]*. [fecha de consulta: 7 setiembre - 15 octubre 2016], Disponible en: <<http://cametvblog.com/tag/user-manual/>>.
- CAME-Tv. (2016). CAME-Argo User Manual 2. *[En Linea]*. [fecha de consulta: 7 setiembre - 15 octubre 2016], Disponible en: <<http://cametvblog.com/tag/user-manual/>>.
- CAME-Tv. (2016). CAME-Mini 3 User Manual V1.0. *[En Linea]*. [fecha de consulta: 7 setiembre - 15 octubre 2016], Disponible en: <<http://cametvblog.com/tag/user-manual/>>.
- CAME-Tv. (2016). *CAME-Mini 3/2/1*. Obtenido de CAME-TV: <http://www.came-tv.com/camemini-3-3axis-gimbal-camera-32bit-boards-with-encoders-p-810.html>
- CAME-Tv. (2016). CAME-Tv Mini 3 Air User Manual V1.0. *[En Linea]*. [fecha de consulta: 7 setiembre - 15 octubre 2016], Disponible en: <<http://cametvblog.com/tag/user-manual/>>.
- Celestron. (2008). NexStar Communication Protocol. *[En Linea]*. [fecah de consulta: 28 Setiembre - 18 Octubre 2016], Disponible en: <[http://www.celestron.com/media/795779/1154108406\\_nexstarcommprot.pdf](http://www.celestron.com/media/795779/1154108406_nexstarcommprot.pdf)>.
- Chatfield, A. (2000). *Fundamentals of High Accuracy Inertial Navigation*. American Institute of Aeronautics and Astronautics.
- CheesyCam [MrCheesyCam]. (28 de Febrero de 2016). CAME-TV ARGO 32bit SBGC Gimbal Stabilizer with Encoder Hardware Product Overview. *[Archivo de Video]*. Obtenido de Youtube: <https://www.youtube.com/watch?v=tGK3y7z1VhQ>
- CheesyCam [MrCheesyCam]. (27 de Marzo de 2016). CAME-TV Mini 3 32 bit SBGC Gimbal with new Encoders Product Overview. *[Archivo de Video]*. Obtenido de YouTube: <https://www.youtube.com/watch?v=f6CtE8XH5u0>
- CopterFrames. (2015). *TwoDee EVO NG 3x 5208 Gimbal Bruhsless Motor*. Obtenido de CopterFrames: <http://copterframes.com/motor-prop-esc/49-1.html>
- CopterFrames. (2015). *TwoDee EVO NG Brushless DSLR Gimbal Assembled*. Obtenido de CopterFrames: <http://copterframes.com/home/183-twodee-evo-ng-brushless-dslr-gimbal-assembled.html>
- Cubert. (s.f.). Software and Instalation Guide. *Cubert Utils Edel Weiss*. Albstr, Alemania.
- Cubert. (s.f.). Technical Specifications UHD 185 "Dragonfly". Cubert.
- Dingjie, W., Hanfeng, L., & Wu, J. (2016). *In-flight initial alignment for small UAV MEMS-based navigation via*. Obtenido de Aerospace Science and Technology: <http://dx.doi.org/10.1016/j.ast.2016.11.014>
- DJI. (Marzo de 2013). *ZENMUSE Z15*. Obtenido de dji: <http://www.dji.com/es/zenmuse-z15/spec>
- DJI. (27 de Marzo de 2013). Zenmuse Z15 User Manual V2.0. *[En Linea]*. [fecha de consulta: 7 setiembre - 15 octubre 2016], Disponible en <[https://cvp.com/pdf/z15\\_user\\_manual\\_v2.0\\_en.pdf](https://cvp.com/pdf/z15_user_manual_v2.0_en.pdf)>.
- DJI. (Octubre de 2015). *Ronin-M*. Obtenido de dji Store: [http://store.dji.com/es/product/ronin-m?site=brandsite&from=buy\\_now\\_bottom](http://store.dji.com/es/product/ronin-m?site=brandsite&from=buy_now_bottom)

- DJI. (Octubre de 2015). Ronin-M User Manual. *[En Línea]*. [fecha de consulta: 7 setiembre - 15 octubre 2016], Disponible en:  
<[http://dl.djicdn.com/downloads/ronin\\_m/en/Ronin\\_M\\_User%20Manual\\_v1.6\\_en.pdf](http://dl.djicdn.com/downloads/ronin_m/en/Ronin_M_User%20Manual_v1.6_en.pdf)>.
- DYS. (2015). *Brushless Gimbal Summer-Air*. Obtenido de dys:  
<http://www.dys.hk/ProductShow.asp?ID=162>
- DYS. (Julio de 2015). User Manual V1.0. *[En Línea]*. [fecha de consulta: 7 setiembre - 15 octubre 2016], Disponible en: <<http://www.dys.hk/Download/Gimbal/EN%20manua%20l%20for%20Summer-Air%2020150702.pdf>>.
- DYS. (Mayo de 2015). User Manual V1.1. *[En Línea]*. [fecha de consulta: 7 setiembre - 15 octubre 2016], Disponible en: <<http://www.dys.hk/Download/Gimbal/EN%20DYS-summer%20manual201505.pdf>>.
- Ebay. (2016). *Black CNC FPV Quadcopter BGC 2-Axis Brushless Gimbal&Controller f GoPro 3 Camera*. Obtenido de Ebay: <http://www.ebay.com/itm/Black-CNC-FPV-Quadcopter-BGC-2-Axis-Brushless-Gimbal-Controller-f-GoPro-3-Camera-/151670015226>
- Ebay. (2016). *DYS BGM4108-130T Hollow shaft Brushless Gimbal Motor stabilizer gyroscope Camera*. Obtenido de Ebay: <http://www.ebay.com/itm/DYS-BGM4108-130T-Hollow-shaft-Brushless-Gimbal-Motor-stabilizer-gyroscope-Camera-/331978838072?hash=item4d4b7b1038:g:zN8AAOSwEeFVS1Xj>
- fitPC. (2016). *Order fitlet*. Obtenido de fitPC Quality fanless computers: <http://www.fit-pc.com/web/purchasing/order-fitlet/>
- GimbalKing [GimbalKing]. (6 de Noviembre de 2015). *DYS SUMMER 3 EJES GIMBAL Guía Rapida*. *[Archivo de Video]*. Obtenido de Youtube: <https://www.youtube.com/watch?v=6sKOSXPIE5w>
- Gyroscopes.org. (2016). *The gyroscope gallery*. Obtenido de GYROSCOPES.ORG:  
<http://www.gyroscopes.org/gallery.asp>
- HD Videoshop [HDVideoshop Berlin]. (30 de Enero de 2016). *NEW CAME-TV ARGO - Features Overview - 3 axis gimbal for cameras up to 3Kg review*. *[Archivo de Video]*. Obtenido de Youtube:  
<https://www.youtube.com/watch?v=skoaprbjRWI>
- iDS. (2016). *uEye Manual 4.81*. *[En Línea]*. [fecha de consulta: 28 Setiembre - 18 Octubre 2016], Disponible en <<https://es.ids-imaging.com/manuals-ueye-software.html>>.
- iKancorp. (2016). *DS1 Beholder Gimbal for DSLRs*. Obtenido de iKan:  
<https://ikancorp.com/Downloads/DS1/DS1-OS.pdf>
- Ixon. (2016). *A-TEMPO Manual de Usuario*. *[Manual Digital]*. [fecha de consulta 14 Diciembre 2016 ].
- Jarrah, M., & Sahawneh, L. (2008). *Development and calibration of low cost MEMS IMV for V applications*. United States: International Symposium on Mechatronics and its Applications.
- Lee, H.-P., & Yoo, I.-E. (2007). *Robust Control Design for a Two-axis Gimbaled Stabilization System*. Daejeon, Korea: Agency for Defense Development.

- Liang, O. (2014). *Servo Camera Gimbal VS Brushless Gimbal*. Obtenido de <https://oscarliang.com/servo-brushless-camera-gimbal-fpv-quadcopter/>
- Lin, C., & Yang, S.-K. (2014). *Camera Gimbal Tracking from UAV Flight Control*. Tainan, Taiwan: Department of Aeronautics and Astronautics National Cheng Kung University.
- Magnussen, Ø., Hovland, G., & Ottestad, M. (2014). Multicopter UAV Design Optimization. *2014 IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA)*. Ancona, Italia.
- Malenovský, Z., Veness, T., & Wallace, L. (2014). HyperUAS-Imaging Spectroscopy from a Multirotor Unmanned Aircraft System. *Journal of Field Robotics*, 571-590.
- Marine, J. (11 de Julio de 2013). *DEFY G2 Gimbal Camera Stabilizer from Relentless Now Available for Pre-Order*. Obtenido de No Film School: <http://nofilmschool.com/2013/07/defy-g2-gimbal-camera-stabilizer-pre-order>
- Minigimbal. (2016). *Minigimbal*. Obtenido de MInigimbal: <http://minigimbal.com/>
- Mitchell, J. (2016). Flight Considerations And Hyperspectral Image Classification for Dryland Vegetation Management from a Fixed-wing UAS. *Environmental Management and Sustainable Development*, 41-57.
- Otlowski, D., & Wiener, K. (2006). *Critical Balancing of Gimballed Sensor Platforms*. Berlin, Alemania: Space Electronics LLC.
- Photon. (s.f.). *Our Technology | Hyperspectral Imaging*. Obtenido de Photon etc: <http://www.photonetc.com/hyperspectral-imaging>
- Priego Torres, B. (Julio de 2011). Técnicas de Inteligencia Computacional Aplicadas al Procesamiento de Señal en Imágenes Hiperespectrales. (*Tesis de Maestría*). Universidad de la Coruña, Coruña, España.
- ProKit. (2015). *Defy G2 Gimbal System-Used*. Obtenido de Prokit: <https://www.prokit.com/defy-g2-gimbal-system/>
- ProKit. (2015). *Defy G5 Gimbal System*. Obtenido de ProKit: <https://www.prokit.com/defy-g5-gimbal-system/>
- Scyllarus. (2016). *Hyperspectral Imaging*. Obtenido de SCYLLARUS Beyond Vision: <https://scyllarus.data61.csiro.au/about/hyperspectral-imaging/>
- Shahid, B., Shahzad, U., Bukhari, R., & Abasi, A. (2016). Autonomous Vehicle Using GPS and Magnetometer with HMI on LabView. *Asia-Pacific Conference on Intelligent Robot Systems* (págs. 163-167). Pakistan: The Islamia University of Bahawalpur.
- Sham, A. (22 de Marzo de 2015). *The Drone Flies*. Obtenido de 2-Axis vs 3-Axis: <http://www.thedronefiles.net/2015/03/22/2-axis-vs-3-axis-how-to-pick-the-right-camera-gimbal/>

