



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

IMPLEMENTACIÓN DEL SISTEMA DE CAPTURA DE IMÁGENES Y PRE-PROCESAMIENTO A BORDO DEL CUBESAT SUCHAI 2

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

DIEGO ANTONIO MALDONADO MUÑOZ

PROFESOR GUÍA:
MARCOS ANDRÉS DÍAZ QUEZADA

MIEMBROS DE LA COMISIÓN:
HOLGER DRASS
CAMILO JOSÉ ROJAS MILLA

SANTIAGO DE CHILE
2022

RESUMEN DE LA MEMORIA PARA OPTAR
AL TÍTULO DE MAGÍSTER EN CIENCIAS
DE LA INGENIERÍA
POR: DIEGO ANTONIO MALDONADO MUÑOZ
FECHA: 2022
PROF. GUÍA: MARCOS DÍAZ QUEZADA

IMPLEMENTACIÓN DEL SISTEMA DE CAPTURA DE IMÁGENES Y PRE-PROCESAMIENTO A BORDO DEL CUBESAT SUCHAI 2

El siguiente trabajo muestra la implementación de un sistema de captura y procesamiento de imágenes y la integración de una cámara científica a un nanosatélite tipo CubeSat. El nanosatélite SUCHAI 2 desarrollado por el Laboratorio de Exploración Espacial y Planetaria de la Universidad de Chile, tendrá como misión, entre otras, estudiar y monitorear la contaminación lumínica en el norte de Chile.

El trabajo de integración de la cámara al SUCHAI 2 consta de dos etapas. La etapa de *hardware* considera la instalación física de la cámara, tanto la parte mecánica como la parte eléctrica, siendo esta última la que se aborda en este informe. La etapa de *software* considera el desarrollo de programas para manejar la cámara de manera remota a través del software de vuelo del satélite y también para el procesamiento previo de las imágenes a bordo del satélite.

Por otro lado, el trabajo considera pruebas de desempeño de la cámara para validar sus características técnicas según los datos indicados por el fabricante, enfocadas en caracterizar tanto el sensor dentro de la cámara como el lente que fue montado a ésta. Las pruebas tienen como objetivo validar el correcto comportamiento de la cámara en el ambiente espacial.

A la fecha, culminada la fase de desarrollo del SUCHAI 2, se considera exitosa la integración de la cámara al satélite, comprobando el sistema de captura de imágenes de manera remota y los programas de procesamiento a bordo que permitirán el envío tanto de imágenes como de meta datos a la estación terrena. A su vez, los resultados de las pruebas de caracterización del sensor validan el uso de la cámara para el cumplimiento del objetivo planteado, verificando las descripciones entregadas por el fabricante y, por sobre todo, la capacidad de la cámara como instrumento para medir y estudiar la contaminación lumínica.

Agradecimientos

Agradezco al profesor Marcos Díaz por haberme confiado el trabajo de este proyecto tan importante. A él, Samuel, Carlos, Camilo, Cristóbal, Matías y al resto del SPEL por el apoyo y conocimiento entregados durante mis sesiones de trabajo en el laboratorio.

Agradezco a Holger y a Camilo por haber participado de la comisión de mi trabajo de título, por los valiosos comentarios entregados sobre este informe y cómo mejorarlo.

Agradezco a Marco Bonati por sus valiosas enseñanzas en cuanto al funcionamiento y caracterización de la cámara, por su tiempo y disposición para ayudarme con mi trabajo. A AURA por haberme recibido en sus instalaciones en La Serena y por haber colaborado en este proyecto. A Alex y a Holger por su apoyo durante mi trabajo en AURA.

Tabla de Contenido

1. Introducción y Marco Teórico	1
1.1. Motivación y Antecedentes	1
1.2. Descripción del problema	1
1.3. Objetivos	3
1.3.1. Objetivo General	3
1.3.2. Objetivos Específicos	3
1.4. Marco Teórico	4
1.4.1. Caracterización de la contaminación lumínica	4
1.4.2. Rectificación de imágenes satelitales	6
1.4.3. Geo-referenciación de imágenes satelitales	8
1.4.4. Caracterización de un sensor CMOS	8
1.4.4.1. Ganancia	8
1.4.4.2. Capacidad Full Well	9
1.4.4.3. No linealidad	9
1.4.4.4. Eficiencia cuántica	9
1.4.4.5. Corriente oscura	10
1.4.4.6. Ruido de disparo	10
1.4.4.7. Función de dispersión de punto	10
1.4.4.8. Distorsión	11
2. El nanosatélite SUCHAI 2	12
2.1. Arquitectura de Hardware	12
2.2. Suchai Flight Software (SFS)	13
2.3. Raspberry Pi Zero W	15
2.4. Cámara pco.panda 4.2 bi UV	16
2.5. Integración de la cámara al satélite	18
2.5.1. Integración de Hardware	18
2.5.2. Integración de Software	20
3. Operación de la cámara	22
3.1. Captura de imágenes y generación de meta datos	22
3.2. Preprocesamiento y extracción de información	24
3.3. Almacenamiento y gestión de la base de imágenes	25
3.4. Recepción de datos y envío de comandos	26
4. Experimentos y Análisis	27
4.1. Configuración experimental	27
4.1.1. Monocromador	27

4.1.2. Láser	29
4.1.3. Calibración de la cámara	29
4.2. Ciclo de operación	30
4.3. Pruebas de vacío	32
4.4. Ganancia	34
4.5. Linealidad	36
4.6. Eficiencia Cuántica	37
4.6.1. Relativa	37
4.6.2. Absoluta	39
4.7. Ruido de disparo	40
4.8. Point spread function PSF	41
4.9. Calibración	42
5. Conclusiones	45
5.1. Integración de la cámara	45
5.2. Caracterización del sensor	45
5.3. Trabajo futuro	46
Bibliografía	47
Anexo A. Escala de Bortle	49
A.1. Escala de Bortle	49

Índice de Tablas

2.1.	Especificaciones generales de los dispositivos considerados	16
2.2.	Especificaciones técnicas de la cámara pco.panda 4.2 bi UV	17
3.1.	Lista de comandos de la cámara PCO	23
3.2.	Directorio de datos de la cámara pco	23
3.3.	Archivo de metadatos asociados a una secuencia de imágenes	24
3.4.	Lista de comandos seleccionados de telemetría	26
4.1.	FPS promedio obtenido por diferentes dispositivos y tiempo de exposición t_e .	31
4.2.	Estadísticas de imágenes negras previo al vacío	33
4.3.	Estadísticas de imágenes blancas previo al vacío	33
4.4.	Estadísticas de imágenes negras posterior al vacío	33
4.5.	Estadísticas de imágenes blancas posterior al vacío	34

Índice de Ilustraciones

1.1.	Mapa de contaminación lumínica [6]	5
1.2.	Respuestas espectrales y espectros de emisión	6
1.3.	Orto-rectificación ilustrada	7
2.1.	Esquema del SUCHAI 2	13
2.2.	Diagrama del modelo UML de la arquitectura del SFS [22]	14
2.3.	Diagrama del modelo UML del patrón de comandos del SFS [22]	15
2.4.	Raspberry Pi Zero W	16
2.5.	Cámara PCO modelo panda 4.2 bi UV	17
2.6.	Placa del Star Tracker	19
2.7.	Switch USB	19
2.8.	Cámara dentro del satélite	20
4.1.	Configuración experimental 1 (<i>setup 1</i>)	28
4.2.	Cámara PCO adaptada al <i>setup 1</i>	28
4.3.	Configuración experimental 2 (<i>setup 2</i>)	29
4.4.	Imagen de prueba para calibración	30
4.5.	Tiempo entre imágenes consecutivas. Laptop (izquierda), RPI3B+ (medio), RPI0W (derecha)	32
4.6.	Ganancia del sensor	35
4.7.	Linealidad del sensor en el rango de 10 a 1000 μs	36
4.8.	Linealidad del sensor en el rango de 1 a 500 ms	37
4.9.	Eficiencia Cuántica relativa	38
4.10.	Linealidad del sensor en el rango de 1 a 500 ms	39
4.11.	Histograma normalizado sobre píxeles de la imagen media	41
4.12.	Región de incidencia de la fuente puntual	41
4.13.	Perfil gaussiano sobre la región de incidencia	42
4.14.	Región de incidencia de la fuente puntual	42
4.15.	Distorsión radial en píxeles	43
4.16.	Distorsión tangencial en píxeles	44
A.1.	Escala de Bortle	49

Capítulo 1

Introducción y Marco Teórico

1.1. Motivación y Antecedentes

La invención de la lámpara incandescente supuso un enorme cambio para el estilo de vida moderno, la generación de luz artificial por medio de electricidad combate la oscuridad de la noche y permite la continuación de las actividades humanas pasada la puesta de sol. Sin embargo, el uso indiscriminado de la iluminaria artificial ha llevado a la aparición de un nuevo problema antes inexistente, la contaminación lumínica, que corresponde al uso excesivo y despreocupado de luz artificial durante el ambiente nocturno.

Se ha demostrado en numerosos estudios cómo esta nueva amenaza afecta a los seres humanos [1] y animales [2], alterando especialmente el ritmo circadiano pero, además, entra en conflicto con las actividades que requieren de la oscuridad natural de la noche, como lo es la observación del cielo. En este sentido, no solo la astronomía que hace uso de avanzados telescopios, se ve afectada sino que además las personas ya no son capaces de apreciar el cielo nocturno en su esplendor dentro de las ciudades, la Vía Láctea por ejemplo, ha dejado de ser visible a simple vista para un tercio de la población mundial.

La amenaza que representa la contaminación lumínica para la astronomía es (o debiera ser) de gran relevancia para el mundo y en especial para Chile, considerando que en el territorio nacional se encuentran los mejores sitios para la observación del cielo a nivel mundial, albergando los mayores proyectos astronómicos y observatorios del espectro visible. Es impensable que a futuro sea imposible seguir realizando astronomía de alto nivel, lo que supondrá una gran pérdida tanto para Chile como para los países que financian estos proyectos; es por esta razón que se debe atacar este problema lo antes posible para el desarrollo de políticas que permitan proteger el cielo nocturno del norte del país.

1.2. Descripción del problema

Una de las principales herramientas para el monitoreo y estimación de la contaminación lumínica son las imágenes satelitales nocturnas, sin dejar de lado la existencia de otros métodos basados en sensores a nivel del suelo (SQM, del inglés *Sky Quality Meter*). Dichas imágenes son captadas por satélites de observación terrestre (EOS, del inglés *Earth Observation Satellite*), los cuales han sido ampliamente desarrollados por los países más avanzados en materia espacial debido a la valiosa información que aportan sobre la superficie terrestre,

en distintos espectros y en distintos momentos de captura. Las principales aplicaciones de los EOS, en un inicio, fueron de espionaje y meteorología, pero el alcance de uso de datos ha sido mayormente explotado en tiempos recientes debido a los avances tecnológicos en el desarrollo de sensores y en el área de procesamiento de imágenes, encontrando aplicaciones de alto impacto como lo son: el uso del agua, la agricultura, el estudio de uso del suelo, el monitoreo de desarrollo urbano, entre otras.

El desarrollo de estos satélites significaba una inversión considerable de dinero, tiempo y recursos humanos, sin embargo, la industria espacial se ha revolucionado en el nuevo milenio gracias a la aparición de nanosatélites estandarizados llamados CubeSat [3], cuya unidad más básica corresponde a un cubo de 10x10x10 cm con una masa de alrededor 1.33 kg (1U). Gracias a este nuevo paradigma de exploración espacial se han abaratado enormemente no sólo los costos de desarrollo y puesta en órbita de estos satélites, sino también de tiempo y recursos humanos, razón por la cual se ha masificado su uso en universidades y en países en vías de desarrollo que no contaban con la capacidad de financiar una agencia espacial propia. En particular, se ha estudiado su utilidad como EOS de bajo costo, demostrando su enorme potencial en comparación con los costosos y grandes satélites tradicionales [4].

En este contexto, el Laboratorio de Exploración Espacial y Planetaria (SPEL, del inglés *Space and Planetary Exploration Laboratory*) de la Universidad de Chile se encuentra en el desarrollo de un CubeSat con una cámara integrada, el cual se convertirá en el primer EOS diseñado y ensamblado íntegramente en Chile, el SUCHAI 2. Uno de los objetivos de esta misión espacial es justamente atacar el problema de la contaminación lumínica en el norte del país. Dicho satélite es un Cubesat de tamaño 3U (3 cubos básicos unidos de manera vertical) y será el sucesor del exitoso primer satélite desarrollado por el SPEL, el SUCHAI 1, puesto en órbita el año 2017 y que logró estar operativo por más de un año. Si bien el equipo de ingenieros cuenta con la experiencia adquirida del desarrollo de un primer satélite, la integración de la cámara como experimento científico es algo nuevo y requiere de un trabajo de integración tanto a nivel de hardware como de software para asegurar el correcto funcionamiento de ésta una vez que el satélite esté en órbita.

Lo anterior constituye solo un paso más en el desarrollo del satélite y aún existe la necesidad de desarrollar un sistema que se encargará de almacenar y procesar las imágenes captadas con el fin de que puedan ser utilizadas de manera efectiva para el cumplimiento de la misión espacial. Las imágenes captadas por el satélite deben ser procesadas para corregir todos los errores sistemáticos inherentes a las imágenes de teledetección, entre los que se encuentran la corrección geométrica debido al movimiento de satélite durante el tiempo de captura, las deformaciones a los bordes de la imagen debido a la curvatura de la tierra y el efecto del relieve de terreno, entre los más relevantes. El proceso de corrección de estos errores para obtener una orto-imagen rectificadas se denomina orto-rectificación. Por otro lado, el proceso de geo-referenciación corresponde a localizar geográficamente cada píxel de una imagen de teledetección, normalmente asignando las coordenadas de latitud y longitud. Una vez que se cuente con una base de datos de imágenes corregidas y etiquetadas, se podrá comenzar a desarrollar mapas de contaminación lumínica de las zonas de interés y, además, evidenciar cómo evolucionan en el tiempo. La capacidad de generar información de este tipo por medio de productos desarrollados nacionalmente supondrá un avance para Chile en materia espacial.

Otro aspecto relevante de estudio es la factibilidad de integrar estas fases de procesamiento de imágenes dentro de los sistemas a bordo del satélite con el fin de maximizar la cantidad de información útil enviada a la estación terrestre durante los instantes en que se establece un enlace de comunicación con la estación terrena. Los resultados obtenidos influirán en el desarrollo de futuras misiones espaciales en donde se considere contar una unidad de procesamiento embebida de alta capacidad [5].