- SHAPE [SHAPE]. (27 de Noviembre de 2015). ISEE+RIG. [*Archivo de Video*]. Obtenido de Youtube: <https://www.youtube.com/watch?v=6-LybzTM3DI>
- SHAPE. (2015). SHAPE Owner's Manual iSEE +. [*En Línea*]. [fecha de consulta: 7 setiembre - 15 octubre 2016], Disponible en: [http://www.shapewlb.com/axisdocument.aspx?id=920&langue=en&download=true&document=ISEE+\\_INSTRUCTION\\_MANUAL.pdf](http://www.shapewlb.com/axisdocument.aspx?id=920&langue=en&download=true&document=ISEE+_INSTRUCTION_MANUAL.pdf).
- Storm32BGC. (23 de Julio de 2016). *Getting Starte*. Obtenido de STorM32 BGC NT: [http://www.olliw.eu/storm32bgc-wiki/Getting\\_Started](http://www.olliw.eu/storm32bgc-wiki/Getting_Started)
- Suomalainen, J., Anders, N., Iqbal, S., Roerink, G., Franke, J., Wenting, P., . . . Kooistra, L. (2014). A Lightweight Hyperspectral Mapping System and Photogrammetric Processing Chain for Unmanned Aerial Vehicles. *remote sensing*, 11013-11030.
- Tarot. (2 de Abril de 2015). *Tarot 5D 3-axis brushless gimbal for X series TL5D001*. Obtenido de Tarot: [http://www.tarot-rc.com/index.php?main\\_page=product\\_info&cPath=65\\_96&products\\_id=1340&zenid=462d6b58c5567348f43b616d92082dba](http://www.tarot-rc.com/index.php?main_page=product_info&cPath=65_96&products_id=1340&zenid=462d6b58c5567348f43b616d92082dba)
- Tarot. (16 de Marzo de 2015). ZTX-5D User Manual. [*En línea*]. [fecha de consulta: 7 Setiembre-15 Octubre 2016 ], Disponible en: <http://site.hobby-wing.com/images/Manual/TL5D001-instruction-manual.pdf>.
- ThanksBuyer. (2015). *HIFLY 3-Axis Gimbal Brushless Handheld Gimbal Copter Stabilizer 5208-200T Newest Version*. Obtenido de ThanksBuyer Great Gadgets, Price & Serving: <http://www.thanksbuyer.com/hifly-3-axis-gimbal-brushless-handheld-gimbal-copter-stabilizer-5208-200t-newest-version-28327>
- ThanksBuyer. (s.f.). *Storm32 BGC 32Bit 3-Axis Brushless Gimbal Controller V1.31 DRV8313 Motor Driver*. Obtenido de ThanksBuyer Great Gadgets, Price & Serving: <http://www.thanksbuyer.com/storm32-bgc-32bit-3-axis-brushless-gimbal-controller-v1-31-drv8313-motor-driver-35504>
- The Photo Performance. (2015). *Pilotfly H1+ Gimbal Stabilizer*. Obtenido de The Photo Performance: <http://photoperformance.photos/gear/pilotfly-h1-gimbal-stabilizer/>
- VR Gimbal. (2014). *VR Gimbal controller*. Obtenido de Virtual Robotix: <https://vrgimbal.wordpress.com/>
- Wolfgang, D. (2016). *Open MP*. Obtenido de Admin New Day HPC: <http://www.admin-magazine.com/HPC/Articles/Programming-with-OpenMP>
- Yelangu. (6 de Enero de 2016). *Yelangu G2 3 Axis Gimbal*. Obtenido de Yelangu: [http://en.yelangu.com/prod\\_view.aspx?TypeId=50012&Id=135&FId=t3:50012:3](http://en.yelangu.com/prod_view.aspx?TypeId=50012&Id=135&FId=t3:50012:3)
- Zarco-Tejeda, P., González-Dugo, V., & Berni, J. (2012). Fluorescence, temperature and narrow-band indices acquired from a UAV platform for water stress detection using a micro-hyperspectral imager and a thermal camera. *Remote Sensing of Environment*, 322-337.

## 8. Anexos

### 8.1. -Fuentes Consultadas para Tabla de Comparación de Gimbals

- Algin. (18 de Abril de 2015). G3-5D Instruction Manual. *[En Línea]*. [fecha de consulta: 7 Setiembre - 15 Octubre 2016], Disponible en: <<http://www.align.com.tw/ManuaLib/Multi/Gimbal/RGG302.pdf>>.
- Align. (2015). *G3-5D Gimbal*. Obtenido de Align Conquer Your Heart: <http://www.align.com.tw/aerial-equip-en/multicopter/g35d/>
- Axis. (s.f.). *Brushless Gimbal + Joystick*. Obtenido de Axis: <http://axisgimbal.com/3-axis-gimbal/brushless-gimbal/>
- Axis. (s.f.). *Brushless Gimbal Startup Bundle*. Obtenido de Axis: <http://axisgimbal.com/3-axis-gimbal/dslr-3-axis-brushless-gimbal-startup-bundle/>
- B&H Photo-Video-ProAudio. (2016). *ikan DS1 Beholder Gimbal for Select DSLRs and Mirrorless*. Obtenido de B&H: [https://www.bhphotovideo.com/c/product/1200662-REG/ikan\\_ds1\\_beholder\\_gimbal\\_for.html](https://www.bhphotovideo.com/c/product/1200662-REG/ikan_ds1_beholder_gimbal_for.html)
- BaseCam Electronics. (2016). *BaseCam SimpleBGC 32-bit*. Obtenido de BaseCam Electronics: <http://www.basecamelectronics.com/simplebgc32bit/>
- CAME-Tv. (2016). *CAME-Argo*. Obtenido de CAME-TV: <http://www.came-tv.com/comeargo-3-axis-gimbal-camera-32bit-boards-with-encoders-p-778.html?zenid=r93le2ejf21gapn988smj7feh4>
- CAME-Tv. (2016). *CAME-Argo User Manual 1*. *[En Línea]*. [fecha de consulta: 7 setiembre - 15 octubre 2016], Disponible en: <<http://cametvblog.com/tag/user-manual/>>.
- CAME-Tv. (2016). *CAME-Argo User Manual 2*. *[En Línea]*. [fecha de consulta: 7 setiembre - 15 octubre 2016], Disponible en: <<http://cametvblog.com/tag/user-manual/>>.
- CAME-Tv. (2016). *CAME-Mini 3 User Manual V1.0*. *[En Línea]*. [fecha de consulta: 7 setiembre - 15 octubre 2016], Disponible en: <<http://cametvblog.com/tag/user-manual/>>.
- CAME-Tv. (2016). *CAME-Mini 3/2/1*. Obtenido de CAME-TV: <http://www.came-tv.com/camemini-3-3axis-gimbal-camera-32bit-boards-with-encoders-p-810.html>
- CAME-Tv. (2016). *CAME-Tv Mini 3 Air User Manual V1.0*. *[En Línea]*. [fecha de consulta: 7 setiembre - 15 octubre 2016], Disponible en: <<http://cametvblog.com/tag/user-manual/>>.
- CheesyCam [MrCheesyCam]. (28 de Febrero de 2016). *CAME-TV ARGO 32bit SBGC Gimbal Stabilizer with Encoder Hardware Product Overview*. *[Archivo de Video]*. Obtenido de Youtube: <https://www.youtube.com/watch?v=tGK3y7z1VhQ>

CheesyCam [MrCheesyCam]. (27 de Marzo de 2016). CAME-TV Mini 3 32 bit SBGC Gimbal with new Encoders Product Overview. *[Archivo de Video]*. Obtenido de YouTube: <https://www.youtube.com/watch?v=f6CtE8XH5u0>

CopterFrames. (2015). *TwoDee EVO NG 3x 5208 Gimbal Bruhsless Motor*. Obtenido de CopterFrames: <http://copterframes.com/motor-prop-esc/49-1.html>

CopterFrames. (2015). *TwoDee EVO NG Brushless DSLR Gimbal Assembled*. Obtenido de CopterFrames: <http://copterframes.com/home/183-twodee-evo-ng-brushless-dslr-gimbal-assembled.html>

DJI. (Marzo de 2013). *ZENMUSE Z15*. Obtenido de dji: <http://www.dji.com/es/zenmuse-z15/spec>

DJI. (27 de Marzo de 2013). Zenmuse Z15 User Manual V2.0. *[En Linea]*. [fecha de consulta: 7 setiembre - 15 octubre 2016], Disponible en <[https://cvp.com/pdf/z15\\_user\\_manual\\_v2.0\\_en.pdf](https://cvp.com/pdf/z15_user_manual_v2.0_en.pdf)>.

DJI. (Octubre de 2015). *Ronin-M*. Obtenido de dji Store: [http://store.dji.com/es/product/ronin-m?site=brandsite&from=buy\\_now\\_bottom](http://store.dji.com/es/product/ronin-m?site=brandsite&from=buy_now_bottom)

DJI. (Octubre de 2015). Ronin-M User Manual. *[En Linea]*. [fecha de consulta: 7 setiembre - 15 octubre 2016], Displonible en: <[http://dl.djicdn.com/downloads/ronin\\_m/en/Ronin\\_M\\_User%20Manual\\_v1.6\\_en.pdf](http://dl.djicdn.com/downloads/ronin_m/en/Ronin_M_User%20Manual_v1.6_en.pdf)>.

DYS. (2015). *Brushless Gimbal Summer-Air*. Obtenido de dys: <http://www.dys.hk/ProductShow.asp?ID=162>

DYS. (Julio de 2015). User Manual V1.0. *[En Linea]*. [fecha de consulta: 7 setiembre - 15 octubre 2016], Disponible en: <<http://www.dys.hk/Download/Gimbal/EN%20manua%20%20for%20Summer-Air%2020150702.pdf>>.

DYS. (Mayo de 2015). User Manual V1.1. *[En Linea]*. [fecha de consulta: 7 setiembre - 15 octubre 2016], Disponible en: <<http://www.dys.hk/Download/Gimbal/EN%20DYS-summer%20manual201505.pdf>>.

Ebay. (2016). *DYS BGM4108-130T Hollow shaft Brushless Gimbal Motor stabilizer gyroscope Camera*. Obtenido de Ebay: <http://www.ebay.com/itm/DYS-BGM4108-130T-Hollow-shaft-Brushless-Gimbal-Motor-stabilizer-gyroscope-Camera-/331978838072?hash=item4d4b7b1038:g:zN8AAOSwEeFVS1Xj>

GimbalKing [GimalKing]. (6 de Noviembre de 2015). *DYS SUMMER 3 EJES GIMBAL Guía Rapida*. *[Archivo de Video]*. Obtenido de Youtube: <https://www.youtube.com/watch?v=6sKOSXPIE5w>

HD Videoshop [HDVideoshop Berlin]. (30 de Enero de 2016). *NEW CAME-TV ARGO - Features Overview - 3 axis gimbal for cameras up to 3Kg review*. *[Archivo de Video]*. Obtenido de Youtube: <https://www.youtube.com/watch?v=skoaprbjRWI>

iKancorp. (2016). *DS1 Beholder Gimbal for DSLRs*. Obtenido de iKan: <https://ikancorp.com/Downloads/DS1/DS1-OS.pdf>

Marine, J. (11 de Julio de 2013). *DEFY G2 Gimbal Camera Stabilizer from Relentless Now Available for Pre-Order*. Obtenido de No Film School: <http://nofilmschool.com/2013/07/defy-g2-gimbal-camera-stabilizer-pre-order>

- Minigimbal. (2016). *Minigimbal*. Obtenido de MInigimbal: <http://minigimbal.com/>
- ProKit. (2015). *Defy G2 Gimbal System-Used*. Obtenido de Prokit: <https://www.prokit.com/defy-g2-gimbal-system/>
- ProKit. (2015). *Defy G5 Gimbal System*. Obtenido de ProKit: <https://www.prokit.com/defy-g5-gimbal-system/>
- SHAPE [SHAPE]. (27 de Noviembre de 2015). ISEE+RIG. [*Archivo de Video*]. Obtenido de Youtube: <https://www.youtube.com/watch?v=6-LybzTM3DI>
- SHAPE. (2015). SHAPE Owner's Manual iSEE +. [*En Línea*]. [fecha de consulta: 7 setiembre - 15 octubre 2016], Disponible en: <[http://www.shapewlb.com/axisdocument.aspx?id=920&langue=en&download=true&document=ISEE+\\_INSTRUCTION\\_MANUAL.pdf](http://www.shapewlb.com/axisdocument.aspx?id=920&langue=en&download=true&document=ISEE+_INSTRUCTION_MANUAL.pdf)>.
- Storm32BGC. (23 de Julio de 2016). *Getting Starte*. Obtenido de STorM32 BGC NT: [http://www.olliw.eu/storm32bgc-wiki/Getting\\_Started](http://www.olliw.eu/storm32bgc-wiki/Getting_Started)
- Tarot. (2 de Abril de 2015). *Tarot 5D 3-axis brushless gimbal for X series TL5D001*. Obtenido de Tarot: [http://www.tarot-rc.com/index.php?main\\_page=product\\_info&cPath=65\\_96&products\\_id=1340&zenid=462d6b58c5567348f43b616d92082dba](http://www.tarot-rc.com/index.php?main_page=product_info&cPath=65_96&products_id=1340&zenid=462d6b58c5567348f43b616d92082dba)
- Tarot. (16 de Marzo de 2015). ZTX-5D User Manual. [*En línea*]. [fecha de consulta: 7 Setiembre-15 Octubre 2016 ], Disponble en: <<http://site.hobby-wing.com/images/Manual/TL5D001-instruction-manual.pdf>>.
- ThanksBuyer. (2015). *HIFLY 3-Axis Gimbal Brushless Handheld Gimbal Copter Stabilizer 5208-200T Newest Version*. Obtenido de ThanksBuyer Great Gadgets, Price & Serving: <http://www.thanksbuyer.com/hifly-3-axis-gimbal-brushless-handheld-gimbal-copter-stabilizer-5208-200t-newest-version-28327>
- ThanksBuyer. (s.f.). *Storm32 BGC 32Bit 3-Axis Brushless Gimbal Controller V1.31 DRV8313 Motor Driver*. Obtenido de ThanksBuyer Great Gadgets, Price & Serving: <http://www.thanksbuyer.com/storm32-bgc-32bit-3-axis-brushless-gimbal-controller-v1-31-drv8313-motor-driver-35504>
- VR Gimbal. (2014). *VR Gimbal controller*. Obtenido de Virtual Robotix: <https://vrgimbal.wordpress.com/>
- Yelangu. (6 de Enero de 2016). *Yelangu G2 3 Axis Gimbal*. Obtenido de Yelangu: [http://en.yelangu.com/prod\\_view.aspx?TypeId=50012&Id=135&FId=t3:50012:3](http://en.yelangu.com/prod_view.aspx?TypeId=50012&Id=135&FId=t3:50012:3)

## 9. Apéndices

### 9.1. -Preguntas Sobre Requisitos de Gimbals

Se presenta en detalle las distintas preguntas realizadas a los encargados del GII para poder perfilar adecuadamente la selección del gimbal, se presenta una razón del porque la pregunta es importante, así como la respuesta dada.