Finalmente, cabe mencionar la posibilidad de orientar el satélite en la dirección opuesta y utilizar la cámara para capturar imágenes del espacio exterior, como otros planetas y estrellas, lo que implicaría validar su uso como telescopio espacial, constituyendo un logro de gran importancia al tratarse del primer telescopio espacial chileno.

1.3. Objetivos

El trabajo a desarrollar se constituye de distintas fases, como desarrollo de software, integración eléctrica y mecánica, caracterización de la cámara y del desempeño de los algoritmos en distintas unidades de procesamiento. Para cumplir con lo planteado se definen los objetivos específicos que se deben lograr en el camino para llegar a cumplir con el objetivo de la misión espacial, del cual se desprende el objetivo general del Trabajo de Título.

1.3.1. Objetivo General

El objetivo general del Trabajo de Título constituye en desarrollar e implementar un sistema de captura de imágenes satelitales por medio de la cámara y de pre procesamiento a bordo del nanosatélite SUCHAI 2, con el fin de cumplir el principal objetivo de la misión espacial designada al *payload*¹, que es el de proveer imágenes satelitales nocturnas con la suficiente calidad, esto es resolución espacial y temporal, para el estudio y caracterización de la contaminación lumínica en el norte de Chile.

1.3.2. Objetivos Específicos

Los objetivos específicos representan el logro de las distintas fases de desarrollo que se abordarán durante el proyecto:

- Integración de la cámara al satélite a nivel de software. Los operadores de la estación terrena deben ser capaces de comandar la cámara remotamente por medio del software de vuelo del sistema satelital.
- Integración física de la cámara al satélite. La cámara debe acoplarse al satélite cumpliendo los requerimientos eléctricos y mecánicos para asegurar su correcto encendido y operación en el espacio.
- Manejo de datos y metadatos. El sistema de procesamiento de imágenes debe administrar las imágenes capturadas y metadatos asociados de manera ordenada con el fin de facilitar la extracción de información de interés y su envío a la estación terrena.

¹ Se denomina *payload* a la carga útil a bordo de un satélite.

- Realización de experimentos de verificación de las características técnicas de la cámara según la descripción entregada por el fabricante.
- Documentar todos los resultados de los experimentos a los que fue sometida la cámara, lo que será de gran importancia no solo para el SPEL sino también para el fabricante de la cámara ya que constituye la primera herencia de vuelo de este tipo de cámara científica, lo que validará su comportamiento en el espacio para futuras misiones espaciales.

1.4. Marco Teórico

En esta sección se explicará el estudio de la contaminación lumínica y la manera en que se enfrentará este problema en Chile por medio del instrumento a bordo del SUCHAI 2. También se discutirá el problema de la rectificación de imágenes y los factores que inducen errores en la captura de imágenes satelitales para luego discutir acerca del problema de geo-referenciación y la importancia para este proyecto.

Es importante aclarar que no se cuenta con una sección de Estado del Arte dado que el Trabajo de Título no comprende el desarrollo de algoritmos de rectificación ni de geo-referenciación, lo cual está a cargo de otro equipo asociado al SPEL, pero si lo incumbe ya que el trabajo que realizará este equipo de investigación deberá ser implementado en el software del satélite.

Este Trabajo de Título comprende fundamentalmente la integración de la cámara al satélite, respecto a lo cual no existe literatura detallada debido a que la manufactura de satélites es generalmente reservada, es decir, no existen publicaciones relevantes sobre cómo construir nanosatélites ni cómo diseñarlos ni operarlos.

1.4.1. Caracterización de la contaminación lumínica

Gracias a las imágenes nocturnas proporcionadas por el satélite *Suomi National Polar-orbiting Partnership* (SNPP), un proyecto impulsado en conjunto por la Oficina Nacional de Administración Oceánica y Atmosférica (NOAA) y la Administración Nacional de Aeronáutica y el Espacio (NASA) de los Estados Unidos, los autores más citados en el área de investigación de la contaminación lumínica han desarrollado un modelo que permite caracterizar los índices de luz artificial a nivel mundial [6]. Los investigadores basan su investigación en un modelo estadístico de la emisión de radiación en dirección ascendente, que es el resultado de un ajuste por máxima verosimilitud de tres diferentes distribuciones de intensidad lumínica. A partir de este modelo y los datos han desarrollado un atlas mundial de luz artificial. En la figura 1.1.a se observa el mapa generado para la región de Sudamérica y en la figura 1.1.b se muestra un acercamiento a Chile, comprendiendo desde Calama al norte hasta Santiago al sur. La escala utilizada se encuentra en el Anexo A.1.

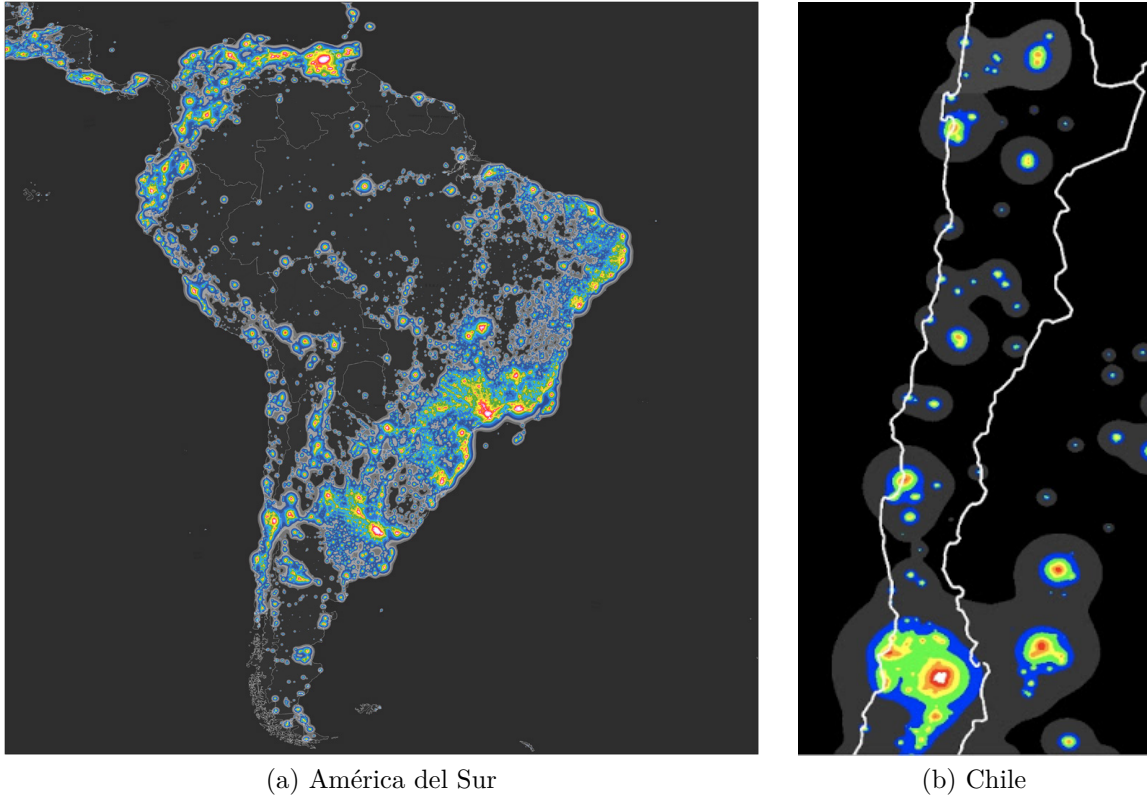
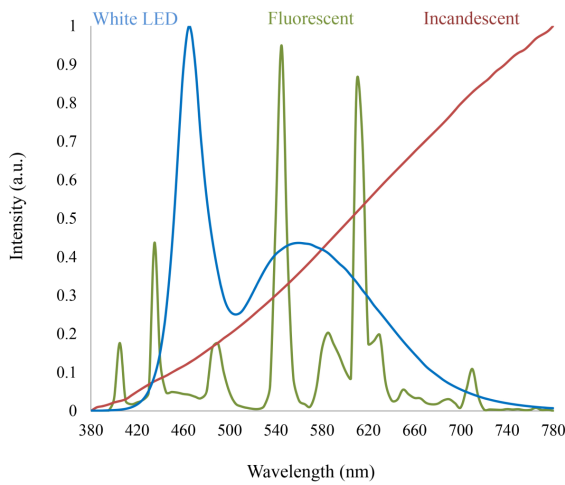


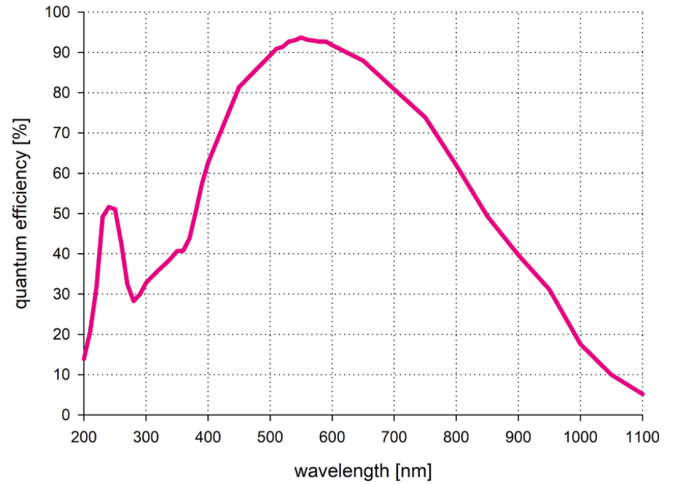
Figura 1.1: Mapa de contaminación lumínica [6]

El atlas publicado, si bien ayuda a visualizar los efectos de la contaminación lumínica incluso a kilómetros de distancia de los núcleos urbanos, no permite entender como varían las zonas afectadas en el tiempo, para lo cual se necesita de un suministro de datos con cierta resolución temporal. Por otro lado, el *Visible-Infrared Imager-Radiometer Suite* (VIIRS), instrumento a bordo del S-NPP que capta las imágenes nocturnas por medio de banda espectral *Day-Night-Band* (DNB), no puede captar un espectro importante de emisión de las luces LED blancas, cuyo uso se ha masificado en el último tiempo debido a su bajo costo de iluminación.

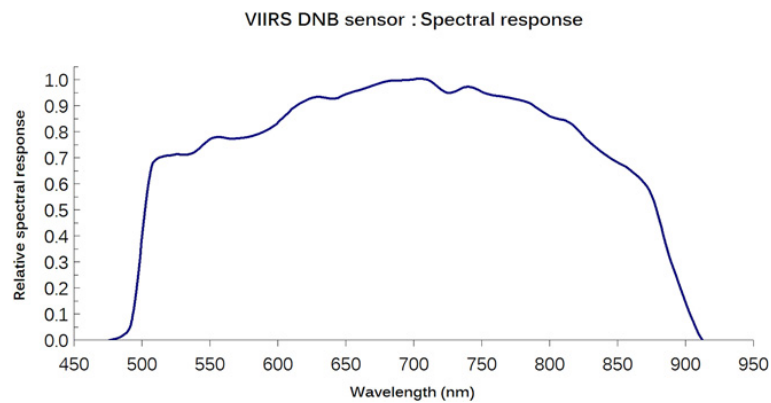
Si bien se logra un factor de eficiencia energética importante con el uso de iluminarias LED, el principal problema radica en que aquellas luces que emiten luz azul (bajo los 500nm) pueden ser altamente contaminantes si no se usan adecuadamente, puesto que la luz azul tiene un mayor índice de dispersión en la atmósfera dada su mayor frecuencia. En la figura 1.2.a se muestra el espectro de emisión de algunas fuentes de luz de uso común, donde cabe remarcar que las luces LED pueden tener su peak de emisión desde los 400nm hasta los 700nm, mientras que los LED blancos tienen su peak de emisión alrededor de los 460nm. En la figura 1.2.c se muestra la respuesta espectral del instrumento VIIRS, donde se evidencia su incapacidad de captar la luz azul bajo los 500nm.



(a) Espectro de emisión de fuentes de luz



(b) Respuesta espectral pco. panda 4.2 bi UV



(c) Respuesta espectral VIIRS

Figura 1.2: Respuestas espectrales y espectros de emisión

En este contexto, es importante notar que la cámara que se utilizará como instrumento en el SUCHAI 2 sí puede captar este espectro importante de emisión. En la figura 1.2.b se muestra la respuesta espectral de una serie de modelos del fabricante alemán PCO, de entre las cuales la pco. panda 4.2 bi UV fue la elegida para la misión espacial. Se espera que las imágenes captadas puedan ser utilizadas para contrastar los datos del VIIRS, permitiendo modelar la variación temporal de los índices de luminosidad dentro de las zonas geográficas de interés y además caracterizar la incidencia de las luces LED blancas sobre la contaminación lumínica en Chile.

1.4.2. Rectificación de imágenes satelitales

Las imágenes captadas directamente por un instrumento a bordo de un EOS tienen poca utilidad en aplicaciones prácticas sin los previos procesamientos necesarios para corregir los errores inherentes asociados a su captura. El proceso de orto-rectificación permite corregir los errores geométricos y generar una imagen planimétricamente correcta. En la figura 1.3 se muestra una ilustración de la idea detrás de la orto-rectificación.

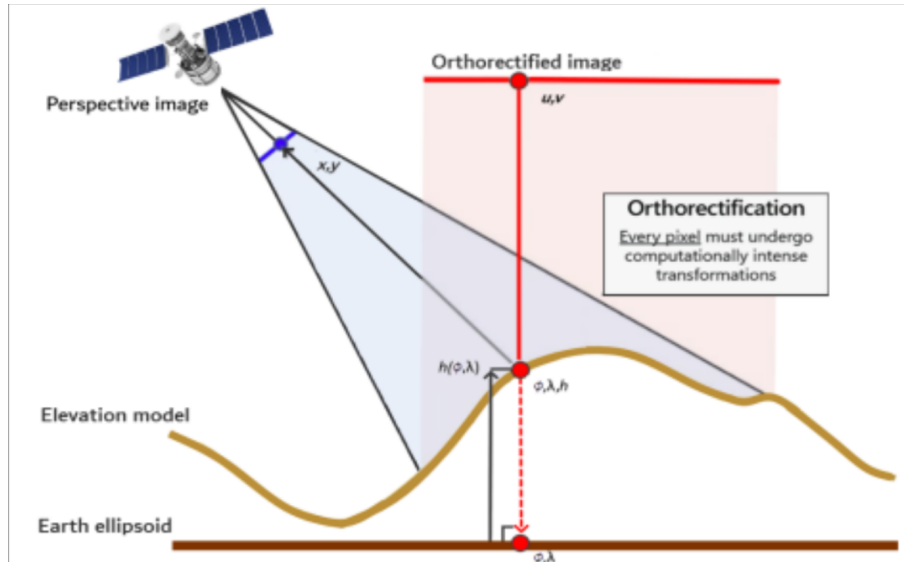


Figura 1.3: Orto-rectificación ilustrada

Los errores en una imagen satelital son múltiples y debido a distintas fuentes. Los errores asociados a la plataforma de adquisición, es decir, el satélite, son debido a su movimiento orbital y a la variación de orientación. Los errores asociados al sensor incluyen variaciones en la mecánica de adquisición de la imagen, variaciones del lente, ángulo de visión y efecto panorámico. Por otro lado, entre los errores asociados al objeto observado se cuentan la curvatura y rotación de la tierra, relieve del terreno y refracción atmosférica.