#### 1. Que prioridad u orden reciben los criterios de elección

- **Razón:** Necesario para poder decidir un perfil final y darlo como propuesta
- **Respuesta:** El orden de prioridad no se puede definir a priori, pero el proceso de selección final no se debe realizar de forma individual sino por una discusión con los distintos encargados donde se perfila dicha prioridad. De momento se debe procurar obtener varios modelos viables y la prioridad de criterios se genera en mano del proceso de elección.

#### 2. Debe ser estrictamente un producto comercial cerrado o se puede contemplar la opción de hacer una unión de elementos separados

- **Razón:** La opción de obtener la estructura, motores, controlador y sensor por aparte para luego unirlos es una opción que no se descarta a priori y se debe confirmar si es válida.
- **Respuesta:** Es una opción válida y se debe contemplar como opción aún en presencia de la falta de especificaciones que presentan este tipo de modelos.

#### 3. Que orientación debe llevar la cámara en todo momento

- **Razón:** En caso de que la orientación siempre sea ver hacia abajo se puede contemplar más hacer uso de gimbals de dos ejes, dejando de lado el eje yaw ya que solo afecta en la orientación del suelo en la imagen y no altera el ángulo entre la LOS de la cámara y la superficie del suelo.
- **Respuesta:** En principio se plantea que la cámara apunte hacia abajo con la LOS perpendicular al plano de superficie del suelo.

#### 4. Que tan necesario es el control del eje yaw y la orientación del suelo en la imagen

- **Razón:** Cuando se habla de la orientación del suelo se hace referencia a la rotación sobre el eje yaw perpendicular al suelo, visto desde el cielo en un plano paralelo al del suelo el eje yaw controla la rotación del suelo respecto a la cámara. Si la orientación del suelo no fuese un factor importante entonces se puede obviar el control en el eje yaw lo que abre las puertas a sistemas de 2 ejes.
- **Respuesta:** Por temas de facilitar el preprocesado de imágenes o inspección rutinaria de una misma sección de suelo si es importante controlar la orientación completa de la cámara. Pero por temas de unión de imágenes en una única imagen panorámica, protoconstrucción panorámica, no afecta; esto en el caso de una snapshot. Para una cámara pushbroom si conviene asegurar totalmente la orientación de la cámara.

#### 5. Se busca tomar fotos o videos

- **Razón:** Las exigencias en fotos o videos son distintas donde los sistemas más precisos se utilizan para video.

- **Respuesta:** En principio se planea tomar fotos pero no por esto se puede reducir la exigencia en la estabilización.
- 6. Peso que puede soportar el UAV**
- **Razón:** El peso máximo es un criterio arbitrario de descarte inicial. Permite desechar modelos de gimbal por un peso excesivo que no puede elevarse en vuelo.
  - **Respuesta:** El proveedor del UAV no ha dado un documento actualizado con especificaciones, pero el documento que presenta el modelo inicial hace mención que la carga útil es de 5 kg. Este valor contempla algunos elementos propios del UAV que luego habrá que quitar o seleccionar, pero en principio se puede trabajar con el valor de 5 kg como referencia máxima. Se aclaró que reducir el peso es un objetivo que se debe mantener a lo largo de la búsqueda.
- 7. Dimensiones físicas de la zona donde se localiza el dron**
- **Razón:** Permiten generar un criterio sobre el tamaño máximo que puede tener el gimbal
  - **Respuesta:** El proveedor no ha dado documento sobre las dimensiones exactas, pero si se sabe que la única que importa es la altura ya que un gimbal que sea más grandes que el soporte de aterrizaje del UAV dificultaría el proceso de aterrizaje e implicaría modificaciones posteriores al UAV.
- 8. La batería del UAV alimenta el gimbal o debe contemplarse esto por aparte**
- **Razón:** La batería aporta en peso y conexiones.
  - **Respuesta:** En la búsqueda si importaría tomar el dato de la batería a utilizar pero la alimentación de energía cae bajo la responsabilidad de otro miembro del proyecto.
- 9. Información sobre las posibles cámaras a utilizar**
- **Razón:** El peso de carga es un parámetro que caracteriza a los gimbal comerciales y las dimensiones de la cámara permiten observar si hay espacio en el gimbal para poder realizar la operación de balanceo.
  - **Respuesta:** Se obtuvo un documento con varias características de una de las posibles cámaras hiperespectrales a utilizar, en esta se detalla que el peso de la cámara es de 470g y sus dimensiones son 195 x 67 x 60 mm. Esto en conjunto con la información dada en el breve análisis de las cámaras hiperespectrales visto en la sección “Comprensión de Cámara Hiperespectral”.

## 9.2. Tabla Completa de Valores de Proporción de Imagen y Valor F para Corrección de Tamaño de Píxel

Tabla 9.2-1 Medición de Proporciones de Imagen y Valor F para Todas las Pruebas

Nombre de Captura	Datos de Imágenes de Barrido e Imagen Real						
	rPB_1	rPB_2	rR_1	rR_2	r_Prom	rR_Prom_Inv	F
GOI_1.1	24.2836	23.78	2.5484	2.5106	24.0318	0.395335046	9.500613
GOI_1.2	27.85	25.35	2.3673	2.5761	26.6	0.404579844	10.76182

GOI_1.3	22.9517	24.1533	2.5824	2.5106	23.5525	0.392695857	9.248969
GOI_1.4	22.85	24.016	2.4286	2.43	23.433	0.411641214	9.645989
GOI_2.1	34.69	33.969	2.5053	2.55	34.3295	0.395624394	13.58159
GOI_2.2	35.4	32.97	2.41671	2.54	34.185	0.403493446	13.79342
Goi_2.3	34.35	35.04	2.604	2.6264	34.695	0.382379933	13.26667
GOI_3.1	41.19	43.0786	2.5	2.5326	42.1343	0.397408894	16.74455
GOI_3.2	42.06	39.66	2.597	2.5326	40.86	0.389893949	15.93107
GOI_3.3	6.1512	6.0005	0.377	0.3527	6.07585	2.740852405	16.65301
GOI_4.1	17.3854	17.77	2.4433	2.5543	17.5777	0.400192092	7.034457
GOI_4.2	16.8567	17.4041	2.5161	2.5106	17.1304	0.397875346	6.815764
GOI_4.3	16.1891	15.9381	2.3469	2.3776	16.0636	0.42332522	6.800127
GOI_4.4	14.339	16.115	2.366	2.4082	15.227	0.418918353	6.37887
	<b>rPB_1</b>	<b>rPB_2</b>	<b>rR_1</b>	<b>rR_2</b>	<b>r_Prom</b>	<b>rR_Prom_Inv</b>	<b>F</b>
GOI_5.1	24.15	24.6431	2.3878	2.5543	24.39655	0.404686267	9.872949
GOI_5.2	23.53	23.61	2.4947	2.3636	23.57	0.411666632	9.702983
GOI_5.3	22.6237	22.4619	2.433	2.3762	22.5428	0.415869583	9.374865
GOI_6.1	10.9006	10.5103	0.93	0.9368	10.70545	1.071352046	11.46931
GOI_6.2	28.54	29.82	2.406	2.5604	29.18	0.402706186	11.75097
GOI_6.3	3.6866	3.9272	0.3672	0.3828	3.8069	2.666666667	10.15173
GOI_7.1	12.6527	12.46	2.4184	2.4639	12.55635	0.409642996	5.143621
GOI_7.2	13.2328	12.88	2.6154	2.3673	13.0564	0.401388805	5.240693
GOI_7.3	12.2278	12.668	2.3636	2.4021	12.4479	0.419665527	5.223955
GOI_7.4	12.301	11.4538	2.3776	2.3737	11.8774	0.420937428	4.999642
GOI_8.1	17.9209	17.0168	2.4526	2.2095	17.46885	0.428991227	7.493983
GOI_8.2	16.2987	16.048	2.3267	2.2941	16.17335	0.432825485	7.000238
GOI_8.3	16.8081	16.4651	2.5208	2.4737	16.6366	0.400440485	6.661968
GOI_9.1	8.8149	8.6523	0.92	0.901	8.7336	1.098297639	9.592092
GOI_9.2	21.1142	21.3958	2.39801	2.4479	21.255	0.41271918	8.772346
GOI_9.3	3.1668	3.107	0.377	0.3546	3.1369	2.733734281	8.575451

### 9.3. Biblioteca “Control\_Pushbroom\_UI3360\_CP\_GII.h”

#### Descripción

La biblioteca “Control\_pushbroom\_UI3360\_CP\_GII.h” permite operar la cámara UI3360-CP conectada al hiperspectrómetro del GII. La biblioteca utiliza y combina múltiples de las funciones de la biblioteca uEye.h oficial del proveedor. Entre sus principales ventajas se encuentra la reducción del ciclo de encendido a una única función así como funciones para la captura, almacenado y manejo de múltiples espacio de memoria. La biblioteca también cuenta con varias funciones de uso general que fueron diseñadas para cumplir con las necesidades del proyecto SIMMAP pero que no recaen en el control propio de la cámara sino solo en el manejo de su información o similares.

#### Requisitos y Consideraciones

Para poder utilizar la biblioteca es necesario realizar las siguientes labores.

1. Instalar el driver “uEye-Full-#-#-WHQL.zip”, donde el primer # se refiere a la versión del driver y el segundo se refiere a si el sistema es de 32 o 64 bits. La biblioteca se diseñó específicamente con la versión 4802-64. Modificaciones a la API en versiones futuras son posibles por lo que queda a cargo del usuario cuidar el detalle. Se adjunta con el manual una copia del driver 4802-64. Los drivers se pueden encontrar en el link: <https://es.ids-imaging.com/download-ueye-win32.html>.
2. El proyecto que haga uso de la biblioteca debe también adjuntar los .h llamados: “uEye.h”. Que es la biblioteca oficial de la empresa IDS, el manual se puede observar en el link <https://es.ids-imaging.com/manuals-ueye-software.html>, se adjunta con el manual una copia de la biblioteca, no obstante al instalar drivers se puede encontrar en la dirección: C:\Program Files\IDS\uEye\Develop\include
3. El proyecto que haga uso de la biblioteca debe también adjuntar el .lib “uEye\_api.lib”. Se adjunta con el manual una copia del archivo .lib, no obstante al instalar drivers se puede encontrar en a dirección: C:\Program Files\IDS\uEye\Develop\lib
4. La biblioteca fue diseñada en Microsoft visual Studio Express 2012 para C++. Para sistemas operativos Linux se debería trabajar con la API específica de Linux, en este caso no se asegura que la biblioteca pueda servir.
5. **CRT\_SECURE\_NO\_WARNINGS:** Dependiendo del ordenador en el que se programe puede que sea o no sea necesario deshabilitar esta opción del compilador de Microsoft Visual Studio. Para mayor instrucciones seguir las instrucciones del link: <http://stackoverflow.com/questions/16883037/remove-secure-warnings-crt-secure-no-warnings-from-projects-by-default-in-vis> en forma reducida el proceso se basa en:
  - a. Click derecho sobre el proyecto en el que se trabaja y seleccionar “propiedades”
  - b. En la pantalla emergente asegurar que “Configuraciones = Todas las configuraciones”
  - c. En las pestaña Configuration Properties -> C/C++ -> Preprocessor hacer click en “preprocessor definitions”
  - d. Seleccionar editar y **agregar** la línea `_CRT_SECURE_NO_WARNINGS` en la parte superior del espacio para escribir. No se debe reemplazar ningún valor ya existente.