Para corregir estos errores se han desarrollado modelos matemáticos que corrigen cada una de las fuentes de error. Los modelos desarrollados se dividen en modelos empíricos, como polinomios y funciones racionales, o en modelos deterministas rigurosos, que modelan con exactitud el proceso de adquisición de la imagen. Por otro lado, para lograr la obtención de una orto-imagen con el menor error planimétrico posible se requiere de meta-datos asociados tanto a la plataforma como al objeto observado. Por ejemplo, para la corrección geométrica debido al movimiento del satélite se necesita su velocidad y variación en los ángulos de orientación durante el instante de captura de la imagen, así como su posición en el cielo y hora exacta. Para corregir los errores del relieve de terreno se necesitan mapas de elevación digital (DEM, del inglés *Digital Elevation Map*), mientras que para corregir los errores de dispersión atmosférica se requiere saber de las condiciones atmosféricas al momento de la captura, como la concentración de aerosoles y nubosidad. Por último, los puntos de control terrestre (GCP, del inglés *Ground Control Point*) son necesarios para poder refinar los parámetros de los modelos matemáticos de corrección.

En resumen, el proceso de orto-rectificación es complejo y en general requiere del uso de múltiples metadatos asociados a la imagen, siendo necesario, además, geolocalizar previamente la imagen para la obtención del DEM pertinente. Se han estudiado distintos modelos en la literatura ya que típicamente el proceso de orto-rectificación dependerá del tipo de sensor utilizado y de la cinemática del satélite. Por último, se hace referencia a un par de trabajos que sirven como guía para la comprensión del proceso en su totalidad, desde la descripción de las fuentes de errores, los modelos existentes y la comparación del desempeño de estos, en

[7] y [8], siendo el segundo una revisión del estado del arte del primer artículo, publicado por el mismo autor.

1.4.3. Geo-referenciación de imágenes satelitales

El proceso de geo-referenciación o geo-localización de una imagen de teledetección, consiste en asociar a cada punto de la imagen una posición geográfica determinada en la superficie terrestre, dado algún sistema geodésico de referencia. La geo-localización de una imagen es de gran importancia para la determinación del DEM que permitirá completar el proceso de orto-rectificación, aunque no debe ser confundido con la corrección geométrica de la imagen. A grandes rasgos, mientras mejor sea el desempeño de los algoritmos de corrección y localización, así como la resolución del DEM y la imagen, la orto-imagen producida tendrá mejor calidad y menor error de localización.

Los algoritmos de geo-referenciación dependen, en general, del tipo de imágenes que se dispongan, ya sean tipo *raster* o *vector*, pudiendo ser agrupados en dos categorías, siguiendo el trabajo propuesto en [9]. En el trabajo, los autores presentan una serie de técnicas de geo-referenciación y diversas aplicaciones de interés para el lector.

En este punto es importante destacar que si bien los procesos de orto-rectificación y geo-referenciación son vitales para la corrección de errores y artefactos en imágenes de teledetección, la complejidad de su implementación los deja fuera del alcance de este trabajo. Su desarrollo e implementación será llevado a cabo por el equipo de investigación del SPEL a lo largo de las fases de desarrollo y de operación del SUCHAI 2.

1.4.4. Caracterización de un sensor CMOS

La cámara a utilizar corresponde a un sensor sCMOS [10] (*scientific Complementary Metal-Oxide Semiconductor*), el cual consiste en un arreglo bidimensional de píxeles donde los competentes principales de cada uno son: un sustrato en cual se generan electrones producto de su interacción con los fotones incidentes y que se acumulan en un pozo de potencial, y un circuito eléctrico que amplifica la señal captada (voltaje). Una vez finalizado el tiempo de exposición a la fuente de luz, la señal de cada píxel se envía por columnas y se convierte en una señal digital gracias al conversor análogo-digital presentes en cada una. Luego, cada lectura se envía a un buffer para almacenar la imagen captada [11] [12].

Con el fin de comprobar el correcto funcionamiento del sensor y validar las especificaciones técnicas, se llevó a cabo una serie de experimentos que permiten calcular las métricas y características relevante que definen la calidad de un sensor tipo CMOS. El conocimiento en detalle sobre el funcionamiento del sensor y sus características son relevantes para comprender el proceso de captura de imágenes y su posterior corrección y estudio. A continuación se explica brevemente aquellas consideradas en este trabajo y la manera en que procedió para estudiar cada una de ellas.

1.4.4.1. Ganancia

La ganancia de un sensor corresponde al gradiente de la relación lineal que existe entre los fotoelectrones liberados y los números digitales (DN, *Digital Number*) producidos, y se

expresa en electrones por número digital ($[e^-/DN]$) [11]. Por ejemplo, si una cámara tiene una ganancia de 5 $[e^-/DN]$ entonces por cada 5 fotoelectrones captados el número digital entregado incrementa en 1 [13].

1.4.4.2. Capacidad Full Well

La capacidad Full Well (*pozo lleno*) de un sensor se define como la cantidad de carga que puede ser almacenada por un solo píxel sin que se llegue a saturar [11]. Se calcula como el producto entre la señal a la cual se tiene la mayor varianza y la ganancia [14].

1.4.4.3. No linealidad

La relación lineal entre los fotoelectrones y los DN antes mencionada no es perfecta en realidad sino aproximada. La no linealidad es, entonces, una medida de la desviación entre la relación real de fotoelectrones versus DN y un ajuste lineal [11]. Un método para calcular la no linealidad es graficar la señal promedio recibida versus el tiempo de exposición, luego ajustar una curva con regresión lineal y calcular las desviaciones para cada punto. Se calculan los valores porcentuales de desviación (no-linealidad) y se remueven los *outliers* usando el rango intercuartil IQR , descartando aquellos valores fuera del rango $(Q_1 - 3 \cdot IQR, Q_3 + 3 \cdot IQR)$, donde Q_1 corresponde al primer cuartil, Q_3 al tercer cuartil y $IQR = Q_3 - Q_1$, para luego calcular un promedio de los valores seleccionados.

1.4.4.4. Eficiencia cuántica

La eficiencia cuántica QE_{abs} (absoluta) es el porcentaje de fotones incidentes que el sensor puede convertir en electrones [15]. Para calcular la QE_{abs} se debe poder estimar la cantidad de fotones incidentes en el sensor, no obstante, es posible calcular una eficiencia cuántica relativa QE_{rel} sin necesidad de dichos datos. La QE_{rel} solo permite saber como se comporta la respuesta de distintas longitudes de onda de manera relativa frente a aquella con la mayor respuesta, por lo que indica un porcentaje de conversión de fotones en electrones, solo permite apreciar la forma de la curva de la QE_{abs} .

Dado lo anterior, el cálculo de la QE_{rel} es más directo que la absoluta, pero aún se requiere medir la potencia del flujo de fotones sin importar el tamaño del foto-diodo a utilizar. Además, al igual que para la QE_{abs} , se requiere de un monocromador, un instrumento capaz de emitir fotones a determinadas longitudes de onda. Así, para una determinada longitud de onda λ , se calcula la eficiencia $qe_r(\lambda)$ de acuerdo a la ecuación 1.1:

$$qe_r(\lambda) = \frac{mean_{DN}}{P \cdot \lambda \cdot e_t} \quad (1.1)$$

, donde $mean_{DN}$ es el valor promedio de píxel, P es la potencia medida por el foto-diodo y e_t el tiempo de exposición de la imagen. Luego, al contar con una serie de $qe(\lambda)$ para distintas longitudes de onda se normaliza dividiendo todos los valores por el máximo entre ellos, obteniendo así la QE_{rel} en función de λ .

Por otra parte, la QE_{abs} se calcula en función de la longitud de onda de acuerdo a la ecuación 4.10:

$$qe_a(\lambda) = \frac{\Phi_e}{\Phi_S} \quad (1.2)$$

, donde Φ_e es el flujo de electrones inducidos por píxel medido en $[e^-/m^2]$ y Φ_S es el flujo de fotones que inciden en un píxel medido en $[fotones/m^2]$. El flujo Φ_e se calcula según la ecuación 1.3:

$$\Phi_e = \frac{mean_{DN}}{A_{pix} \cdot gain} \quad (1.3)$$

, donde $mean_{DN}$ es el valor promedio de píxel, A_{pix} es el área del píxel y $gain$ es la ganancia del sensor. El flujo Φ_S se calcula según la ecuación 1.4:

$$\Phi_S(\lambda) = \frac{\Phi_P}{pw_{ratio}} = \frac{P \cdot \lambda \cdot e_t}{pw_{ratio} \cdot h \cdot c \cdot A_{ph}} \quad (1.4)$$

, donde Φ_P es el flujo de fotones sobre el foto-diodo medido en número de fotones por metro cuadrado, P es la potencia medida por el foto-diodo, e_t el tiempo de exposición de la imagen, h la constante de Planck, c la velocidad de la luz en el vacío y A_{ph} el área del foto-diodo en metros cuadrados. Se hace notar que el flujo Φ_S se estima a partir del flujo Φ_P dividiéndolo por el factor pw_{ratio} , que corresponde a la proporción entre P y la potencia medida por el foto-diodo ubicado en el lugar del sensor.

1.4.4.5. Corriente oscura

La corriente oscura (*dark current*) es el resultado de electrones generados aleatoriamente en el sensor debido a su propia temperatura, lo que ocurre en todo momento, incluso cuando el sensor no está expuesto a la luz [11]. Entre mayor sea el tiempo de exposición mayor será la corriente oscura acumulada, por lo cual se suele medir en electrones por píxel por segundo $[e^-/pixel/s]$. A su vez, la carga generada por esta corriente se denomina ruido oscuro (*dark noise*) y se calcula como la raíz cuadrada del producto entre la *dark current* y el tiempo de exposición [16].

1.4.4.6. Ruido de disparo

El ruido de disparo (*Shot Noise*) se produce cuando el número de fotones incidentes es suficientemente pequeño para dar lugar a la aparición de fluctuaciones estadísticas apreciables en una medición [11]. Este ruido no puede ser controlado y sigue una distribución de Poisson. Se calcula como la raíz cuadrada de la señal de electrones generada.

1.4.4.7. Función de dispersión de punto

Esta función (PSF, *Point Spread Function*) describe la respuesta de una imagen a una fuente de luz puntual y, de manera más general, la PSF es la respuesta del sistema al impulso de un sistema óptico enfocado. Es de gran relevancia tener información sobre la PSF de un sensor ya que el grado de dispersión de un objeto puntual es una forma de medir su calidad como sensor. En este trabajo, para analizar la PSF se ajusta un perfil gaussiano sobre la respuesta de los píxeles en una región alrededor de la incidencia de una fuente de luz puntual. [17]

1.4.4.8. Distorsión

Un sistema de captura de imágenes sensor+lente introduce distorsiones geométricas en la imagen resultante principalmente debido a la acción del lente para enfocar los fotones en el sensor [18]. Estos efectos pueden ser estimados y corregidos mediante la determinación de parámetros intrínsecos según el modelo de cámara estenopeica (*Pinhole Camera Model*) gracias a algoritmos en librerías como OpenCV [19] [20].

Capítulo 2

El nanosatélite SUCHAI 2

En este capítulo se introducirá a grandes rasgos el nanosatélite SUCHAI 2, su arquitectura y *payloads* a bordo además del Suchai Flight Software (SFS), el software de vuelo desarrollado por el SPEL tanto para el funcionamiento del comando central satélite, los *payloads* a bordo y la estación terrena, así como del sistema satelital completo, permitiendo la comunicación entre todos los nodos.

Luego, se introduce al módulo a bordo encargado de tomar las imágenes satelitales, el sensor CMOS pco.panda 4.2 bi UV del fabricante alemán de cámaras y equipos científicos PCO. Se detallarán algunas características técnicas relevantes así como también el software de control de ésta, disponible en el sitio web del fabricante ([PCO Linux API](#)).

2.1. Arquitectura de Hardware

El nanosatélite SUCHAI 2 sigue el estándar de CubeSat 3U [3]. Está compuesto de 3 unidades básicas acopladas verticalmente, siendo cada una un cubo de tamaño 10x10x10 cm y con una masa de alrededor 1.33 kg. En la figura 2.1 se muestra un diseño estructural del SUCHAI 2 donde se aprecian los componentes del satélite al interior de los cubos, siendo el cubo 1 (C1) el superior, el cubo 2 (C2) el del medio y el cubo 3 (C3) el inferior.

En el cubo C1 se encuentra el computador central del SUCHAI 2, el OBC (del inglés, *On Board Computer*), que corresponde a un procesador Atmel AVR32UC3 de 32 bits, encargado de controlar las funciones básicas del satélite y de la comunicación entre la estación terrena y los otros *payloads* a bordo. También se encuentra el modulo transmisor-receptor TRX que permite la comunicación con la estación terrena, las baterías y el sistema de suministro de energía (EPS, del inglés *Energy Power System*) y la placa del *payload* Suchai Star Tracker (STT), experimento cuyo objetivo es demostrar la capacidad de determinación de orientación del satélite por medio del análisis de la posición de las estrellas. El computador a cargo del STT embebido en esta placa es una Raspberry Pi Zero W, el cual también está a cargo de controlar la cámara sobre la cual se desarrolla este Trabajo de Título y que abordará en detalle más adelante.

En el cubo C2 se encuentran diversos *payloads* tales como: ruedas de reacción (RW del inglés *Reaction Wheels*), que permiten orientar el satélite al cambiar los momentos de inercia de determinados ejes; transmisores-receptores de banda S; magnetómetros para medir

la intensidad del campo magnético; GPS para experimentos de posicionamiento con esta tecnología.