6. **Lógica por Hilos:** En caso de aplicar lógica de hilos a la biblioteca se debe tomar en cuenta que las funciones originales de la biblioteca uEye.h no permiten que dos hilos llamen a una misma función. Por esta razón no es posible utilizar dos funciones “save” o “destrucmem” al mismo tiempo en dos hilos distintos pero si es posible utilizar una función “capt” junto con una función “destrucmem” (asumiendo que no actúan sobre la misma lista de memorias). Además en caso de paralelizar las funciones “save” y “capt” es necesario saber que el acceso a memoria RAM se maneja a través de un único recurso por lo que no se obtendrá un paralelismo genuino.

### Explicación de Estructuras

Se procede a explicar las estructuras creadas para la biblioteca. Se presenta el nombre de la estructura, su utilidad, el código de declaración y la descripción de cada una de sus partes.

#### *Cmemoria*

Estructura que almacena toda la información necesaria de un “espacio de memoria” de trabajo de la cámara UI3360-CP. La API de la cámara UI3360-CP permite generar espacios en memoria RAM para almacenar las imágenes que captura, la estructura cmemoria contiene las propiedades que identifican dicho espacio de memoria así como otros para controlar el estado de dicha memoria. Las propiedades existen y tienen sentido en visión de las funciones de control de la cámara.

```
typedef struct cmemoria{
    INT ID;
    char* CImgMem;
    int capturaOK;
    int guardarOK;
    int numeroimg;
    cmemoria *cmemoriasiguiente;
}*ptrcmemoria;
```

- **Cmemoria:** El nombre dado al tipo de variable, toda variable que siga esta estructura deberá llamarse por este nombre, todo en minúsculas.
- **Ptrcmemoria:** Formato específico de los punteros a variables cmemoria, equivale a la sintaxis “cmemoria\*”
- **ID:** Primer valor que identifica el espacio de memoria, se le es asignado automáticamente a la estructura cuando la misma pasa a través del método “crearcmemoria”
- **CImgMem:** Segundo valor que identifica el espacio de memoria, se le es asignado automáticamente a la estructura cuando la misma pasa a través del método “crearcmemoria”
- **CapturaOK:** Valor que almacena si el espacio de memoria logró obtener una imagen de la cámara en el método “capt”. Se inicializa automáticamente en el método “crearmemoria” y se utiliza en las funciones “capt” y “Check” y “Datos\_Save”.
- **GuardarOK:** Valor que almacena si el espacio de memoria logró pasar su información a disco en el método “save”(es decir si se guardó o no efectivamente la imagen). Se inicializa automáticamente en el método “crearmemoria” y se utiliza en las funciones “save” y “Check” y “Datos\_Save”.

- **Numeroimg:** Variable que identifica mediante un número al espacio de memoria, útil para identificar un espacio de memoria de otro en las listas. Se inicializa en el método “seccionlistacmemoria”
- **Cmemoriasiguiente:** Puntero a una variable cmemoria, permite concatenar variables del tipo cmemoria en una lista dinámica.
- **NOTAS:** Para entender más y ver un ejemplo de la estructura al usarse ver el método “creacmemoria” y “seccionlistacmemoria”

### Tosave

Estructura que almacena toda la información necesaria para utilizar el método “save”, es decir todos las propiedades que dicha función recibe de entrada. Para comprenderla adecuadamente es necesario ver las funciones “grandinico” y “save”

```
typedef struct tosave{
    HIDS Cam; //Identificador de cámara
    ptrcmemoria memorias; //Inicio de lista de memorias
    int cantidad; //Cantidad de memorias
    std::string carpeta; //Nombre de carpeta en la cual guardar la información
    UINT formato; //Formato para guardar
};
```

- **Cam:** Handle de cámara de la cual guardar la información
- **Memorias:** Puntero a estructura cmemoria sobre el cual se inicia el guardado
- **Cantidad:** Cantidad de espacios de memoria que guardar, este valor debe ser menor o igual a la cantidad de memorias concatenadas al puntero de la variable “memorias”. (las variables cmemoria se concatenan a través de su propiedad cmemoriasiguiente).
- **Carpeta:** Variable que almacena el nombre de la carpeta sobre la cual se quiere guardar las imágenes, si se inserta “” como la propiedad de entrada se usara la dirección por defecto (la carpeta donde se encuentre la aplicación que ejecuta el método).
- **Formato:** Formato a guardar, más información en el método “save”.

### Explicación de Funciones

Se procede a explicar las funciones, se enuncia el nombre del método, sus propiedades de entrada, sus salidas, las funciones de la biblioteca oficial uEye que se utilizan (en caso de que se desee conocer a mayor profundidad las funciones o para modificaciones futuras), cualquier detalle de cuidado adicional y un ejemplo de su uso.

NOTA: Los tipos de variable HIDS, INT, UINT se definen en la biblioteca uEye.h

*grandinico*(HIDS Cam, INT triggermode, INT mododisplay, INT colormode, double fps, double expo)

- **Labor:** Se encarga de inicializar la cámara UI3360-CP con ciertas propiedades dados por el usuario. Si bien el método permite mucha versatilidad sobre como inicializar la cámara las demás

funciones tienden a asumir que la cámara se inició de una cierta forma, en la sección de notas se explican las propiedades exactas de trabajo de la biblioteca.

- **Entradas**

- Cam: Identificador HIDS, definido en biblioteca uEye, con el cual se identifica la cámara. El valor específico puede variar de cámara a cámara y es configurable por otras funciones de la API (no implementadas). Por defecto las cámaras poseen un identificador de "1".
- Triggermode: Variable que define la manera de operar la captura de la cámara, específicamente puede adoptar cualquiera de los valores de la figura 9.3-1. Para controlar la captura con funciones de la biblioteca se debe seleccionar el modo IS\_SET\_TRIGGER\_SOFTWARE

**Input parameters**

hCam	Camera handle	
nTriggerMode		
	Trigger mode	Trigger event
IS_SET_TRIGGER_OFF	Off	-
IS_SET_TRIGGER_HI_LO	Hardware trigger Enables in the <a href="#">memory mode</a> the post-trigger	Falling signal edge
IS_SET_TRIGGER_LO_HI	Hardware trigger Enables in the memory mode the post-trigger	Rising signal edge
IS_SET_TRIGGER_PRE_HI_LO	Enables the pre-trigger in the memory mode	Falling signal edge
IS_SET_TRIGGER_PRE_LO_HI	Enables the pre-trigger in the memory mode	Rising signal edge
IS_SET_TRIGGER_HI_LO_SYNC	Freerun sync./hardware trigger <sup>*1</sup>	Falling signal edge
IS_SET_TRIGGER_LO_HI_SYNC	Freerun sync./hardware trigger <sup>*1</sup>	Rising signal edge
IS_SET_TRIGGER_SOFTWARE	Software trigger	Call of <a href="#">is_FreezeVideo()</a> (single frame mode) Call of <a href="#">is_CaptureVideo()</a> (continuous mode)
IS_GET_EXTERNALTRIGGER	Returns the trigger mode setting	
IS_GET_TRIGGER_STATUS	Returns the current signal level at the trigger input	
IS_GET_SUPPORTED_TRIGGER_MODE	Returns the supported trigger modes	

\*1 The [freerun synchronization](#) mode is currently only supported by the sensors of the [UI-146x/UI-546x](#) series.

Figura 9.3-1 Posibles Valores de la propiedad triggermode

- Mododisplay: Variable que define el modo de presentar la imagen, puede optar por los valores de la figura 9.3-2; no obstante el valor por defecto es IS\_SET\_DM\_DIB

**Input parameters**

hCam	Camera handle
Mode	
IS_SET_DM_DIB	Captures an image in system memory (RAM). Using <a href="#">is_RenderBitmap()</a> , you can define the image display (default).
IS_SET_DM_DIRECT3D	Image display in Direct3D mode
IS_SET_DM_DIRECT3D   IS_SET_DM_MONO	Monochrome image display in Direct3D mode
IS_SET_DM_DIRECT3D   IS_SET_DM_BAYER	Raw Bayer format image display in Direct3D mode
IS_SET_DM_OPENGL	Image display in OpenGL mode
IS_SET_DM_OPENGL   IS_SET_DM_MONO	Monochrome image display in OpenGL mode
IS_SET_DM_OPENGL   IS_SET_DM_BAYER	Raw Bayer format image display in OpenGL mode
IS_GET_DISPLAY_MODE	Returns the current setting.

Figura 9.3-2 Posibles valores de la propiedad mododisplay

- colormode: Define el formato de captura de imágenes, es decir entre una captura monocromática u otra a escala RGB, a su vez define la cantidad de bits de profundidad por píxel en la imagen. EJ: IS\_CM\_MONO16 es un formato de captura monocromático con 16 bits de profundidad por píxel. La figura 9.3-3 presenta algunos más no todos los valores posibles de captura. **NOTA: Ver en el método creacmemoria el cuidado especial que debe existir entre el formato de captura definido por colormode y el proceso de asignar memoria RAM**

Input parameters

hCam	Camera handle
<input type="checkbox"/> <b>Mode:</b> Color mode to be set. For a list of all available color formats and the associated input parameters, see the <a href="#">Appendix: Color and memory formats</a> section.	
IS_CM_MONO16	Grayscale (16), for monochrome and color cameras, LUT/gamma active
IS_CM_MONO12	Grayscale (12), for monochrome and color cameras, LUT/gamma active
IS_CM_MONO10	Grayscale (10), for monochrome and color cameras, LUT/gamma active
IS_CM_MONO8	Grayscale (8), for monochrome and color cameras, LUT/gamma active
IS_CM_SENSOR_RAW16	Raw sensor data (16), for monochrome and color cameras, LUT/gamma not active
IS_CM_SENSOR_RAW12	Raw sensor data (12), for monochrome and color cameras, LUT/gamma not active
IS_CM_SENSOR_RAW10	Raw sensor data (10), for monochrome and color cameras, LUT/gamma not active
IS_CM_SENSOR_RAW8	Raw sensor data (8), for monochrome and color cameras, LUT/gamma not active

Figura 9.3-3 Algunos valores posibles para la propiedad colormode

- fps: Valor de framerate o frames por segundo que se desea asignar a la cámara. **NOTA:** Los posibles valores de tiempo de exposición de la cámara se ven afectados por el framerate, a menor fps mayor tiempo de exposición se puede asignar.
- Expo: Tiempo de exposición (en ms) con el cual se capturarán imágenes.
- **Retorna:** Principalmente el método logra inicializar la cámara, de ese punto en adelante todas las demás funciones requieren reutilizar la propiedad Cam para poder identificar a la cámara. Al ser una función del tipo **int** se retorna un número, el número implica un código de error en caso de que el método falle y permite identificar en que parte del proceso de inicialización se dio el fallo. Los posibles valores de retorno son:
  - 0: No hubo error, el método termino con éxito
  - 60: Error al asignar el tiempo de exposición
  - 50: Error al asignar el framerate
  - 40: Error al asignar el formato de captura (variable colormode)
  - 30: Error al asignar el modo de display
  - 20: Error al asignar el tipo de trigger (variable triggermode)
  - 10: Error con el método de inicialización de la cámara

Además se imprime en pantalla el código de error según la lista previa y un código propio de la biblioteca uEye que explica con más detalle el error en cuestión. Para poder entender el código de uEye ver el manual de Ids, sección C "Programming" en el apartado "Error Codes".