En el cubo C3 se encuentran los magnetómetros que son bobinas que estiman la intensidad del campo magnético a partir de la corriente inducida y, en la parte inferior del cubo, se encuentra la cámara PCO (cubo azul) apuntando en el sentido negativo del eje Z. Como se mencionó, la cámara será comandada por la Raspberry del STT ubicada en el C1 y se requiere de un cable USB para la conexión de energía y datos.

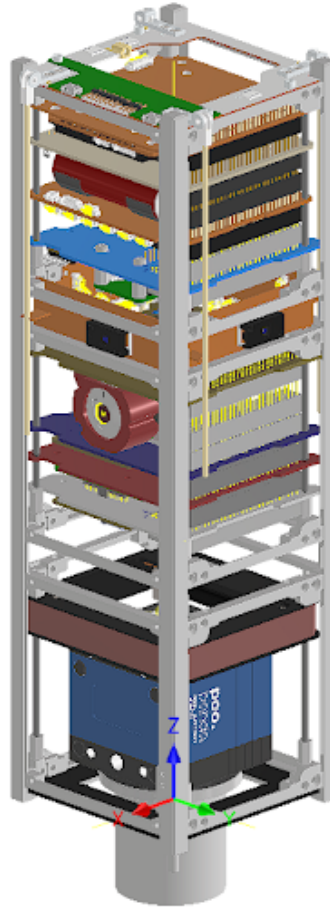


Figura 2.1: Esquema del SUCHAI 2

2.2. Suchai Flight Software (SFS)

El SFS es el software principal del programa espacial SUCHAI [21] [22]. Desarrollado originalmente para el SUCHAI 1, está presente en todos los procesadores de todos los satélites actualmente en desarrollo por el SPEL, el PlantSat, el SUCHAI 2 y SUCHAI 3, así también como en la estación terrena, facilitando la comunicación entre todos los nodos. El principio de diseño sobre el cual se desarrolla del SFS se basa en el cumplimiento requerimientos funcionales, como la ejecución de comandos tanto remotos como autónomos y el almacenamiento y envío a la estación terrena de datos, y requisitos no funcionales, como la extensibilidad, portabilidad, escalabilidad y modularidad del software. De acuerdo con estos requisitos, el

patrón de diseño más apropiado, y sobre el cual se basa el SFS, es el patrón de comandos [23] (o *command pattern design*) implementado sobre una arquitectura de capas. En la figura 2.2 los autores presentan un diagrama UML [24] que ilustra la arquitectura general del SFS sobre este modelo de capas.

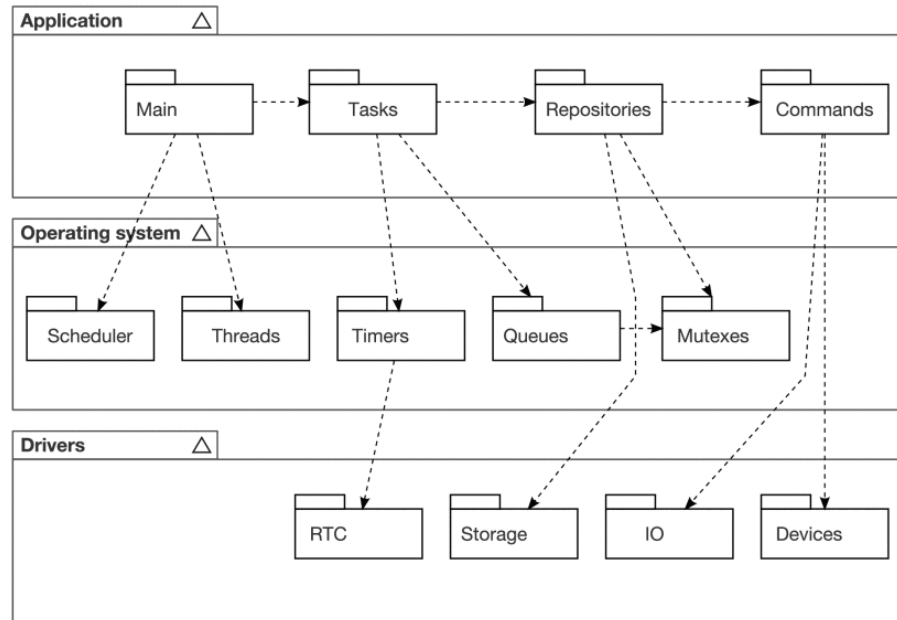


Figura 2.2: Diagrama del modelo UML de la arquitectura del SFS [22]

En la capa inferior se encuentran los drivers, y se cuenta con soporte para 4 arquitecturas: procesadores Atmel, ESP32, X86 y ARM. En la capa intermedia se encuentra el sistema operativo (SO), teniendo soporte para 2 SO, GNU/Linux para computadores como la Raspberry Pi y FreeRTOS para sistemas embebidos como el OBC, mientras que en la capa superior de aplicación donde se encuentran las tareas, los comandos y los repositorios, capa en donde se desarrolla el software de cada *payload* del SUCHAI 2, y en particular el de la cámara pco. En la figura 2.3 los autores presentan un diagrama UML que ilustra el patrón de comandos implementado sobre la capa de aplicación. Bajo esta lógica, cada cliente controla un subsistema específico del satélite solicitando la ejecución de comandos al *Invoker*, quien maneja las solicitudes de todos los clientes y se las entrega al *Receiver* quien se encarga de la ejecución, haciendo referencia a los repositorios de comandos (*command repository*) y de (*data repository*). De acuerdo al requisito funcional de ejecución de comandos, las solicitudes pueden ser autónomas mediante la implementación de tareas (*tasks*) del SFS, que corresponden a procesos de ejecución, *pthreads* en GNU/Linux y *tasks* en FreeRTOS, que invocan comandos del repositorio de acuerdo a los requerimientos específicos del subsistema que controla, o bien pueden ser solicitudes remotas mediante telemetría desde la estación terrena y programadas en el plan de vuelo (*flight plan*).

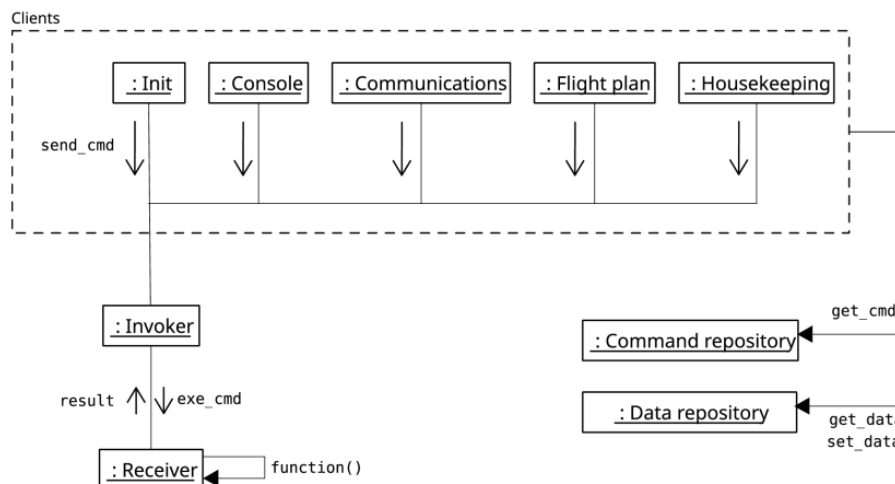


Figura 2.3: Diagrama del modelo UML del patrón de comandos del SFS [22]

De esta manera se simplifica la instalación e implementación del SFS y el desarrollo de aplicaciones dentro de éste, siendo totalmente extensible para que los *payloads* desarrollen su código sin entrar en conflicto o incompatibilidades con el núcleo del software. Gracias a la arquitectura del SFS la integración de la cámara PCO al satélite a nivel de software consiste en el desarrollo de tareas y comandos necesarios para la operación de ésta y la implementación del repositorio de datos. El computador a cargo de la cámara también debe operar el STT, por lo que se requiere de una aplicación conjunta que contenga los comandos de ambos *payloads* en un único (*command repository*) y se implementen las estructuras de datos de ambos en un mismo (*data repository*).

2.3. Raspberry Pi Zero W

La Raspberry Pi Zero W (RPI0W) es un poderoso computador para su reducido tamaño. Desarrollado por la empresa británica Raspberry Pi, la RPI0W ha encontrado numerosas aplicaciones en sistemas IoT (*Internet of Things*) gracias a su potencia, reducido tamaño y conectividad, al poseer WiFi y Bluetooth integrado. Por estas mismas razones el SPEL la ha designado como el computador a cargo de distintos *payload* a bordo de sus satélites, el SUCHAI 2 contando con 3 de éstas, además del procesador NanoMind de OBC. Pese a que la conectividad inalámbrica es innecesaria en el espacio (contraproducente de hecho, debido al consumo energético), es de inmensa utilidad para interacción con esta durante las fases de desarrollo en el laboratorio, ya que al contar con Linux se puede establecer una conexión por medio de *ssh* facilitando enormemente su programación y depuración en comparación a otros procesadores embebidos que necesitan de un módulo de hardware aparte para interactuar con ellos. En la figura 2.4 se muestra una imagen de la RPI0W, cuyas dimensiones son de 65x30 milímetros. Es importante destacar la presencia de la conexión de entrada micro USB tipo B a través de la cual se conectará la cámara PCO.

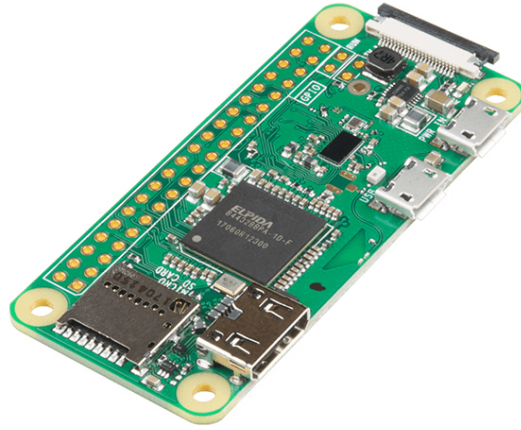


Figura 2.4: Raspberry Pi Zero W

En esta sección cabe mencionar los otros dispositivos considerados para las pruebas. Uno es una laptop convencional de gama media, el que fue dispositivo principal de desarrollo del software, y el otro es una Raspberry Pi 3B+, un modelo superior a la RPI0W, utilizado principalmente para comparar las capacidades de estos modelos. En la tabla 2.1 se resumen las características relevantes a considerar para las pruebas de software.

Tabla 2.1: Especificaciones generales de los dispositivos considerados

Dispositivo	Sistema operativo	Núcleos	Frecuencia	Procesador	Arquitectura	RAM
laptop	Ubuntu 20.04	4	2.2GHz	AMD A8-7410	64 bits	8Gb
RPI3B+	Raspbian Buster	4	1.4GHz	ARM Cortex A53	64 bits	1Gb
RPI0W	Raspbian Buster Lite	1	1GHz	ARM1176	32 bits	512MHz

2.4. Cámara pco.panda 4.2 bi UV

Las principales razones por las que fue elegido este modelo en particular de cámara son dos: primero, la respuesta espectral de la cámara permite captar parte de la banda UV, en específico, la zona de interés es alrededor de los 460 nanómetros, como se explicó en el capítulo anterior; segundo, las dimensiones de la cámara son ideales para ser montada dentro de uno de los cubos del satélite, siendo aproximadamente un cubo con lados de 7cm sin el lente. La cámara se muestra en la figura 2.5.



Figura 2.5: Cámara PCO modelo panda 4.2 bi UV

Otras especificaciones técnicas relevantes de la cámara se listan en la tabla 2.2. Es importante notar el alto consumo energético, lo que será un limitante para la operación continua de la cámara, sin embargo, dado que las zonas de interés para el estudio de la contaminación lumínica están acotadas desde el norte de Chile hasta la zona central se espera que el periodo de captura de imágenes sea apenas unos 5 minutos, durante el cual el satélite se desplaza unos 2100 kilómetros.

Tabla 2.2: Especificaciones técnicas de la cámara pco.panda 4.2 bi UV

tipo de sensor	CMOS monocromático
resolución (h x v)	2048 × 2048 píxeles
tamaño de píxel (h x v)	6.5 μm × 6.5 μm
formato de sensor	13.3 mm × 13.3 mm
rango espectral	190 nm .. 1100nm
corriente oscura (typ.)	42 e^- /pixel/s a 21 celsius
frame rate máximo a máxima resolución	40 fps
tiempo de exposición	10 μs .. 500 ms
rango dinámico A/D	16 bit
factor de conversión A/D	0.7 e^- /DN
consumo energético	typ. 4.5W (max. 6.0W)

La manera de operar la cámara es mediante una interfaz USB 3.1 a través del puerto de entrada tipo USB C. Dado que el puerto de entrada micro USB de la RPI0W es USB 2.0 la velocidad de transferencia de datos será de aproximadamente 10 veces menor que la usual de acuerdo a las especificaciones del estándar USB [25]. De todas maneras se ha comprobado el correcto funcionamiento y operación de la cámara mediante esta conexión.

2.5. Integración de la cámara al satélite

2.5.1. Integración de Hardware

Para operar la cámara solo se requiere de una conexión USB a un computador, sin embargo, se consume una cantidad de energía considerable para el satélite sólo para mantenerla encendida. De acuerdo a la tabla 2.2 la cámara consume 4.5W al encenderse, lo que se traduce en una corriente de 900mA dados los 5V de la conexión USB, lo cual se ha comprobado experimentalmente en el laboratorio. Por lo tanto, no es factible energizar la RPI0W por largos períodos de tiempo, sumando además a que el consumo de energía se traduce en un aumento de la temperatura de la cámara, calor que se propaga al resto del satélite por conducción y que es difícil de disipar en el espacio ya que al haber vacío solo se puede disipar fuera del satélite por radiación.

Una solución sobre la cual se trabajó para remediar el problema del consumo energético fue la conexión de un switch USB entre la RPI0W y la cámara para controlar el flujo de energía hacia ésta, permitiendo encendido y apagado mediante comandos por software y, de esta manera, se podría utilizar el computador sin que la cámara gaste innecesariamente las baterías del satélite. Este requerimiento también va en beneficio del otro *payload* que debía soportar la RPI0W, el STT.