El código del método identifica con el comentario //\$\$\$\$\$\$\$\$\$\$\$\$ la sección en que se imprime esta información en caso de que se desee modificar la biblioteca para tomar otras acciones. Se presenta en la figura 9.3-4 un ejemplo del mensaje que se vería en pantalla si se da un fallo.

```
Error de código: 10
Error según librería uEye.h: 3
```

Figura 9.3-4 Ejemplo de Error en Función grandinicio

En este caso el código es 10 lo que implica que es un error con el método de inicialización y según el manual de uEye el error 3 implica "IS\_CANT\_OPEN\_DEVICE" que implica que la cámara no estaba conectada

- **Fuciones uEye:** Se utilizan las siguientes funciones de la biblioteca uEye.h
  - Is\_InitCamera
  - Is\_SetExternalTrigger
  - Is\_SetDisplayMode
  - Is\_SetColorMode
  - Is\_SetFrameRate
  - Is\_Exposure
- **Notas:** Si bien el método da mucha variedad en como inicializar la cámara, para poder utilizar la biblioteca se necesita que la propiedad triggermode sea IS\_SET\_TRIGGER\_SOFTWARE y la propiedad mododisplay sea IS\_SET\_DM\_DIB. La ganancia no es un parámetro que se controle pero por defecto es de x1. Por otro lado si el método no responde por error de la propiedad Cam y se desconoce si el mismo se ha cambiado se recomienda referirse al manual de uso en la sección, donde se habla de cómo modificar o conocer el valor identificador de la cámara. **IMPORTANTE: El valor de Cam para acceder a la cámara del hiperspectrómetro en el GII es de "1".**
- **Ejemplo:** En el ejemplo se puede observar como la variable Cam de entrada debe tener el valor específico de la cámara a conectar, además se espera que la cámara se inicialice correctamente con un framerate de 10 fps, un tiempo de exposición de 83 ms en los modos de trigger de software y DIB de display. Además se imprimirá en pantalla un valor de 0 en caso de que no se hayan dado errores, en caso de errores aparecerá un mensaje similar al de la figura 4 seguido de la impresión en pantalla del código de error.

```
HIDS hCam = 1; //Identificador de cámara
int resultado =
grandinicio(hCam, IS_SET_TRIGGER_SOFTWARE, IS_SET_DM_DIB, IS_CM_MONO16, 10, 83);
//Inicialización de cámara
std::cout<<"\nEstado de inicialización: "<<resultado;
```

*is\_ExitCamera(HIDS hCam)*

- **Información y utilidad:** No es propiamente una función de la biblioteca "Control\_Pushbroom\_UI360\_CP\_GII.h" sino de la biblioteca "uEye.h", se encarga de desconectar la cámara. Dado que su operación es directa y simple se consideró innecesario realizar una función propia. La única propiedad que recibe es el identificador de cámara que se desea desconectar.

*crearcmemoria(HIDS Cam, int bpp)*

- **Labor:** Función encargada de crear e inicializar una estructura del tipo cmemoria, efectivamente crea un espacio en memoria RAM en el cual se puede almacenar una imagen capturada, la estructura cmemoria contiene la información de dicho espacio de memoria. Formalmente inicializa los parámetros CImgMem, ID, capturaOK y guardarOK.
- **Entradas**
  - Cam: El identificador de la cámara en uso, debe ser un identificador de una cámara inicializada por el método grandinicio.
  - Bpp: Valor de bits de profundidad por píxel necesario para asignar la cantidad de memoria específica. El valor a utilizar depende directamente del modo de captura seleccionado con la propiedad “colormode” del método grandinicio. La figura 9.3-5 presenta parte de la tabla de la biblioteca oficial donde se relacionan valores de Bpp con su colormode respectivo.

Name (API)	Description	Bits/ channel	Bits/ Pixel	Total bits	Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
IS_CM_SENSOR_RAW16	<b>RAW</b>											
	Color sensor (1st row)	16	16	16	16*		16*					
	Color sensor (2nd row)	16	16	16	16*			16*				
	Monochrome sensor	16	16	16	16*							
IS_CM_SENSOR_RAW12	<b>RAW</b>											
	Color sensor (1st row)	12	16	16	12		12					
	Color sensor (2nd row)	12	16	16	12			12				
	Monochrome sensor	12	16	16	12							
IS_CM_SENSOR_RAW10	<b>RAW</b>											
	Color sensor (1st row)	10	16	16	10		10					
	Color sensor (2nd row)	10	16	16	10			10				
	Monochrome sensor	10	16	16	10							
IS_CM_SENSOR_RAW8	<b>RAW</b>											
	Color sensor (1st row)	8	8	8	8	8	8					
	Color sensor (2nd row)	8	8	8	8	8	8					
	Monochrome sensor	8	8	8	8							
IS_CM_MONO16	<b>Grey</b>	16	16	16	16*							
IS_CM_MONO12	<b>Grey</b>	12	16	16	12							
IS_CM_MONO10	<b>Grey</b>	10	16	16	10							
IS_CM_MONO8	<b>Grey</b>	8	8	8	8							

Figura 9.3-5 Tabla de Correspondencia entre Formato de Captura y Bits de Profundidad por Píxel

EJ: Si se selecciona IS\_CM\_MONO16 en el colormode del método grandinicio entonces todos los espacios de memoria se deben crear con bpp=16. Cabe destacar que el número de bpp es el de “Total bits” de la figura 9.3-5 y no el de Bits/cannel o Bits/píxel

La tabla entera se puede observar en el manual del proveedor IDS, en la sección de “Appendix”-“Color and memory formats”

- **Retorna:** El método retorna un puntero ptrcmemoria con dos posibles resultados.
  - Si el método se ejecuta correctamente: Se retorna un puntero hacia la estructura cmemoria creada, mediante este puntero se puede acceder a la información.
  - Si el método falla: Se retorna un puntero vacío hacia NULL.
- **Funciones uEye:** Se utiliza el método is\_AlloclmageMem
- **Notas:** Dada la forma en que se diseñó la biblioteca Control\_Pushbroom\_UI3360\_CP\_GII no es necesario en ningún momento alterar los valores CImgMem e ID de una estructura cmemoria, se recomienda nunca alterar estos valores. El método intenta asignar la memoria múltiples veces en

caso de fallo, para así no fallar por errores de turno, cuando se retorna un puntero a NULL es porque se ha superado 200 intentos de asignar la memoria. Finalmente los valores de caputraOK y guardarOK se inicializan con valor 0, la propiedad numeroimg no se inicializa.

- **Ejemplo:** En el ejemplo se observa como una nueva variable tipo ptrcmemoria recibe el resultado del método crearcmemoria y en caso de que la misma no sea NULL imprime en pantalla los valores de ID y CImgMem. Es importante destacar que como el método grandinicio se utilizó con un colormode IS\_CM\_MONO8 el valor de bpp en el método crearcmemoria fue de 8, todo de acuerdo a la información en la figura 5.

```
int resultado =
grandinicio(hCam,IS_SET_TRIGGER_SOFTWARE,IS_SET_DM_DIB,IS_CM_MONO8,10,83);
//Inicialización de cámara
ptrcmemoria ej = crearcmemoria(hCam,8);
if(ej != NULL){
    std::cout<<ej->CImgMem<<"\n";
    std::cout<<ej->ID<<"\n";
}
```

*Seccionlistacmemoria*(HIDS Cam,ptrcmemoria &iniciolista,int cantidad,int inicio, int bpp)

- **Labor:** El método crea una lista dinámica de estructuras tipo cmemoria, cada una inicializada y creada mediante el método cmemoria. Se vale de la propiedad “cmemoriasiguientes” de cada estructura cmemoria para concatenar una con otra. En su operación asigna la propiedad “numeroimg” correspondiente a cada estructura de la lista para que sirva así de identificador.
- **Entradas**
  - Cam: El identificador de la cámara en uso, debe ser un identificador de una cámara inicializada por el método grandinicio.
  - iniciolista: Variable del tipo ptrcmemoria en la cual se entrega el puntero a la primera estructura cmemoria de la lista dinámica. No requiere ningún valor inicial pero si se desea se puede introducir una variable inicializada en NULL.
  - Cantidad: La cantidad de memorias a conformar la lista dinámica.
  - Inicio: Número que permite a el método enumerar las estructuras desde un valor variable. **El valor de la propiedad “numeroimg” de la primera estructura** de la lista dinámica siempre responde a la ecuación  $numeroimg = inicio + 1$ . El uso principal de la propiedad es para mantener un orden entre múltiples listas dinámicas. VER EJEMPLO PARA MAS DETALLE
  - Bpp: Bits de profundidad por píxel, aplican las mismas reglas que en el caso del método crearcmemoria
- **Retorna:** Inicialmente la variable de entrada iniciolista se ve modificada y contiene la dirección a la primera estructura cmemoria de la lista dinámica. Propiamente el método puede retornar dos valores distintos.
  - 1: Cuando no se logra generar la lista con toda la cantidad de memorias deseadas, además de esto se imprime en pantalla un mensaje que enseña el último espacio de memoria que se pudo crear. Se marca con el comentario //\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$ el sector del código que presenta el mensaje de error en caso de que se desee alterar este proceso.

- 0: Cuando el método se ejecuta sin error alguno.
- **Funciones uEye:** De forma indirecta se utiliza el método `is_AllocImageMem`
- **Notas:** N/A
- **Ejemplo:** En el ejemplo se observa cómo se inicializan dos cadenas dinámicas de memoria, ambas a 16 bpp dada el método de `grandinico` previa. La primera cadena contiene 100 elementos y dado que su valor de inicio es 0 la primera estructura `cmemoria` contiene el parámetro `numeroimg=1`, el último elemento de la primera cadena contiene el `numeroimg=100`, la segunda cadena es de 100 elementos y dado que su valor de inicio es de 100 el primer elemento de la cadena contiene el `numeroimg=101`. Efectivamente las dos listas forman una gran cadena de 0 a 200 seccionada en dos partes 0-100 y 101-200.

```
int resultado =
grandinico(hCam,IS_SET_TRIGGER_SOFTWARE,IS_SET_DM_DIB,IS_CM_MONO16,10,83);
//Iniciación de cámara
ptrcmemoria mems; //Espacio de Memorias Número 1
ptrcmemoria mems2; //Espacio de Memorias 2
mems = NULL;
mems2 = NULL;
resultado = seccionlistacmemoria(hCam,mems,100,0,16);
resultado = seccionlistacmemoria(hCam,mems2,100,100,16);
std::cout<<mems->numeroimg<<"\n";
std::cout<<mems2->numeroimg<<"\n";
```

*tamanolistacmemoria(ptrcmemoria iniciolista)*

- **Labor:** El método toma un puntero `ptrcmemoria` y retorna la cantidad de estructuras `cmemoria` ligadas. Efectivamente da el tamaño de una lista dinámica.
- **Entradas**
  - **Iniciolista:** Puntero a estructura `cmemoria` sobre la cual empezar el conteo de lista dinámica. No hay problema en caso de ser un valor `NULL`. Cabe destacar que el método cuenta a partir del puntero dado pero no tiene conocimiento de los posibles elementos previos.
- **Retorna:** Retorna el número de estructuras `cmemoria` en la lista dinámica. El valor de retorno es una variable tipo `int`.
- **Funciones uEye:** Ninguna
- **Notas:** N/A
- **Ejemplo:** En el ejemplo se crea una lista de memorias con el método `seccionlistacmemoria`, luego se realiza un conteo de la cantidad de estructuras `cmemoria` hay en dicha lista y se imprime en pantalla. El valor impreso debe ser igual al número de la variable "cantidad".

```
ptrcmemoria mems; //Espacio de Memorias Número 1
mems = NULL;
int cantidad = 10; //Cantidad de imágenes a tomar
resultado = seccionlistacmemoria(hCam,mems,cantidad,0,16);
int cantidadreal = tamanolistacmemoria(mems);
```

```
std::cout<<"La cantidad real de memorias es: "<<cantidadreal;
```