En la figura 2.6 se muestra un esquema de la placa del *payload* STT, cuyo principal componente es la RPI0W. Otros componentes del STT son: el giroscopio (Gyro) para obtener estimaciones de los vectores de posición y velocidad y la cámara STT que corresponde a una cámara especial diseñada para los computadores Raspberry Pi y que se conecta a un puerto serial especial (CSI, Camera Serial Interface). Por otro lado, el microcontrolador PIC16F1847 presente en la placa cumple la función de establecer una comunicación en serie con el OBC y otros *payloads* mediante los protocolos I2C (Inter-Integrated Circuit) y CSP (CubeSat Protocol). De esta manera se puede comandar directamente la RPI0W desde la estación terrena pasando a través del nodo central, el OBC, quien recibe los comandos por telemetría y los redirige al *payload* correspondiente según su nodo, siguiendo el protocolo CSP.

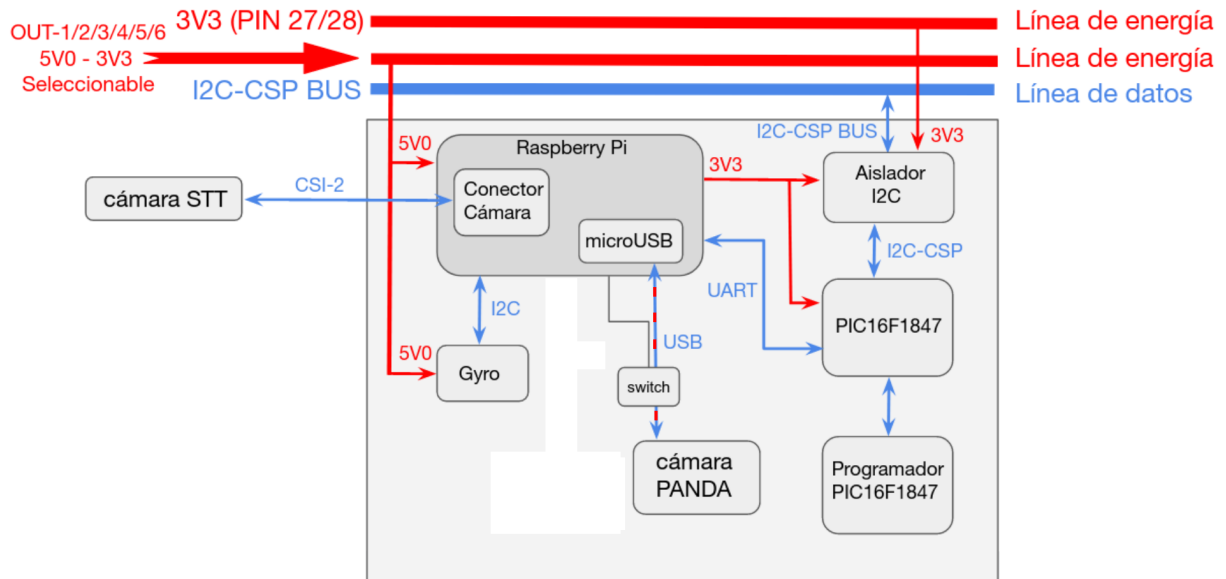


Figura 2.6: Placa del Star Tracker

El switch USB adquirido del fabricante 8086 [26] fue elegido por su simpleza y reducido tamaño. En la figura 2.7 se aprecian los 3 pines de conexión, uno de los cuales se utiliza para permitir el paso de energía de un extremo USB al otro mientras que los otros 2 son salidas de voltaje y tierra. Para comandar el switch se utiliza uno de los pines del GPIO (General Purpose Input Output) de la RPI0W asignándolo como pin de salida y cambiando su lógica digital a alta para habilitar el switch y baja para deshabilitarlo mediante un simple programa en lenguaje python.

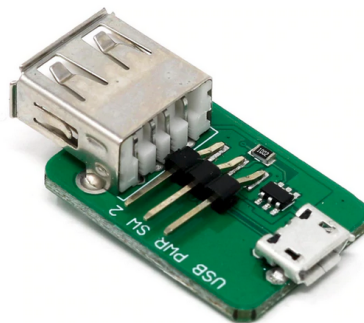


Figura 2.7: Switch USB

El inconveniente que produce agregar este switch es la necesidad de manufacturar un cable USB personalizado a partir de otros cables para establecer la conexión desde la cámara (tipo USB C) al switch (extremo tipo USB A), con el largo justo y removiendo conector tipo A debido a su gran tamaño, y otro cable desde el switch (extremo tipo micro USB B) a la RPI0W (tipo micro USB B), también del largo indicado. Después de reiterados esfuerzos cortando y soldando cables no se pudo establecer una conexión estable entre la RPI0W y la cámara usando esta configuración, por lo que finalmente se optó por descartar el switch y establecer una conexión directa con la cámara, sacrificando la posibilidad de apagar la

cámara cuando la RPI0W está energizada y por ende restando facultades a ambos *payloads* para hacer uso de la RPI0W.

En la figura 2.8 se muestra la cámara montada al satélite sujeta por una estructura diseñada especialmente para este propósito y fabricada con Windform ([sitio web](#)), un tipo de plástico especial para aplicaciones espaciales. Se aprecia además el lente ya montado a la cámara, dentro de la estructura de Windform y por fuera de la estructura del satélite.

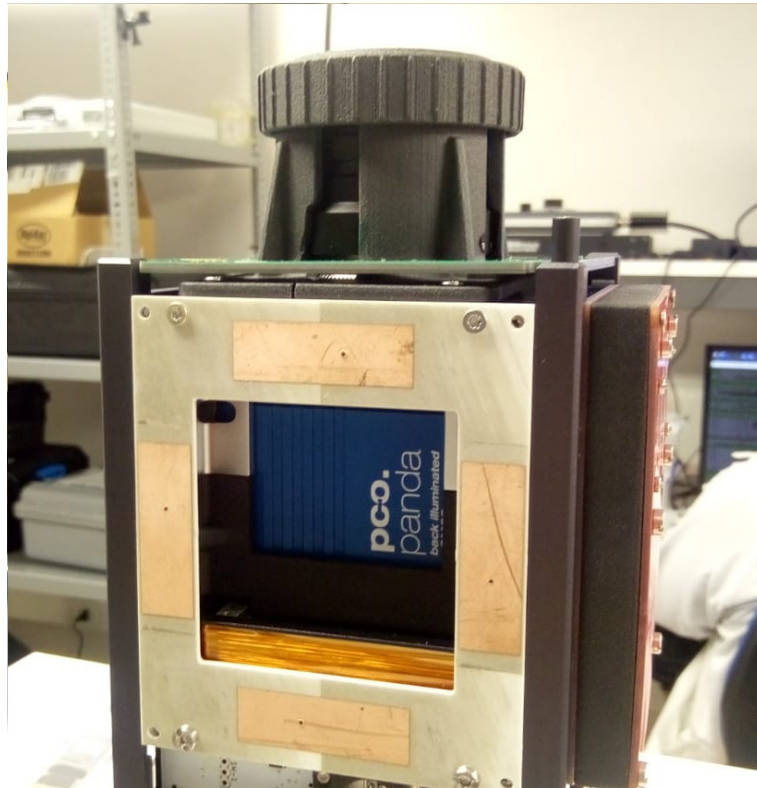


Figura 2.8: Cámara dentro del satélite

2.5.2. Integración de Software

El fabricante de la cámara pone a disposición dos herramientas de software para la operación de sus equipos, una basada en una interfaz gráfica desarrollada para el sistema operativo Windows y el otro es un kit de desarrollo de software (SDK) para la implementación de aplicaciones personalizadas, tanto para Windows como para Linux [27].

El software de control de la cámara se desarrolló en base al SDK para Linux, modificando el comportamiento interactivo original del software a uno basado en parámetros para facilitar su invocación a través del software de vuelo. El software se ejecuta invocando la aplicación compilada pasando 3 argumentos necesarios; el primero indica la cantidad de fotos a tomar, el segundo indica la base del tiempo de exposición, en microsegundos con un 0 o bien en milisegundo con un 1, y el tercero indica el valor del tiempo de exposición. Además de estos argumentos se cuenta con dos parámetros de interés, uno para obtener una medición de la temperatura interna de la cámara sin capturar fotos y otro para cambiar el modo de captura

agrupando píxeles, con la opción de establecer agrupamientos de píxeles en zonas de 2x2 y en zonas de 4x4, reduciendo así el tamaño de la imagen a un cuarto y a un dieciseisavo, respectivamente, lo que se traduce en un aumento de la velocidad de captura.

La ejecución del software de la cámara se realiza a través de un comando del software de vuelo, mientras que otros comandos se encargan de manejar las imágenes y los metadatos. La operación de la cámara, y del *payload* en general, se detallará en el capítulo siguiente.

Capítulo 3

Operación de la cámara

Para operar la cámara se debe ejecutar el programa desde un comando del SFS. Para este *payload* se han desarrollado una serie de comandos que permiten capturar imágenes, administrar la base de datos (imágenes y metadata), ejecutar funciones de pre procesamiento como compresión, escalamiento, cortes, entre otras, y algoritmos de extracción de información. Estos comandos permitirán evaluar la calidad y relevancia de las imágenes para determinar con seguridad cuales deben ser descartadas y cuales serán objeto de estudio.

Por otro lado, el SFS incluye comandos para enviar datos desde y hacia a la estación terrena con la posibilidad de recuperar paquetes que se hayan perdido en el canal y para la reconstrucción del archivo original a partir de estos paquetes. Estos comandos son vitales para recuperar información desde cualquier *payload* en el satélite y en particular para bajar las imágenes captadas por la cámara

En la tabla 3.1 se presenta un resumen de los comandos específicos para la cámara PCO. En el resto de este capítulo se explicarán la función que desempeña cada uno y los parámetros que reciben. Se muestran igualmente los comandos desarrollados para encender y apagar la cámara mediante el el switch controlado por un pin del GPIO de la Raspberry, pero que están en desuso por las razones explicadas en la sección 2.5.1. En la tabla 3.2 se listan los directorios donde se maneja la base de imágenes y el contenido de cada uno.

3.1. Captura de imágenes y generación de meta datos

Para capturar imágenes se cuenta con el comando `cam_grab`, el cual recibe 4 parámetros, los 3 primeros son los mismos argumentos que se deben pasar al programa de la cámara, indicados en el capítulo anterior, mientras que el cuarto parámetro indica el agrupamiento de píxeles deseado, 1 para no usar agrupamiento, 2 para agrupar en regiones de 2x2 y 4 para regiones de 4x4. Además de capturar las fotos, el programa de la cámara se encarga de generar un archivo `.csv` de metadatos asociados a dichas imágenes, el cual contiene las columnas: id de la imagen, nombre de la imagen, timestamp de la imagen y temperatura de los 3 sensores internos de la cámara.

Las imágenes y metadatos son almacenados dentro de un directorio creado con el timestamp al momento de captura de la primera imagen, la cantidad de fotos y el tiempo de exposición utilizado, y este a su vez dentro del directorio de imágenes *raw*. En la tabla 3.3 se

Tabla 3.1: Lista de comandos de la cámara PCO

comando	nr de parámetros	lista de parámetros
<code>cam_turn_on</code>	0	None
<code>cam_turn_off</code>	0	None
<code>cam_get_temp</code>	0	None
<code>cam_ls_raw</code>	0	None
<code>cam_thumbnail</code>	4	<code>img_folder</code> , <code>id</code> , <code>div_factor</code> , <code>is_raw</code>
<code>cam_convert</code>	3	<code>dir_to_convert</code> , <code>format</code> , <code>rm_raw</code>
<code>cam_grab</code>	4	<code>n</code> , <code>timebase</code> , <code>exp_time</code> , <code>binning</code>
<code>cam_hist</code>	3	<code>dirname</code> , <code>rangemode</code> , <code>bins</code>
<code>cam_get_hist</code>	2	<code>dirname</code> , <code>imid</code>
<code>cam_threshold</code>	1	<code>filename</code>
<code>cam_crop_image</code>	5	<code>filename</code> , <code>xo</code> , <code>yo</code> , <code>xf</code> , <code>yf</code>
<code>cam_list_images</code>	0	None
<code>cam_check_storage</code>	0	None

Tabla 3.2: Directorio de datos de la cámara pco

directorio	contenido
<code>/home/pi/pco_cam_data/</code>	directorio base, abreviado como “.”
<code>./pco_raw_imgs</code>	imágenes <i>raw</i> (sin procesar)
<code>./pco_converted_imgs</code>	imágenes comprimidas
<code>./pco_thumbnail_imgs</code>	imágenes en miniatura
<code>./pco_crop_imgs</code>	imágenes recortadas
<code>./pco_threshold_imgs</code>	imágenes binarias

muestra un ejemplo del archivo de meta datos asociados a una secuencia de 5 imágenes. Se observa que el nombre de cada imagen de la secuencia (*filename*) lleva la marca de tiempo (*timestamp*) al momento de ejecutar el software de la cámara (por lo a veces puede darse que esta marca esté desfasada en 1 segundo con la indicada por la columna (*timestamp*), el *id* de la imagen en la secuencia y el tiempo de exposición utilizado. Por su parte, la columna *timestamp* representa el momento exacto en que se capta la imagen con una resolución de microsegundos. El resto de las columnas, *camtemp*, *mostemp* y *pstemp* representan las temperaturas en grados al momento de captura según los 3 sensores internos en la cámara, uno en la electrónica, otro en el sensor y el otro en la fuente de energía, respectivamente.

Por otro lado, se cuenta con el comando `cam_get_temp` para obtener un reporte de la temperatura de la cámara, el cual no recibe parámetros y ejecuta el programa de la cámara con la orden de capturar 0 imágenes y pasando un parámetro que le indica medir la temperatura.

Tabla 3.3: Archivo de metadatos asociados a una secuencia de imágenes

id	filename	timestamp	camtemp	cmostemp	pstemp
0	211108_174206_000_496us.tif	2021-01-08_17:42:06.106419	40	29.0	27
1	211108_174206_001_496us.tif	2021-01-08_17:42:06.131364	40	29.0	27
2	211108_174206_002_496us.tif	2021-01-08_17:42:06.156308	40	29.0	27
3	211108_174206_003_496us.tif	2021-01-08_17:42:06.181252	40	29.0	27
4	211108_174206_004_496us.tif	2021-01-08_17:42:06.231141	40	29.0	27

3.2. Preprocesamiento y extracción de información

Dado que las imágenes originales (*raw*) tiene una profundidad de 16 bits (rango dinámico A/D en tabla 2.2), a máxima resolución, es decir, 2048x2048 píxeles, cada una tiene un tamaño de $2 \times 2^{11} \times 2^{11} = 2^{23}$ bytes, lo que se traduce en 8 mebibytes (MiB) o, aproximadamente, 8.4 megabytes (MB)², por lo que la gestión de la base de imágenes *raw* es de suma relevancia para no saturar prematuramente la memoria de la RPI0W. Por otro lado, el enlace de comunicación entre la estación terrena y el satélite es restringido en tiempo, entre 10 a 15 minutos de enlace por pasada sobre la estación, y en velocidad de datos, típicamente de unos 10 kilobits por segundos, por lo que es imposible la descarga de una imagen *raw* durante una pasada, e incluso muy difícil considerando varias pasadas ya que hay que tener en cuenta que el enlace debe ser compartido con otros *payloads* que igualmente desean bajar datos y subir comandos. El procesamiento previo a bordo de las imágenes *raw* permitirá extraer información relevante para manejar los datos generados y seleccionar adecuadamente aquellos que se enviarán a la estación terrena.

Una de las funciones básicas de pre procesamiento con la que debe contar el payload es la compresión de imágenes. El comando `cam_convert` permite convertir las imágenes *raw* en formato .tif a otros formatos con compresión con pérdidas (.jpg y .webp) o bien sin pérdidas (.png). El comando recibe tres parámetros: el primero es el nombre de la carpeta que contiene las imágenes *raw* que se desean convertir, el segundo es el formato de conversión (alguno de los 3 soportados) y el tercero es una señal para eliminar las imágenes *raw* luego de convertirlas (0 para mantenerlas, 1 para eliminarlas). Las imágenes comprimidas almacenan en `./pco_converted_imgs`, en una carpeta con el mismo nombre original, copiando igualmente los metadatos.

Otro procesamiento de utilidad es la creación de miniaturas de imágenes, es decir, imágenes con resolución reducida con tal de que su tamaño sea considerablemente menor pero que aún se puedan apreciar los objetos relevantes presentes en la imagen. El comando `cam_thumbnail` toma una imagen *raw* o comprimida y crea una miniatura dado un factor de conversión y la

² [Detalles sobre prefijos binarios](#)

almacena en una carpeta con el mismo nombre que aquella que contiene a la original, y esta a su vez dentro del directorio `./pco_thumbnail_imgs`. El comando recibe cuatro parámetros: el primero es el nombre de la carpeta que contiene la imagen a escalar, el segundo es el identificador (*id*) de la imagen, el tercero es el factor de conversión (un número entero entre 1 y 100) y el cuarto indica si la imagen es *raw* (con un 1) o si es una comprimida (con un 0). De esta manera se obtiene una miniatura que puede pesar hasta mil veces menos que una imagen *raw* y manteniendo las características relevantes presentes, es decir, unos cuantos kilobytes, lo cual es factible de enviar durante una pasada del satélites.

El comando `cam_threshold` transforma imágenes *raw* en imágenes binarias, es decir, con dos valores posibles para cada píxel, 0 o 1 (negro o blanco), reduciendo su tamaño al bajar la profundidad de bits de 16 a 1 y sin reducir su resolución. El comando recibe un solo parámetro, el nombre de la carpeta con las imágenes *raw* a convertir y son almacenadas en `./pco_threshold_imgs`.

El comando `cam_crop_image` crea una imagen a partir de una sección de una imagen *raw*, lo que permitirá seleccionar zonas de interés. El comando recibe cinco parámetros: el primero es el nombre de la carpeta con las imágenes *raw* objetivo y los siguientes cuatro parámetros determinar la zona que se desea seleccionar, x_o , x_f , y_o e y_f . Las imágenes cortadas son almacenadas en `./pco_crop_imgs`.

Finalmente, se cuenta con un comando que calcula el histograma de una imagen, lo que permite extraer información preliminar sobre el contenido de una imagen, pudiendo determinar rápidamente con los valores del histograma si una imagen está en negro, en blanco, o bien si es una imagen a priori “válida” observando la distribución de valores de píxeles. El comando `cam_hist` recibe tres parámetros: el primero es el nombre de la carpeta con las imágenes *raw* objetivo, el segundo es el modo de calcular el rango del histograma, “minmax” para utilizar un rango según el mínimo y máximo valor de una imagen y “full” para utilizar todo el rango de valores con 16 bits, de 0 a 65535, y el tercer parámetro es el número de bins del histograma. Los histogramas se almacenan en un archivo `.csv` dentro de la misma carpeta que contiene las imágenes, donde la primera columna es el *id* de las imágenes y las siguiente son los bins. Para obtener el histograma de una imagen desde la estación terrena se cuenta con el comando `cam_get_hist`, el cual recibe 2 parámetros: el primero es el nombre de la carpeta que contiene la imagen objetivo y el segundo es el *id* de la imagen objetivo. Dado que se pueden calcular varios histogramas se retorna a la estación terrena todos los histogramas encontrados en la ruta.

3.3. Almacenamiento y gestión de la base de imágenes

Para la gestión de la base de imágenes se cuenta con comandos y con una tarea de “limpieza” (ver sección 2.2) que se ejecuta cada minuto en la RPI0W. El comando `cam_ls_raw` lista todas las carpetas dentro del directorio `./pco_raw_imgs` y entrega la cantidad total de imágenes *raw* y el total de bytes que ocupa dicho directorio en memoria. Por otro lado, la tarea de “limpieza” ejecuta el comando `cam_list_images` cada minuto, el cual calcula el espacio total disponible en la RPI0W, el espacio libre y el usado del total y también el espacio usado por el directorio base de la pco. Esta información es almacenada en un archivo `.json`. Además, la tarea ejecuta cada 10 minutos el comando `cam_check_storage` el cual, a partir

de la información en el archivo `.json` antes mencionado, elimina imágenes según antigüedad si es espacio usado en la RPI0W supera el 95 %, hasta llegar al 70 %.

3.4. Recepción de datos y envío de comandos

Para la comunicación entre el *payload* pco y la estación terrena se cuenta con comandos ya implementados en el SFS, entre los cuales se seleccionan los relevantes para este *payload* en la tabla 3.4.

Tabla 3.4: Lista de comandos seleccionados de telemetría

comando	nr de parámetros	lista de parámetros
<code>com_send_cmd</code>	2	node, command
<code>tm_send_file_parts</code>	5	file_name, file_id, start_frame, end_frame, node
<code>tm_merge_file</code>	2	file_name, file_id

Para el envío de comandos desde la estación terrena se cuenta con el comando `com_send_cmd`, el cual recibe el nodo de destino (designado como 4 para este *payload*) y el comando a ejecutar.

Para el envío de datos desde el *payload* a la estación terrena se tiene el comando `tm_send_file_parts` que envía un archivo por partes de 200 bytes cada una. El comando recibe 5 parámetros: el primero es la ruta completa al archivo que se desea enviar (`file_name`), el segundo es el identificador que se desea asignar a este archivo (`file_id`), que debe ser único entre los archivos enviados por este comando, el tercer y el cuarto parámetros es el rango de partes que se desea enviar y el quinto parámetro es el nodo de destino (la estación terrena esta designada como el nodo 10). Para un primer intento de envío de un archivo se debe usar el rango de partes $(0, n)$, donde n es un número arbitrariamente grande tal que $200n$ es mayor al tamaño del archivo. Luego, si en la estación terrena resulta que se perdieron partes en el canal se puede volver a utilizar este comando para enviar las partes faltantes indicando el rango deseado.

Para la reconstrucción del archivo original a partir de todas sus partes, en la estación terrena se ejecuta el comando `tm_merge_file`, el cual recibe el nombre que se dará al archivo reconstruido, idealmente el mismo que tenía antes de ser enviado, y el identificador utilizado para su envío.

Capítulo 4

Experimentos y Análisis

4.1. Configuración experimental

4.1.1. Monocromador

En la figura 4.1 se muestra la configuración experimental utilizada para capturar imágenes controlando la longitud de onda de la luz incidente al sensor, obteniendo además mediciones de la potencia radiada. Esta configuración consiste fundamentalmente en un monocromador (artefacto gris detrás de la esfera blanca en el recuadro rojo número 4), un instrumento que permite controlar el flujo de fotones de entrada proveniente de una fuente de luz blanca (lámpara en el recuadro rojo número 3) hacia la salida que da a una esfera integradora (esfera blanca antes mencionada), filtrando según la longitud de onda deseada. Dicha esfera integradora concentra el flujo de fotones hacia la salida (tubo negro hacia la izquierda), en donde se monta un fotodiodo en su parte superior (recuadro morado) y cuyas mediciones de potencia percibida se procesan y se muestran en el potenciómetro (recuadro rojo número 1).

El flujo de fotones viaja por el tubo con tal de llegar de manera uniforme y paralela al sensor que se ubica dentro de la cámara metálica (recuadro rojo número 5), el cual cuenta con una abertura a un costado para operar el sensor y un sistema de enfriamiento en su parte superior. Por otro lado, en el recuadro rojo número 2 se muestra el instrumento que controla la potencia radiada por la lámpara mediante el ajuste de la corriente de alimentación.

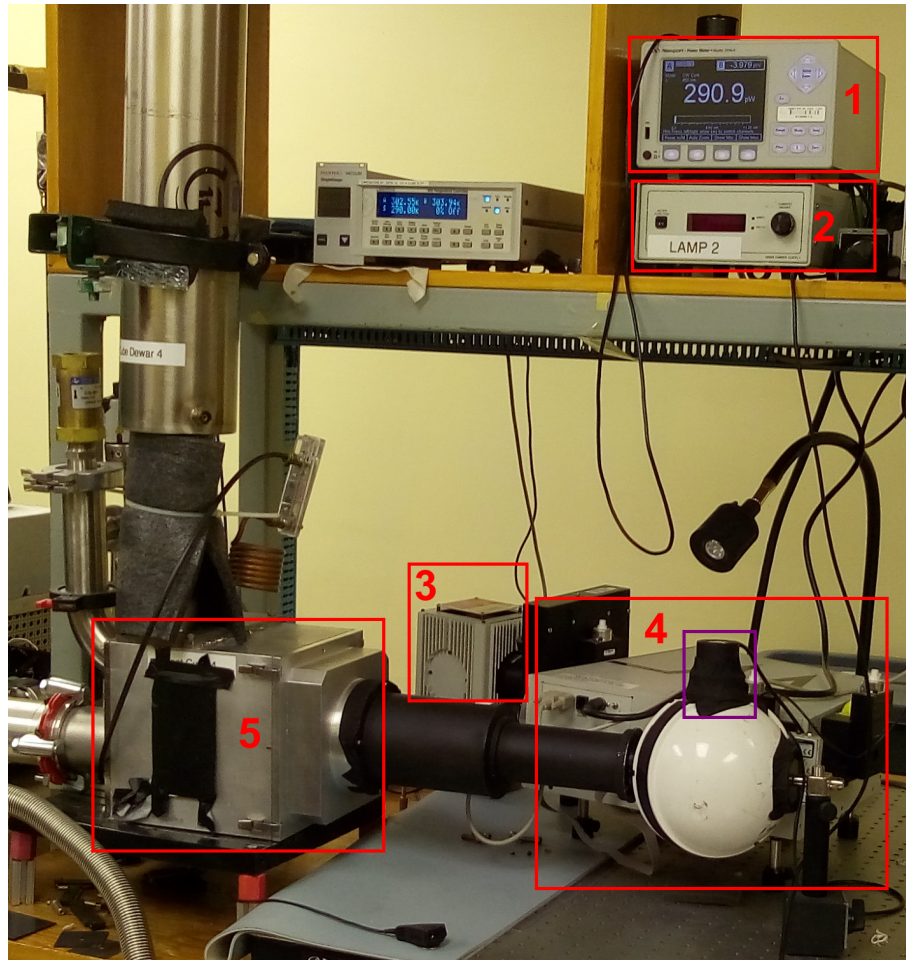


Figura 4.1: Configuración experimental 1 (*setup 1*)

Sin embargo, cabe destacar que la cámara metálica que aloja a los sensores a testear no fue diseñada para alojar una cámara completa, por lo que se debió adaptar la configuración para montar la cámara PCO. En la figura 4.2 se muestra como queda la cámara acoplada a la salida de la esfera integradora, fijada con cinta adhesiva negra para bloquear el máximo de interferencia proveniente de fuentes de luces externas, a lo que se suma una manta especial negra que bloquea la luz puesta tapando toda la configuración y el apagado de las luces del laboratorio.

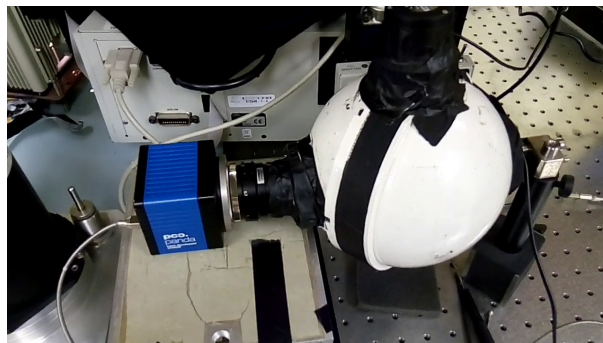


Figura 4.2: Cámara PCO adaptada al *setup 1*

4.1.2. Láser

En la figura 4.3 se muestra la configuración experimental usada para capturar imágenes en donde se concentra un haz de luz en un único píxel. Esta configuración consiste en un láser de Helio-Neón cuya longitud de onda es de 632.8 nanómetros y con una potencia de salida de 5 miliwatts. En el recuadro rojo número 1 de la figura 4.3 se muestra dicho láser y cómo su salida está conectada a una fibra óptica (cable amarillo) la cual conduce el haz de luz hacia un colimador que se aprecia en el recuadro rojo número 2. Este colimador cumple la función de paralelizar el haz de luz (luz colimada) de entrada para luego apuntar este haz de luz colimada a la cámara PCO que se encuentra fuera del plano de la imagen hacia la derecha.