`capt(HIDS Cam,ptrcmemoria &lista,int cantidad)`

- **Labor:** El método `capt` se encarga de tomar una cantidad específica de imágenes, almacenando cada una en las estructuras de memoria de una lista dinámica. Altera además la propiedad `capturaOK` de las estructuras `cmemoria`.
- **Entradas**
  - `Cam`: El identificador de la cámara en uso, debe ser un identificador de una cámara inicializada por el método `grandinicio`.
  - `Lista`: Puntero a una estructura `cmemoria` que sirva de inicio para la lista dinámica de memorias.
  - `Cantidad`: Cantidad de imágenes que se desean tomar.
- **Retorna:** Al ser una función tipo `void` no retorna un valor formal propio; no obstante el espacio de memoria de cada estructura `cmemoria` contiene una imagen con el formato dado en “`colormode`” del método `grandinicio`. Además en las estructuras de memoria donde la imagen se toma adecuadamente el valor de `capturaOK` se cambia a 1. (Esto permite conocer cuales estructuras `cmemoria` poseen una imagen asociada en memoria y cuales tienen un espacio de memoria reservado pero sin imagen).
- **Funciones uEye:** Se usan las funciones `is_SetImageMem` y `is_FreezeVideo`
- **Notas:** La cantidad de memorias asociadas al puntero “`lista`” **DEBE ser igual o mayor al valor de `cantidad`**, caso contrario el método imprimirá en pantalla un mensaje de error apuntando este error; se marca en código con el comentario `//$$$$$$$$$$$$$` el momento donde se imprime dicho mensaje de error en caso de que se desee modificar.
- **Ejemplo:** En el ejemplo se observa cómo se crea la lista de memorias con el método `seccionlistacmemoria` y luego se realiza la captura de imágenes, finalmente se imprime en pantalla el valor `capturaOK` de la primera estructura `cmemoria`. Se debe poner énfasis en que la variable “`cantidad`” se reutiliza para evitar problemas.

```
ptrcmemoria mems; //Espacio de Memorias Número 1
mems = NULL;
int cantidad = 100;
resultado = seccionlistacmemoria(hCam,mems,cantidad,0,16);
capt(hCam,mems,cantidad);
std::cout<<mems->capturaOK<<"\n";
```

`replaceChar(std::string str, char chRemover, char chNuevo)`

- **Labor:** Función que se encarga de sustituir un primer carácter por otro en una cadena `string` definida.
- **Entradas**
  - `Str`: `String` sobre el cual operar
  - `chRemover`: Variable `char` del carácter a remover
  - `chNuevo`: Carácter con el cual sustituir a `chRemover`
- **Retorna:** El método retorna el `string str` modificado.

- **Funciones uEye:** N/A
- **Notas:** El método se utiliza principalmente en el método `crearcarpeta`; no obstante queda disponible al usuario.
- **Ejemplo:** En el código el string prueba que contiene el mensaje “Esto:es:un:ejemplo” se introduce en el método `replaceChar` para cambiar todos los “:” por “\_”, el resultado se almacena en la variable `cambio`. Al correr el código se debe ver en pantalla el mensaje “Esto\_es\_un\_ejemplo”

```
std::string prueba = "Esto:es:un:ejemplo";
std::string cambio = replaceChar(prueba, ':', '_');
std::cout<<cambio;
```

#### *crearcarpeta()*

- **Labor:** El método crea una carpeta cuyo nombre sigue el formato Día\_Mes\_#Día\_Hora\_Minuto\_Segundo\_Año. Y retorna al usuario el nombre de la carpeta para su uso en otras funciones.
- **Entradas:** No tiene.
- **Retorna:** Retorna una variable `std::string` que contiene el nombre de la carpeta.
- **Funciones uEye:** N/A
- **Notas:** La carpeta se crea de forma local, es decir se localiza en la misma dirección donde se encuentre el ejecutable del programa que haga uso del método.
- **Ejemplo:** Considerando que el uso del método es muy directo, se plantea el ejemplo para entender el formato del nombre de la carpeta creada. Asumiendo que fuera el día Miercoles 21 de Diciembre del 2016 a las 10:48:2 (hora-minutos-segundos). La carpeta se generaría con el nombre. “Wed\_Dec\_21\_10\_48\_02\_2016”; además asumiendo que el ejecutable se localiza en el escritorio, la carpeta se crearía en el escritorio. En el código planteado el mismo nombre se imprimiría en pantalla, demostrando así que la variable “fecha” almacena el nombre de la carpeta.

```
std::string fecha= creacarpeta();
std::cout<<"Nombre carpeta: "<<fecha;
```

#### *save(HIDS Cam,ptrcmemoria &lista,int cantidad, std::string carpeta, UINT FileType)*

- **Labor:** Almacena una cantidad definida de imágenes de una lista dinámica de estructuras `cmemoria` en una carpeta dada. Efectivamente pasa las imágenes de memoria RAM a la memoria de disco
- **Entradas**
  - Cam: El identificador de la cámara en uso, debe ser un identificador de una cámara inicializada por el método `grandinicio`.
  - Lista: Puntero a una estructura `cmemoria` que sirva de inicio para la lista dinámica de memorias.
  - Cantidad: Cantidad de imágenes que se desean guardar

- Carpeta: String con el nombre de la carpeta en la cual se desean guardar las imágenes, la carpeta debió ser creada previamente, sea por métodos externos o con el método creacarpeta de esta biblioteca.
- FileType: Define el formato de imagen a guardar, asume que se utiliza una de las tres opciones definidas en la biblioteca uEye.h. IS\_IMG\_JPG, IS\_IMG\_BMP o IS\_IMG\_PNG para los formatos jpg, bmp y png respectivamente.
- **Retorna:** Al ser tipo void no retorna ningún valor; no obstante, las estructuras cmemoria donde las imágenes se guardan adecuadamente cambian su parámetro guardarOK a 1.
- **Funciones uEye:** is\_ImageFile
- **Notas:** La cantidad de memorias asociadas al puntero “lista” **DEBE ser igual o mayor al valor de cantidad**, caso contrario el método imprimirá en pantalla un mensaje de error apuntando este error; se marca en código con el comentario //\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$\$ el momento donde se imprime dicho mensaje de error en caso de que se desee modificar. Existe una relación entre el formato de color dado en el método “grandinicio” con la propiedad colormode y la propiedad FileType; no todo valor de “colormode” puede almacenarse en los formatos de FileType, el proveedor no da mayor información de este fenómeno. De forma empírica se conocen las siguientes reglas.
  - **PNG:** Admite todos los “colormode” posibles
  - **BMP:** Admite únicamente los colormode con profundidad de píxel (bpp) de 8. EJ: IS\_CM\_MONO8
  - **JPG:** Admite formatos de profundidad de píxel de 8 y 10 bits.
- **Ejemplo:** En el ejemplo se crea una lista de cmemorias en la variable mems, se toman 10 imágenes con el método capt, se crea una carpeta con el método creacarpeta almacenando su nombre en el string “fecha” y se corre el método de guardado (usando el espacio de memorias mems y usando la variable “cantidad” asegurando que la cantidad a guardar sea la misma que la de capturas). Se debería de tener en la carpeta creada 10 imágenes en formato png con profundidad de píxel de 16 bpp. **EXTRA:** Para probar más el método se puede cambiar “cantidad” por “cantidad+1” en el método save, esto provocaría que el mensaje de error se imprima en pantalla; se puede cambiar por “cantidad-1” y observar cómo se almacenan solo 9 imágenes correctamente.

```
ptrcmemoria mems; //Espacio de Memorias Número 1
mems = NULL;
int cantidad = 10; //Cantidad de imágenes a tomar
std::string fecha= creacarpeta();
resultado = seccionlistacmemoria(hCam,mems,cantidad,0,16);
capt(hCam,mems,cantidad); //Captura de imágenes
save(hCam,mems,cantidad,fecha,IS_IMG_PNG);
```

*Datos\_Save*(HIDS Cam, std::string carpeta, double ang, double CantLineas, double Tc,ptrcmemoria lista, double expo)

- **Labor:** El método almacena la información dada en las entradas. Se guarda la información en un .txt de nombre DATOS#, donde # depende de la información en la entrada “lista”. El txt contiene la fecha de captura seguida de una variedad de números. Todo archivo DATOS#.txt respeta la siguiente forma:

- Nombre de carpeta
- Ángulo de Barrido
- Cantidad de Líneas
- Tiempo de Captura
- Velocidad de Barrido
- Tiempo de Exposición
- Ganancia
- Imagen con Error 1
- Imagen con Error n
- **Entradas**
  - **Cam:** El identificador de la cámara en uso, debe ser un identificador de una cámara inicializada por el método `grandinicio`.
  - **Carpeta:** Variable `std::string`, es el primer elemento en aparecer en el `.txt`. En su diseño se concibió que el nombre de la carpeta que contiene a las imágenes se almacene en esta variable; no obstante, en la práctica permite guardar cualquier texto que el usuario desee como primera línea.
  - **Ang:** Variable `double`, es el segundo valor en aparecer en él `.txt`. Se diseñó pensando en almacenar el ángulo de barrido (grados) de una captura de imágenes; no obstante, en la práctica permite almacenar algún número `double` en la segunda línea del `txt`.
  - **CantLineas:** Variable `double`, forma parte del tercer valor en aparecer en él `.txt`. Se diseñó pensando en almacenar la cantidad de líneas de una captura de imágenes; no obstante, en la práctica permite almacenar algún número `double` como parte de la tercera línea del `txt`. **Ver notas para aclaración importante de la tercera línea del txt.**
  - **Tc:** Variable `double`, es el cuarto valor en aparecer en él `.txt`. Se diseñó pensando en almacenar el tiempo de captura (s) de una única imagen; no obstante, en la práctica permite almacenar algún número `double` en la cuarta línea del `txt`.
  - **Expo:** Variable `double`, es el sexto valor en aparecer en él `.txt`. Se diseñó pensando en almacenar el tiempo de exposición (ms) de la cámara; no obstante, en la práctica permite almacenar algún número `double` en la sexta línea del `txt`.
  - **Lista:** Puntero a la primera estructura `cmemoria` de una lista dinámica, dentro del método se analizan los parámetros “`capturaOK`” y “`guardarOK`” para definir si una imagen se capturó y almaceno con éxito, si alguno de los valores es “0” se almacena el identificador “`numeroimg`” de la estructura en el `.txt`. Abarca del octavo espacio en adelante del `.txt`. **El nombre del txt es Datos#, donde # es el identificador “numeroimg” de la primera estructura cmemoria de la variable lista.**
- **Retorna:** El método no retorna ningún valor no obstante en caso de no poderse abrir el archivo `txt` se imprime un mensaje de error, en el código se marca con `//$$$$$$$$$$$$$$$$$` la impresión del mensaje de error en caso de que se desee modificar el método para tomar otras acciones.
- **Funciones uEye:** `is_SetHWGainFactor`
- **Notas:** Se debe aclarar ciertas particularidades de las líneas 3, 5 y 7 del archivo `.txt`.
  - Línea 3: El valor almacenado es la cantidad real de líneas es decir *CantLineas* – *LineasError* donde *LineasError* es la cantidad de líneas dentro de la lista dinámica “*lista*”

que fallaron sean en captura o almacenamiento de información. Por ese motivo si se usa la variable CantLineas para guardar otro valor se debe considerar este efecto.

- Línea 5: Dado que a la hora de diseño se asumió que “ang sería el ángulo de barrido”, “Cantlineas sería la cantidad de líneas del barrido” y “Tc es el tiempo de captura de una imagen”. Se asignó la quinta línea para la velocidad de barrido que respeta la fórmula:  $Velocidad = \frac{ang}{Tc * CantLienas}$ , si se utilizan los parámetros ang, CantLineas y Tc para guardar otros valores se debe entender que la quinta línea siempre aplicara esta función a dichos valores. Se comenta con la forma //XXXXXXXXXXXXXXXXX la línea de código donde se hace la operación en caso de que el usuario desee cambiar la fórmula por otra.
- La séptima línea es el tiempo de exposición el cual se toma directamente de la cámara en tiempo real, dado que la biblioteca Control\_Pushbroom\_UI3360\_CP no contempla método de alterar la ganancia se espera que este valor siempre sea de 1, el valor defecto.