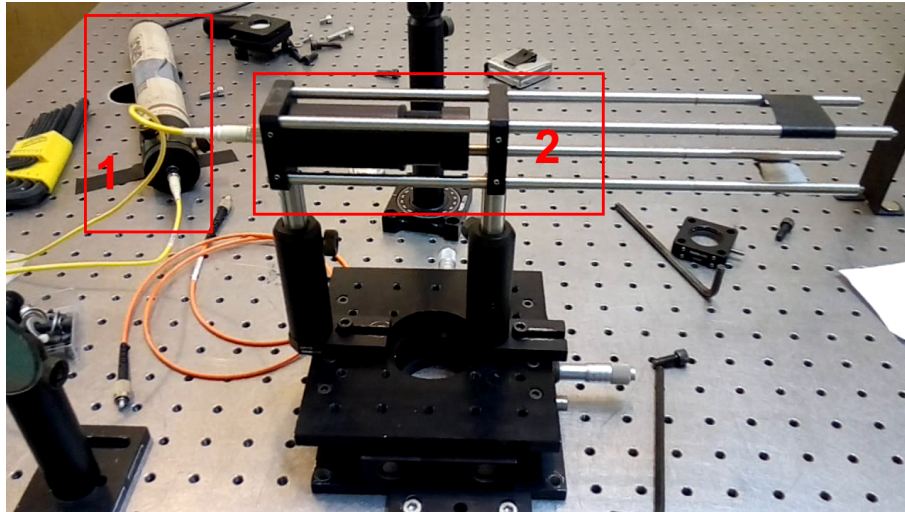


Figura 4.3: Configuración experimental 2 (*setup 2*)

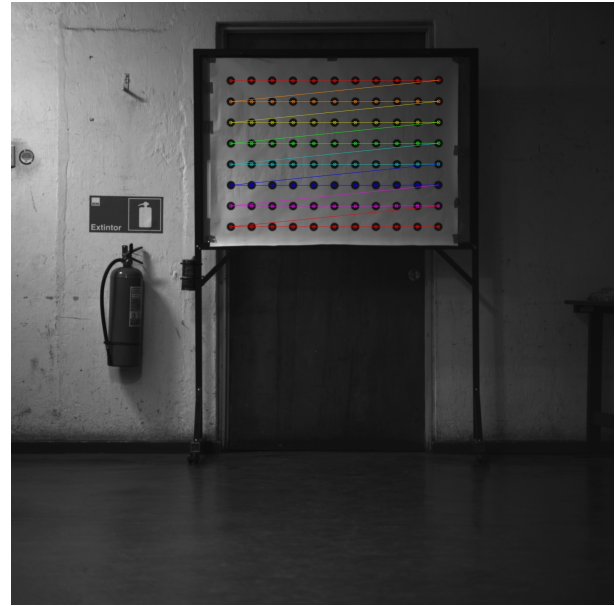
4.1.3. Calibración de la cámara

Para estimar la distorsión en las imágenes y obtener los parámetros de calibración de la cámara se utilizó el método de detección de un patrón regular basado en círculos negros equidistantes y de igual tamaño sobre un fondo blanco 4.4.a. Dado que el patrón no es lo suficientemente grande para cubrir toda la imagen desde la distancia de foco designada se debió capturar una serie de imágenes moviendo el patrón horizontalmente e inclinando la cámara verticalmente de modo que cada región de píxeles del sensor logre captar el patrón, y, además, porque el algoritmo que permite estimar los parámetros de calibración requiere de varias imágenes para aumentar su robustez. Con este conjunto de imágenes se utilizan los algoritmos proporcionados por la librería OpenCV y detallados en su guía de calibración de cámaras ³ para detectar el patrón de círculos en cada imagen, como se muestra en la figura 4.4.b. Luego, con la información de la posición de los círculos tanto en el espacio real como en el espacio de la imagen se procede a estimar los parámetros de calibración.

³ [Guía de calibración OpenCV \(en inglés\)](#)



(a) Imagen original



(b) Imagen con detección de puntos

Figura 4.4: Imagen de prueba para calibración

4.2. Ciclo de operación

Un parámetro importante para el análisis de imágenes es la velocidad de fotogramas o fotogramas por segundo (FPS, del inglés *Frames Per Second*) que puede lograr el sistema en general. La velocidad de fotogramas a la que la cámara puede capturar las imágenes y almacenarlas en la memoria depende principalmente de 4 factores. En primer lugar, la cámara en sí, como se indica en la hoja de datos, puede alcanzar 40 FPS a resolución máxima, lo que se puede lograr utilizando una interfaz USB 3.1.

Estos dos factores antes mencionados (interfaz y computadora) están restringidos al diseño general del sistema, es decir, el diseño del satélite. La computadora que operará la cámara es una Raspberry Pi Zero W (RPI0W), una computadora integrada capaz de ejecutar Linux pero con recursos limitados. Desafortunadamente, aunque la RPI0W es capaz de controlar la cámara, la interfaz de conexión es USB 2.0, que es aproximadamente 10 veces más lento que el USB 3.1 recomendado, y tendrá un gran impacto en el FPS real alcanzable.

Otro factor relevante es el rendimiento del software. Dado que la RPI0W es capaz de soportar una distribución de Linux, podemos utilizar el kit de desarrollo de software (API, del inglés *Application Programming Interface*) proporcionado por el fabricante para controlar la cámara con un programa en C++. El software tuvo que ser modificado para ajustarse a los requisitos del software de vuelo del satélite, basado en un patrón de diseño de comandos en vez del programa interactivo original.

El software se adaptó para satisfacer las necesidades de la misión espacial para el sistema de la cámara y para ser lo más eficiente posible. Los parámetros relevantes son el tiempo de exposición y el número de imágenes a adquirir. El software se instaló y se probó en los 3

dispositivos indicados en la tabla 2.1.

El experimento consiste en adquirir 25 imágenes consecutivas con cada dispositivo. La tabla 4.1 muestra los fotogramas promedio por segundo obtenidos calculados a partir de las marcas de tiempo de las imágenes. Queda claro que el más rápido es la laptop, seguido de la RPI3B+ alimentado por USB. Cuando se alimentó la RPI3B por los pines se obtuvo un valor de FPS menor debido a la potencia limitada que se puede entregar a través de los pines a la cámara. En estos 2 dispositivos no se obtuvo variación en los FPS al cambiar el tiempo de exposición, demostrando que el mayor inconveniente en el rendimiento se debe a la conexión de la interfaz USB 2.0.

Por último tenemos el RPI0W con una frecuencia de cuadro promedio de 2.6Hz. No observamos una diferencia importante al alimentar el RPI0W por USB o pines, probablemente debido a la baja potencia consumida por la computadora que puede ser suministrada suficientemente por estos 2 métodos. Se notan algunas diferencias en fps al cambiar el tiempo de exposición. No se puede detectar una tendencia clara, por lo que nuestra conclusión es que puede deberse al tiempo de las tareas del procesador cuando se ejecutan varios subprocesos en un solo núcleo.

Tabla 4.1: FPS promedio obtenido por diferentes dispositivos y tiempo de exposición t_e

t_e / dispositivo	laptop	RPI3B+ usb	RPI3B+ pin	RPI0W usb	RPI0W pin
10us	4.54	4.31	3.72	2.56	2.66
1ms	4.54	4.31	3.72	2.60	2.55
10ms	4.54	4.31	3.72	2.55	2.57

También es importante notar la variación (Δt) de tiempo entre imágenes consecutivas. La figura 4.5 muestra el tiempo en segundos entre la imagen i y $i + 1$, de arriba a abajo. A la izquierda observamos que el tiempo es casi constante (menos de un microsegundo de diferencia) para las imágenes adquiridas por el portátil. En el medio, observamos que en algunos momentos el rpi3b + tarda más en adquirir la siguiente imagen (en rojo) pero también hay un momento con un gran retraso antes de adquirir la siguiente imagen (en azul), para el resto de imágenes el Δt es casi constante.

[0.22449900000000156,	[0.2244990000000023,	[0.34922000000000253,
0.2244989999999998,	0.2244989999999978,	0.349221,
0.22449900000000156,	0.2244990000000023,	0.3741659999999534,
0.2244989999999998,	0.2244999999999992,	0.32427600000000467,
0.22449900000000156,	0.2494430000000003,	0.3991100000000003,
0.22449900000000156,	0.2244989999999978,	0.3242759999999757,
0.2244989999999998,	0.2244989999999978,	0.3741649999999786,
0.2244999999999903,	0.2244989999999978,	0.349221,
0.22449900000000156,	0.24944400000000044,	0.37416500000000497,
0.2244989999999998,	0.2244989999999978,	0.4240539999999815,
0.22449900000000156,	0.2244989999999978,	0.698442,
0.22449900000000156,	0.22449900000000067,	0.4240539999999815,
0.2244989999999998,	0.2494439999999955,	3.5670420000000007,
0.22449900000000156,	0.2244989999999978,	0.3741649999999997,
0.2244989999999998,	0.22449900000000067,	0.42405400000000004,
0.22450000000000026,	0.2494429999999942,	0.349221,
0.2244989999999998,	0.24944400000000044,	0.349221,
0.22449900000000156,	0.2244989999999978,	0.3741650000000001,
0.2244989999999998,	0.2244989999999978,	1.6213819999999997,
0.22449900000000156,	0.22449900000000067,	0.3991100000000003,
0.2244989999999998,	0.2494439999999955,	0.3741649999999964,
0.22449900000000156,	5.363034999999999,	0.4739430000000002,
0.2244999999999903,	0.22449900000000156,	0.349221,
0.22449900000000156]	0.2494429999999942]	0.3741649999999964]

Figura 4.5: Tiempo entre imágenes consecutivas. Laptop (izquierda), RPI3B+ (medio), RPI0W (derecha)

A la derecha de la figura 4.5, observamos Δt para las imágenes adquiridas por RPI0W. Este tiempo tiene mayor variación que en los otros 2 dispositivos, observándose también los largos retardos como se señaló anteriormente (en azul). Para calcular el promedio de FPS, no se consideraron estos Δt marcados en azul. Creemos que puede deberse a una escasez de memoria RAM, por lo que cuando hay muchas imágenes almacenadas en la RAM, el sistema operativo tarda un poco en liberar algo de memoria y luego continúa con el hilo que ejecuta el software de la cámara. Se realizarán más experimentos para identificar la causa y luego solucionar o mitigar este problema.

4.3. Pruebas de vacío

Los dos primeros experimentos representan el estado de la cámara antes de la prueba de vacío. Los experimentos se llevaron a cabo en un ambiente de temperatura estable a 23 grados centígrados, mientras que la propia cámara funcionaba a unos 35 grados, aunque no contamos con los datos precisos de temperatura.

El primer tipo de experimentos tiene como objetivo mostrar diferencias en las estadísticas de imágenes, en una forma alternativa de medir la corriente oscura del sensor CMOS. El segundo tipo de experimentos muestra los diferentes fotogramas por segundo (FPS) que logra el software de la cámara en diferentes plataformas, entre las que destaca la RPI0W ya que es la computadora la que alimentará y controlará la cámara.

Para medir la corriente oscura, se tomaron un conjunto de imágenes bloqueando toda la luz incidente en el sensor CMOS y con 5 tiempos de exposición diferentes, que van desde 100 microsegundos (μs) hasta 100 mili segundos. Por cada tiempo de exposición, se tomaron 50 imágenes en negro.

Considere ahora las imágenes tomadas con un tiempo de exposición de $100 \mu s$. Para cada imagen, calculamos el valor promedio (número digital DN) y la desviación estándar en todos los píxeles y luego, calculamos la media y la desviación de las estadísticas anteriores en todas las imágenes. Repita este proceso para los 5 conjuntos de imágenes. Los resultados obtenidos se resumen en la tabla 4.2.

Tabla 4.2: Estadísticas de imágenes negras previo al vacío

black	100us	1ms	10ms	20ms	100ms
mean over mean	98.70	98.55	99.06	99.06	100.01
std over mean	0.044	0.038	0.035	0.041	0.072
mean over std	2.72	2.80	2.95	3.13	4.44
std over std	0.055	0.056	0.003	0.006	0.019

En otro experimento, apuntamos la cámara directamente frente a una superficie blanca (una hoja de papel) y tomamos 50 imágenes considerando 3 tiempos de exposición diferentes. Calculamos las mismas estadísticas descritas anteriormente para las imágenes en negro. La tabla 4.3 muestra los resultados obtenidos para el conjunto de imágenes en blanco.

Tabla 4.3: Estadísticas de imágenes blancas previo al vacío

white	1ms	10ms	20ms
mean over mean	6067.26	60058.7	65535
std over mean	20.83	14.86	0
mean over std	658.54	5770.70	0
std over std	28.76	5.16	0

Después de colocar la cámara al vacío, se realizaron los mismos experimentos descritos anteriormente, con el fin de establecer si la cámara sufrió una degradación en su rendimiento. La tabla 4.4 muestra las estadísticas obtenidas para el conjunto de imágenes negras como se explicó anteriormente, considerando los mismos valores para el tiempo de exposición. La tabla 4.5 muestra las estadísticas obtenidas para el conjunto de imágenes en blanco tomadas en un experimento similar al anterior con la hoja de papel en blanco.

Tabla 4.4: Estadísticas de imágenes negras posterior al vacío

black	$100\mu s$	1ms	10ms	20ms	100ms
mean over mean	98.57	98.55	98.92	98.98	100.14
std over mean	0.147	0.180	0.267	0.118	0.292
mean over std	2.74	2.79	2.98	3.20	4.63
std over std	0.010	0.017	0.055	0.015	0.140

Tabla 4.5: Estadísticas de imágenes blancas posterior al vacío

white	1ms	10ms	20ms
mean over mean	9104.7	65519.8	65535
std over mean	136.56	0.63	0
mean over std	1079.19	216.76	0
std over std	45.37	5.55	0

A partir de esto, se pueden detectar algunas diferencias en la media sobre la desviación estándar para todos los tiempos de exposición, pero mayores para los 100 ms. Esto puede explicarse por la variación de la temperatura interna de la cámara, ya que las imágenes tomadas después del vacío se capturaron continuamente sin dejar que se enfríe. Si la cámara hubiera sufrido un mal funcionamiento importante debido al entorno de vacío, habría dado resultados considerablemente diferentes que no pueden explicarse por la dependencia de la temperatura de la corriente oscura. Sin embargo, a partir de este punto, todas las imágenes tomadas llevarán metadatos que indiquen la medición de temperatura actual de los 3 sensores colocados dentro de la cámara, el sensor CMOS, el sensor de la fuente de alimentación y el sensor interno.

Por otro lado, las diferencias en las estadísticas de las imágenes blancas se pueden explicar por la variación de la iluminación al apuntar a la superficie blanca. La conclusión es que la cámara pasó con éxito las pruebas de vacío sin que se observara degradación en el rendimiento. Por lo tanto, puede continuar con más experimentos.

4.4. Ganancia

Para estimar la ganancia del sensor se siguieron los siguientes pasos. La suma se usa para eliminar el factor 2 que aparece en La varianza al restar la imágenes.