Por último cabe aclarar que el método Datos\_Save fue diseñada considerando un tipo de barrido de la cámara pushbroom por velocidad constante, donde se asigna una velocidad al elemento encargado del barrido y la cámara toma fotos en dicho trayecto. Las variables ang, CantLineas y Tc respaldan este hecho y es deber del usuario definir las mismas según la forma en que implemente el barrido por movimiento constante.

- **Ejemplo:** En el ejemplo asumiendo que la variable “mems” apunta a una lista dinámica de 10 elementos, de los cuales el 3 (con numeroimg = 3) y el 7 (con numeroimg = 7) fallaron (es indiferente si fallaron en captura o guardado) y el 1 tiene asignado “numeroimg=1” se almacenará la siguiente información en un archivo denominado Datos1.txt
  - Día\_Mes\_Día#\_Hora\_Minuto\_Segundo\_Año (a como se explicó en el método crearcarpeta)
  - 5
  - 8
  - 0.1
  - 5
  - 83
  - 1
  - 3
  - 7

```
std::string fecha= crearcarpeta();  
Datos_Save(hCam, fecha, 5, cant, 0.1, mems, 83);
```

```
destrucCmem(HIDS Cam, ptrcmemoria &lista)
```

- **Labor:** Elimina toda una lista dinámica, retornando al sistema toda la memoria RAM asignada así como la memoria utilizada en los punteros que unen la lista.
- **Entradas**
  - El identificador de la cámara en uso, debe ser un identificador de una cámara inicializada por el método grandinicio.

- Lista: Puntero a una estructura cmemoria que sirva de inicio para la lista dinámica de memorias.
- **Retorna:** No retorna ningún valor directo, no obstante la variable “lista” se re-assigna a NULL ya que toda la lista dinámica se elimina.
- **Funciones uEye:** is\_FreelImageMem
- **Notas:** Se elimina toda la cadena a partir del puntero lista, las memorias previas a las del puntero lista no se eliminan.
- **Ejemplo:** En el código se genera una lista de una única memoria con el método seccionlistacmemoria y se imprime en pantalla el identificador numeroimg, se procede a borrar la lista con el método destrucCmem y se imprime un mensaje para este efecto. Los if permiten notar la existencia o no de elementos en la lista.

```
ptrcmemoria mems; //Espacio de Memorias Número 1
mems = NULL;
seccionlistacmemoria(hCam, mems, 1, 0, 16);
if(mems !=NULL){
    std::cout<<"\nIdentificador: "<<mems->numeroimg;
}
destrucCmem(hCam, mems);
if(mems==NULL){
    std::cout<<"\nSe borró toda la memoria";
}
```

#### Check(ptrcmemoria lista)

- **Labor:** Función que recorre una lista dinámica de memorias y almacena en dos variables internas cuales estructuras fallaron en la captura o guardado de imágenes.
- **Entradas**
  - Lista: Puntero a una estructura cmemoria que sirva de inicio para la lista dinámica de memorias.
- **Retorna:** Propiamente no retorna ningún valor, no obstante imprime en pantalla la lista de imágenes que fallaron y el motivo bajo el que fallaron, si captura o guardado. En el código se marca con //\$ la zona donde se imprime la lista en caso de que el usuario desee modificar el código. La lista de estructuras que fallaron en la captura de imágenes se denomina listaerrorcaptura y la lista de estructuras que fallaron en el guardado de imágenes se denomina listaerrorguardado.
- **Funciones uEye:** N/A
- **Notas:** N/A
- **Ejemplo:** En el código se genera una lista de memorias y sin realizar ninguna captura o similar se pasa a través del método check, dado que ninguna estructura cmemoria posee un parámetro capturaOK=1, el método check imprime en pantalla todos los campos como campos con error.

```
ptrcmemoria mems; //Espacio de Memorias Número 1
mems = NULL;
seccionlistacmemoria(hCam, mems, 10, 0, 16);
check(mems);
```

## Ejemplo General

Finalmente se presenta el ejemplo de un proceso de inicialización, captura y almacenamiento de imágenes completo con la explicación del mismo. Se presenta el código primero y luego la explicación. Cabe destacar que es la forma más directa de usar la biblioteca más no la única.

```
HIDS hCam = 1; //Identificador de cámara
int resultado =
grandinico(hCam,IS_SET_TRIGGER_SOFTWARE,IS_SET_DM_DIB,IS_CM_MONO16,10,83);
//Inicialización de cámara
if(resultado ==0 ){
    int cant = 70; //Cantidad de imágenes a tomar
    ptrcmemoria mems; //Espacio de Memorias Número 1
    mems = NULL;
    resultado = seccionlistacmemoria(hCam,mems,cant,50,16); //Genera lista de memorias
    if(resultado == 1){ //Analiza que la lista de memorias no sufra de error
        destrucCmem(hCam,mems);
    }
    else{
        capt(hCam,mems,cant); //Captura de imágenes
        std::string fecha= creacarpeta();
        save(hCam, mems, cant, fecha, IS_IMG_PNG);
        check(mems);
        Datos_Save(hCam,fecha,50,cant,0.1,mems,83);
        destrucCmem(hCam,mems);
    }
}
```

Inicialmente se inicia la cámara con el método grandinico, en caso de que exista algún error (dado por el valor de retorno del método grandinico) se evita el resto del programa, caso contrario se continua la ejecución. Seguidamente se genera una lista dinámica de memorias, variando el valor de la variable “cant” se puede alterar la cantidad de estructuras cmemoria, si por alguna razón no se logra generar toda la lista de memorias se evita el resto del programa y se elimina la sección de lista generada. Caso contrario se procede con la captura de imágenes, la creación de la carpeta, el almacenamiento, el chequeo de las estructuras, el guardado de información en el .txt y la destrucción de las memorias.

Cabe destacar la relación entre la forma en que se inicializa la memoria, el valor de bits de profundidad por píxel y el formato de guardado. El colormode en el método grandinico es IS\_CM\_MONO16, por lo que el método seccionlistacmemoria se debe aplicar con bpp = 16, por otro lado en el método “save” se usa un formato que sea capaz de manejar información con 16 bits de profundidad por píxel, es decir png.

Por otro lado la variable cant que define la cantidad de memorias a crear en el método seccionlistacmemoria se reutiliza en todas las funciones que necesiten una cantidad de estructuras cmemoria a utilizar (capt o save por ejemplo).

Como el valor de “inicio” en el método seccionlistacmemoria es de 50, se puede esperar que el archivo txt generado por Datos\_Save posea el nombre “Datos51.txt”

En caso de que fallen algunas capturas o guardados se presentara en pantalla un mensaje con el número de estructura que fallo (gracias a el método check), además las líneas 8 en adelante del archivo Datos51.txt contendrán la misma información.

#### 9.4. Biblioteca “Control\_Snapshot\_Gii.h”

##### Descripción

La biblioteca “Control\_Snapshot\_Gii.h” permite establecer comunicaciones con el hiperespectrómetro UHD-185 y enviar comandos para operar la cámara.

##### Requisitos y Consideraciones

Al utilizar la biblioteca se debe tener en cuenta los siguientes temas

1. La comunicación con la cámara se basa en conexión TCP/IP por lo que no es necesario que el equipo de control tenga los drivers de la cámara UHD-185 instalados; no obstante el equipo que se conecta directamente a la cámara si los requiere. La conexión TCP/IP logrará que se pueda controlar la cámara por una red wifi o bien directamente en el equipo conectado a la cámara.
2. Por defecto la IP de la cámara UHD-185 es 192.168.10.100 y el puerto de comunicación TCP/IP es 3000.
3. Para que la biblioteca funcione se debe iniciar el programa C-Link que controla la cámara. En caso de utilizar el Mini-PC fitlet-iA10 esto implica tres opciones. I) Conectar el Mini-PC a una pantalla por el puerto HDMI e iniciar el programa directamente. II) Conectar algún tercer ordenador con la aplicación C-Link mediante conexión wifi al fitlet-iA10 lo que inicia la aplicación C-link del fitlet. III) Asignar que el programa C-Link se inicie de forma automática cuando el fitlet enciende, en este caso se debe tener cuidado de conectar la UHD-185 antes de encender el Mini-PC
4. La biblioteca se diseñó para C++ mediante Microsoft Visual Studio Express 2012.

##### Explicación de Funciones

Se procede a explicar las funciones, se enuncia el nombre del método, sus parámetros de entrada, sus salidas, cualquier detalle de cuidado adicional y un ejemplo de su uso.

*InitSocket\_UHD(PCSTR DireccionIP, addrinfo\* &PosibleComunicaciones)*

- **Labor:** Se encarga de iniciar los procesos necesarios para la comunicación TCP/IP, así como obtener la información para conexión según una dirección IP específica.
- **Entradas**
  - DireccionIP: Variable PCSTR en la cual se asigna la dirección IP con la cual conectar. El método busca todos los dispositivos con los cuales se puede comunicar con dicha dirección.



- **Notas:** La comunicación se inicia y se acaba a lo largo del método, es decir por cada comando enviado se inicia y cierran comunicaciones; esto por la particularidad de cómo opera el UHD-185. Se adjunta con la biblioteca el pdf "Command List Edelweiss 1.5.5" en la cual se encuentra el formato y explicación de una variedad de posibles comandos a enviar. **Se recomienda observar el manual ya que el control de la cámara depende más del formato de los comandos que del método sending propiamente**, el método sending facilita el envío de instrucciones.
- **Ejemplo:** Se observa inicialmente como la variable "result" se inicializa en el método InitSocket\_UHD para luego utilizarse como parámetro en el método sending. Por otro lado el mensaje de la variable sendbuf provoca que la cámara asigne un tiempo de exposición de 83 ms.

```
char *sendbuf = ""; //Crea mensaje general
struct addrinfo* result = NULL;
int fact; //Variable para pausas y selección de opciones
fact = InitSocket_UHD("192.168.10.100",result); //Inicio de comunicaciones y almacena en
result las posibles direcciones a conectar
if(fact==0){
    sendbuf = "<Cmd>SetInt=\"83\"</Cmd>";
    int envio = sending(result,sendbuf);
    if(envio == 0){
        std::cout<<"Se envió adecuadamente el comando";
    }
}
}
```

Exporting(addrinfo \*result)

- **Importante:** El método exporting se diseñó para pruebas del proceso de exportación y **no es una función completa para uso continuo**, se puede usar como estructura en caso de querer enviar múltiples comandos o bien para conocer el proceso de exportación.
- **Labor:** Formalmente realiza el proceso de exporta una imagen del fichero .cub a una serie de imágenes tiff; no obstante siempre exporta la información de un cubo denominado Prueba000 de una carpeta
- **Entradas**
  - Result: Contiene la lista de conexiones posibles, debe provenir del resultado de una llamada al método InitSocket\_UHD
- **Retorna:** Retorna 1 en caso de fallos en la ejecución y 0 en caso de una ejecución correcta.
- **Notas:** El método no se debe usar por sí misma, se debe tomar como estructura en caso de que se desee enviar una secuencia específica de comandos. Se puede observar el orden de los comandos enviados en caso de que se desee conocer el proceso de exportación.
- **Proceso de Exportación:** Se enumeran los comandos a enviar en caso de querer realizar la exportación de datos cub a una variedad de imágenes .tiff.
  - "<Cmd>SetPanInterpolation=\"HighQualityBilinear\"</Cmd>" //Asigna estilo de interpolación en caso de utilizar PanScale distintas de 1
  - "<Cmd>SetExportLabels=\"0\"</Cmd>"
  - "<Cmd>SetPanScale=\"20\"</Cmd>" //Asigna nivel de difuminación de imagen hiperspectral 50x50 en imagen escala grises 1000x1000
  - "<Cmd>SetExportFirstgray=\"1\"</Cmd>" //Define que se debe exportar la imagen en gris

- "`<Cmd>SetExportSelection=\0;1;2;3;4\</Cmd>`" //Define cuales líneas hiperespectrales se exportan, los valores puede continuar hasta 127
- "`<Cmd>SetSpecFactor=1\</Cmd>`"
- "`<Cmd>Load="C:/Users/admin/Documents/Cubert/2016_12_23/Prueba000"\</Cmd>`" /Carga la imagen a enviar en este caso una de nombre Prueba000 almacenada en la carpeta 2016\_12\_23 del directorio Cubert (donde se almacenan por defecto las capturas del UHD\_185)
- "`<Cmd>ExportTiff="C:/Users/admin/Desktop/Prueba/\</Cmd>`" //Comando propio de exportación, recibe por parámetro la dirección donde se guardarán las imágenes .tiff resultantes.

### Para Diseño de Otras Bibliotecas

Considerando que la biblioteca se diseñó para C++ y específicamente mediante Microsoft Visual Studio Express, se enumeran las distintas consideraciones en caso de querer generar otras bibliotecas para otros sistemas operativos o lenguajes de programación.

1. El único requisito para poder comunicarse con la cámara es poder generar una conexión TCP/IP, sea cual sea el método lograr este tipo de conexión asegura el poder operar a cámara.
2. La dirección IP por defecto es 192.168.10.100 y el puerto de conexión es el 3000.
3. Aun cuando los comandos poseen formato XML, se deben enviar como una cadena de texto, respetando cada carácter exacto sean <, >, " o \.
4. Es necesario iniciar comunicaciones cada vez que se desea enviar un comando y cerrar la conexión una vez recibida la respuesta del hiperespectrómetro.

## 9.5. Biblioteca "Control\_SBGC\_GII.h"

### Descripción

La biblioteca "Control\_SBGC\_GII.h" permite operar el controlador SBGC de la compañía Basecam localizado en el gimbal CAME Mini 3-air. Dados los requisitos del proyecto SIMMAP la biblioteca diseñada controla solamente las funciones de encendido y apagado de motores y el envío de comandos.

### Requisitos y Consideraciones

Para poder utilizar la biblioteca es necesario realizar las siguientes labores.

1. Si se trabaja con el gimbal CAME Mini 3-air. **NO ACTUALIZAR EL FIRMWARE.**
2. Instalar el driver CP210xVCPInstaller adjunto a la biblioteca en la carpeta "Driver Comunicación Serial". También descargable de <http://www.silabs.com/products/mcu/pages/usbtouartbridgevcpcdrivers.aspx>
3. El proyecto que haga uso de la biblioteca debe también adjuntar los .h llamados: "SBGC.h", "SBGC\_cmd\_helpers.h", "SBGC\_command.h", "inttypes.h" y "stdint.h". Todos contienen definiciones necesarias para que la biblioteca diseñada opere debidamente.

4. La biblioteca se diseñó para el firmware 2.56b9 del SBGC, no se asegura que la biblioteca opere adecuadamente con versiones futuras del firmware.
5. La biblioteca fue diseñada en Microsoft visual Studio Express 2012 para C++, no se asegura su funcionalidad en sistemas operativos Linux.
6. Las funciones implementadas de la Serial API del SBGC son CMD\_MOTORS\_ON, CMD\_MOTORS\_OFF y CMD\_CONTROL. Se adjunta el manual de comunicación serial en caso de que se quiera comprender a más profundidad su labor o demás.

## Explicación de Funciones

Se procede a explicar las tres funciones, se enuncia el nombre del método, sus parámetros de entrada, sus salidas, su utilidad y cualquier detalle de cuidado adicional.

*Initserial*(HANDLE &hComm, std::string Puerto, int BaudRate, int Paridad, int BitParada)

- **Labor:** Se encarga de iniciar la comunicación serial a través de un puerto dado por el usuario y con características determinadas por el usuario.
- **Entrada**
  - HANDLE “&hCam”: Variable tipo HANDLE que sirve de identificador, no requiere ningún valor específico sino solamente que la variable como tal, el método por su cuenta se encarga de asignarle un valor.
  - Std::string “Puerto”: Nombre del puerto al cual se desea establecer al comunicación serial, el usuario debe definir por su cuenta el puerto a utilizar y pasar su nombre en el formato std::string. EJ: “COM4”
  - Int Baudrate: El baudrate que se desea, se recuerda utilizar los valores definidos para este caso. EJ: 9600, 115200, entre otros. El método no permitirá el inicio de comunicaciones si se introduce un valor no estándar.
  - Paridad: Variable tipo int que define la paridad a usar, se puede usar las definiciones propias de la biblioteca Windows.h. EJ: NOPARITY/0
  - BitParidad: Variable tipo int que define los bit de parada, similar al caso de paridad se puede usar las definiciones de la biblioteca Windows.h. EJ: ONESTOPBIT
- **Retorna:** Dado que el tipo del método es “void” no retorna ningún valor propiamente; no obstante se debe aclarar que la variable hCam se ve modificada y adquiere el valor del HANDLE de la comunicación, esta variable se utiliza en las funciones siguiente. Por otro lado el método imprime en pantalla mensajes de éxito o fallo de las comunicaciones, en el código se pone un comentario de la forma //\$\$\$\$\$\$\$\$\$\$\$\$\$ donde se imprimen dichos mensajes, de esta forma en caso de que se desee modificar los mensajes de éxito y fallo se puede distinguir en que zona colocarlos.
- **Notas:** En el caso del gimbal CAME Mini 3 air las entradas del método deben ser
  - Puerto: VARIA
  - Baudrate: 115200

- Paridad: NOPARITY
- BitParada: ONESTOPBIT
- **Ejemplo:** En el ejemplo se inicia una comunicación serial a través del puerto COM4, un baudrate de 115200, sin paridad y un bit de parada. Cabe destacar la forma en que se utiliza el handle hComm, es necesario solo crear el handle para poder utilizarlo en el método.

```
HANDLE hComm;
std::string com = "COM4";
initiserial(hComm,com,115200,NOPARITY,ONESTOPBIT);
```

*EnvioComandoControl*(HANDLE hComu, float ModoControl, float VelRol, float AngRol, float VelPitch, float AngPitch, float VelYaw, float AngYaw)

- **Labor:** Se encarga de enviar el comando de control de gimbal, según las entradas esto permite desplazar el gimbal a una orientación específica, iniciar un desplazamiento a velocidad constante sobre un eje (cabeceo, alabeo o guiñada), entre otros.
- **Entrada**
  - **HANDLE hComu:** El handle de la comunicación serial, en este caso el handle debe haber pasado por el método initiserial previamente ya que se asume que hComu representa un handle de una comunicación inicializada.
  - **ModoControl:** Un número que especifica el modo de control, puede adquirir los valores de:
    - 0/MODE\_NO\_CONTROL: Elimina cualquier orden previa y retorna el gimbal a su operación estándar.
    - 1/MODE\_SPEED: Inicia el movimiento a una velocidad constante sobre cada eje según lo que digan las variables VelRol/Pitch/Yaw, se ignoran las variables “Ang”
    - 2/MODE\_ANGLE: El gimbal cambia su orientación en los ejes cabeceo (pitch), alabeo (roll) y guiñada (yaw) a la velocidad indicada por las variables AngRoll/Pitch/Yaw. La velocidad de movimiento lo definen las variables Vel, en caso de asignar 0 a en alguna variable Vel se utiliza la velocidad asignada en la GUI del SBGC.
    - 3/MODE\_SPEED\_ANGLE: El gimbal se desplaza a una velocidad constante dada por las variables Vel pero manteniendo la orientación exigida por las variables Ang.
    - 4/MODE\_RC: Los parámetros de Ang se imponen con mayor prioridad sobre los de la comunicación a control remoto.
    - 5/MODE\_ANGLE\_REL\_FRAME:CONSULTAR MANUAL DE COMUNICACIÓN SERIAL
  - **VelRol/Pitch/Yaw:** Tres variables que implican lo mismo pero en distintos ejes, es la velocidad a la que se desea mover el gimbal. Asume que los valores se dan en grados/segundos
  - **AngRol/Pitch/Yaw:** Variable de ángulo al que se desea desplazar el gimbal, respectivamente son Alabeo(rol), Cabeceo (pitch) y guiñada(yaw). Asume que el valro se da en grados.

- **Retorna:** El método es de tipo int y retorna un 1 en caso de que el mensaje se entregó con éxito y un 0 en caso de que se dio un error. Además imprime mensajes de éxito y fallo en pantalla para este mismo efecto. Nuevamente se marca con el comentario `//$$$$$$$$` los puntos del código donde sucede esto en caso de que se desee modificar. Si por alguna razón se recibe el valor de retorno 99 esto implica que el método de envío “no funciona pero a su vez no cometió errores”, no debería de suceder un caso así pero se tiene el mensaje 99 en caso de que se dé.
- **Notas:** N/A
- **Ejemplo:** Inicialmente se muestra como la variable `hComm` debe ser la misma tanto en la inicialización como en el llamado del método. Al llamar el método se debería trabajar en el modo de ángulo desplazando el gimbal a los ángulos 10, 10 y -10 (alabeo, cabeceo y guiñada respectivamente) a una velocidad de 1, 3 y 5 grados por segundo (alabeo, cabeceo y guiñada respectivamente).

```
HANDLE hComm;
std::string com = "COM4";
initiserial(hComm,com,115200,NOPARITY,ONESTOPBIT);
```

```
EnvioComandoControl(hComm,2,1,10,3,10,5,-10);
```

```
envioMotorOn_OFF(HANDLE hComu, int Modo)
```

- **Labor:** Se encarga de encender o apagar los motores del gimbal.
- **Entrada**
  - `hComu`: El handle para comunicación serial
  - `Modo`: Variable tipo int que define si se desea encender o apagar los motores del gimbal, formalmente puede adquirir solo dos valores
    - `SBGC_CMD_MOTORS_ON/77`: Sea a variable o el número identifica el encender motores
    - `SBGC_CMD_MOTORS_OFF/109`: Sea la variable o el número identifica el apagar motores
- **Retorno:** El método es de tipo int y retorna un 1 en caso de que el mensaje se entregó con éxito y un 0 en caso de que se dio un error. Además imprime mensajes de éxito y fallo en pantalla para este mismo efecto. Nuevamente se marca con el comentario `//$$$$$$$$` los puntos del código donde sucede esto en caso de que se desee modificar. Si por alguna razón se recibe el valor de retorno 99 esto implica que el método de envío “no funciona pero a su vez no cometió errores”, no debería de suceder un caso así pero se tiene el mensaje 99 en caso de que se dé.
- **Notas:** El método no discrimina otras entradas en la variable `Modo`, es decir por defecto no rechaza valores distintos a 77 o 109, se debe tener especial cuidado por esta razón.
- **Ejemplo:** Inicialmente se muestra como la variable `hComm` debe ser la misma tanto en la inicialización como en el llamado del método. El resultado de usar el método debe ser que los motores del gimbal se apagan.

```
HANDLE hComm;
std::string com = "COM4";
initiserial(hComm,com,115200,NOPARITY,ONESTOPBIT);
```

```
envioMotorOn_OFF(hComm, SBGC_CMD_MOTORS_OFF)
```

## Explicación de Estructuras

Especialmente para el caso del método EnvioComandoControl la biblioteca SBGC\_cmd\_helpers.h provee una estructura que puede ayudar a ordenar la información. Usarla es opcional ya que el manejo de las entradas del método EnvioComandoControl se puede realizar por aparte.

*SBGC\_cmd\_control\_t*

```
// CMD_CONTROL
typedef struct {
    uint8_t mode;
    int16_t speedROLL;
    int16_t angleROLL;
    int16_t speedPITCH;
    int16_t anglePITCH;
    int16_t speedYAW;
    int16_t angleYAW;
} SBGC_cmd_control_t;
```

- El tipo de estructura es SBGC\_cmd\_control\_t
- speedROLL/speedPITCH/speedYAW: Implican lo mismo que VelRol/VelPitch/VelYaw del método EnvioComandoControl
- angleROLL/anglePITCH/angleYAW: Implican lo mismo que AngRol/AngPitch/AngYaw del método EnvioComandoControl
- Mode: Implica lo mismo que la variable Modo en el método EnvioComandoControl