1. Ajustar el monocromador a la longitud de onda con la mayor eficiencia cuántica en el sensor.
2. Fijar un rango de tiempos de exposición y definir P valores equiespaciados.
3. Para cada tiempo de exposición tomar 2 imágenes, $img1$ e $img2$.
4. Determinar una región cuadrada de píxeles R de 200x200 en el centro de la imagen.
5. Calcular el valor promedio de los píxeles en dicha región para ambas imágenes y sumarlos. Se tiene el valor $mean = mean(img1[R]) + mean(img2[R])$.
6. Calcular una imagen con la diferencia de ambas imágenes, $diff = img1 - img2$.
7. Calcular la varianza de los píxeles de la imagen $diff$ sobre la región R . Se tiene el valor $variance = var(diff[R])$.
8. Ajustar una curva sobre los P pares de valores $(mean, variance)$ con regresión lineal.

Se consideraron $P = 50$ puntos de medición sobre dos rangos distintos de tiempos de exposición. El primer rango es de 10 a 10000 μs , con pasos de 200 μs aproximadamente. Siguiendo los pasos indicados se tienen los resultados mostrados en la figura 4.6.b, donde se observa que la curva se ajusta de buena manera a los datos dando como resultado un ganancia estable en este rango. La ganancia corresponde a la inversa de la pendiente de la curva: $gain_us = 1/1.673 \approx 0.6 [e^-/DN]$.

El segundo rango es de 1 a 500 ms , con pasos de 10 ms aproximadamente. Siguiendo los pasos indicados se tienen los resultados mostrados en la figura 4.6.a. En este caso, una curva global no se ajustaba de buena manera a todos los puntos debido a la saturación de los píxeles hacia tiempos de exposición más extensos, por que se ajustaron 4 curvas por intervalos. En este caso, la saturación completa corresponde al valor 131070 $[DN]$ dado que consideramos la suma dos imágenes.

El primer intervalo de 1 a 20 ms (fit1 en verde) presenta una ganancia similar al rango de microsegundos. En el segundo intervalo de 30 a 260 ms (fit2 en azul) se tiene $gain_ms_2 = 1/1.422 \approx 0.7 [e^-/DN]$. En el tercer intervalo de 270 a 440 ms (fit3 en rojo) se observa mayor dispersión respecto al ajuste, donde $gain_ms_3 = 1/1.117 \approx 0.9 [e^-/DN]$. Ya en el último intervalo de 450 a 500 ms (fit4 en cian) el ajuste pierde sentido al entrar a la zona de saturación.

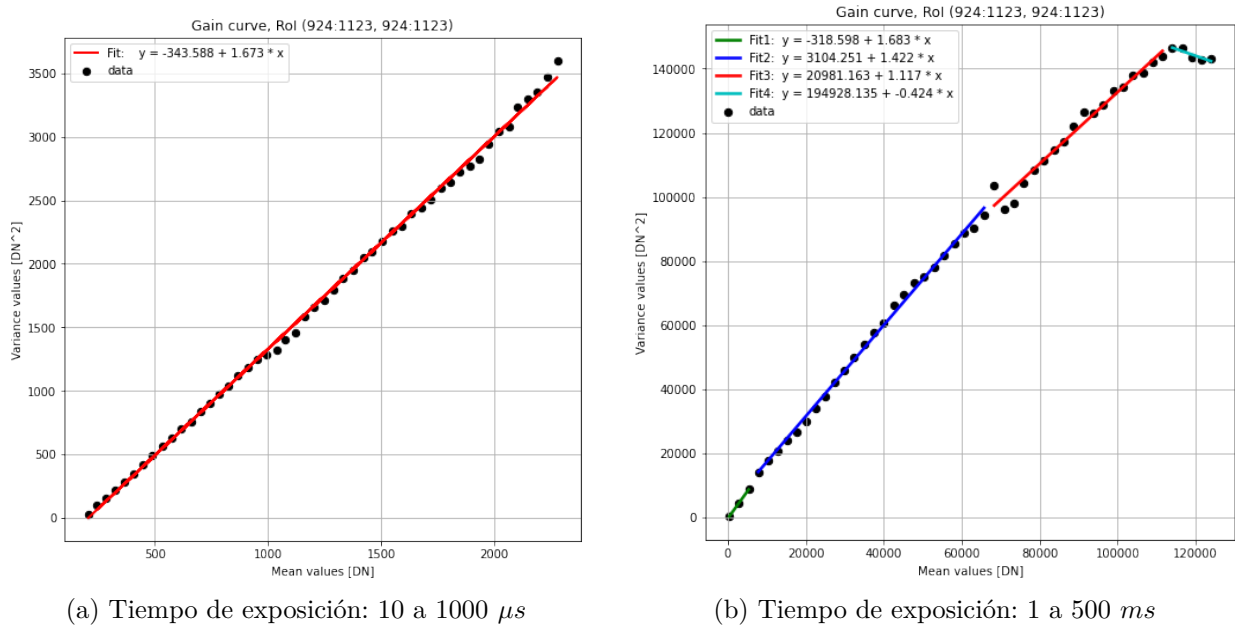


Figura 4.6: Ganancia del sensor

Con estos resultados se comprueba en mayor medida el valor de ganancia indicado por el fabricante de $0.7 [e^-/DN]$, comprobando además que la ganancia del sensor disminuye al ir aumentando las cuentas (DN), es decir, se necesitan menos fotoelectrones para incrementar cada cuenta, dejando fuera, ciertamente, a la región de saturación.

4.5. Linealidad

Para estimar la linealidad del sensor se siguieron los mismos pasos utilizados para la ganancia exceptuando los pasos 6 y 7, y, en este caso, el ajuste de curva indicado en el paso 8 es sobre los P pares de valores ($mean, exp_time$).

Se utilizaron los mismos datos obtenidos en la sección anterior. En la figura 4.7 se muestra a la izquierda la curva de linealidad obtenida donde a simple vista se aprecia un ajuste “perfecto”. A la derecha se muestra un gráfico de la no-linealidad, es decir, las desviaciones de cada punto a la línea ajustada. Calculando un promedio robusto sobre los porcentajes de no-linealidad se tiene un valor de 0.052 %.

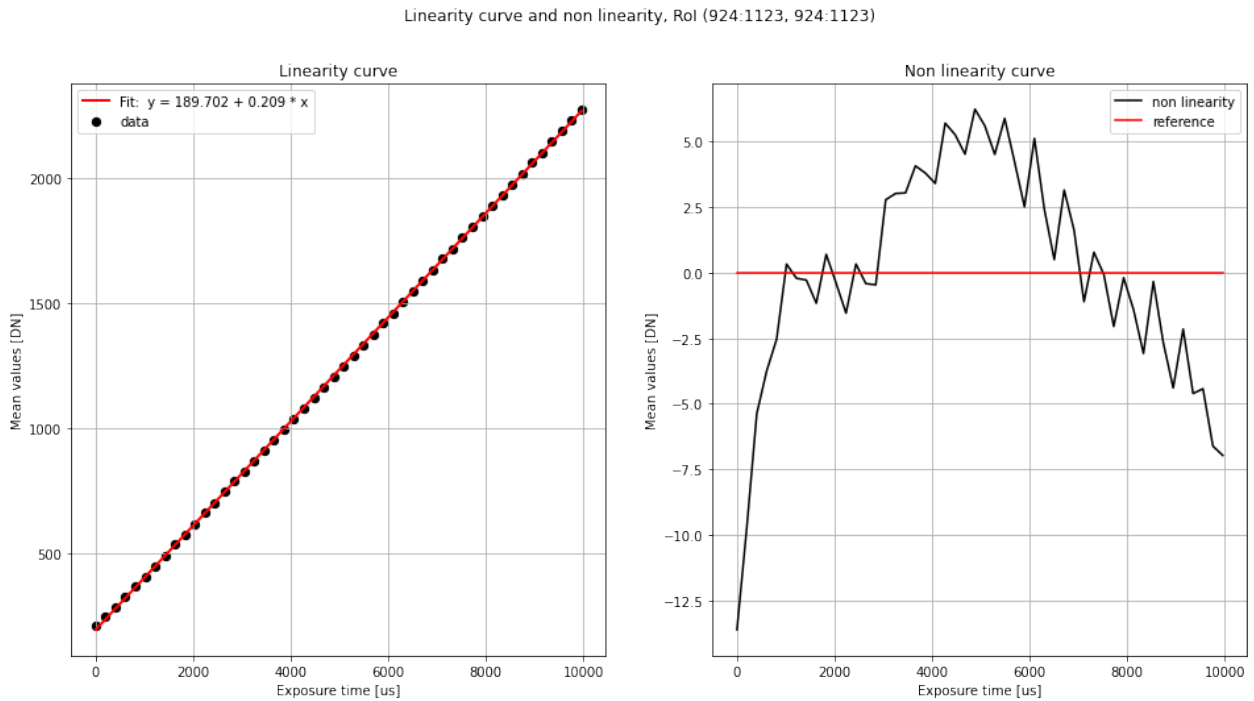


Figura 4.7: Linealidad del sensor en el rango de 10 a 1000 μs

Para el segundo rango, se muestran los resultados en la figura 4.8, arrojando un valor de no-linealidad de 0.147. Estos valores de linealidad están dentro del margen indicado por el fabricante, que asegura una no linealidad menor al 0.6 %.

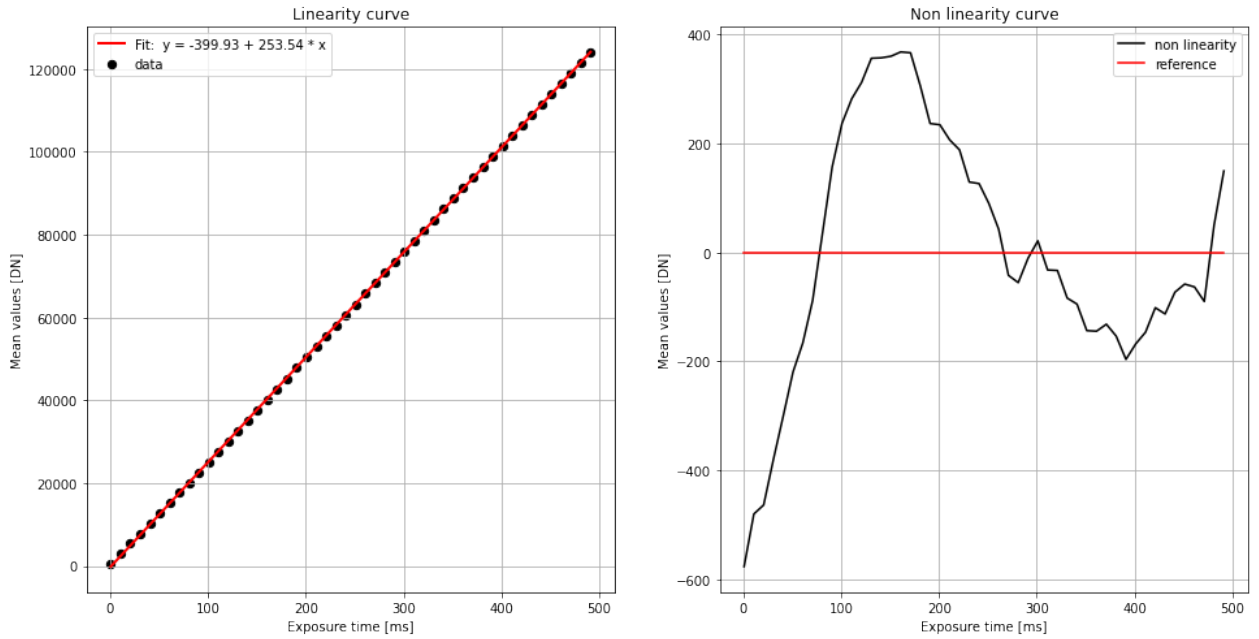


Figura 4.8: Linealidad del sensor en el rango de 1 a 500 ms

4.6. Eficiencia Cuántica

Al estimar la eficiencia cuántica ya sea relativa o absoluta se consideran 2 configuraciones experimentales, una con el objetivo montado al sensor y otra sin el objetivo, puesto que el lente influye en la incidencia de fotones sobre el sensor. Al montar el lente se aumenta la distancia del sensor a la abertura de la esfera integradora, como se observa en la imagen INSERTAR IMAGEN a diferencia de la configuración sin lente REFERENCIA IMAGEN, por lo que el flujo de fotones que inciden en el sensor es menor, requiriendo un tiempo de exposición mayor para evitar el efecto del ruido en imágenes con bajas cuentas.

4.6.1. Relativa

Para estimar la eficiencia cuántica relativa (QE_{rel}) del sensor se siguieron los siguientes pasos.

1. Fijar un rango de longitudes de onda para el monocromador y definir P valores equidistantes.
2. Fijar un tiempo de exposición tal que se evite tanto la saturación en el rango de mayor respuesta espectral como el efecto del ruido en zonas de menor respuesta.
3. Para cada longitud de onda tomar N imágenes y tomar K mediciones de de la potencia medida por el foto-diodo.
4. Para cada conjunto de imágenes calcular la imagen media, $mean_img = \frac{1}{N} \sum_{n=1}^N img_n$.
5. Calcular la QE_{rel} de acuerdo a la ecuación 1.1.

En este experimento se consideraron $P = 51$ valores de longitud de onda sobre el rango de 300 a 1000 nm , con pasos de 14 nm ; se tomaron $N = 5$ imágenes y se realizaron $K = 20$ mediciones para cada punto. En el caso de la configuración **sin** lente se determinó utilizar un tiempo de exposición igual a 276 ms , mientras que en la configuración **con** lente se fijó en 500 ms , el máximo valor posible, y se debió aumentar la potencia de la lámpara dado que la pérdida de flujo fue considerable.

En la figura 4.9.a se muestra la QE_{rel} para el caso **sin** lente. Al comparar esta curva con la QE indicada por el fabricante (figura 1.2.b) se comprueba que el peak de respuesta está alrededor de los 550 nm y la forma general de la curva desde los 400 hasta los 1000 nm , sin embargo, se observa una incongruencia bajo los 400. De acuerdo al fabricante, al descender de los 400 nm se llega a un mínimo de eficiencia de un 30 % cercano a los 300 nm para luego subir hasta una eficiencia del 50 % en torno a los 250 nm , por otro lado, la curva obtenida alcanza el mínimo de eficiencia de un tercio relativo al peak, valor que si se corresponde, cercano a los 350 nm para luego volver a subir, como es esperado. Es muy probable que esta diferencia observada sea debido a que se está operando el monocromador cercano a su límite inferior de longitud de onda.

Por otro lado, en la figura 4.9.b se muestra la QE_{rel} para el caso **con** lente. Se observa el efecto del lente al obtener una más suave, sin embargo, se obtiene una respuesta anormal en torno a los 300 nm puesto que la eficiencia aumenta considerablemente, llegando incluso a ser similar la eficiencia en los 300 nm con aquella en los 550 nm .

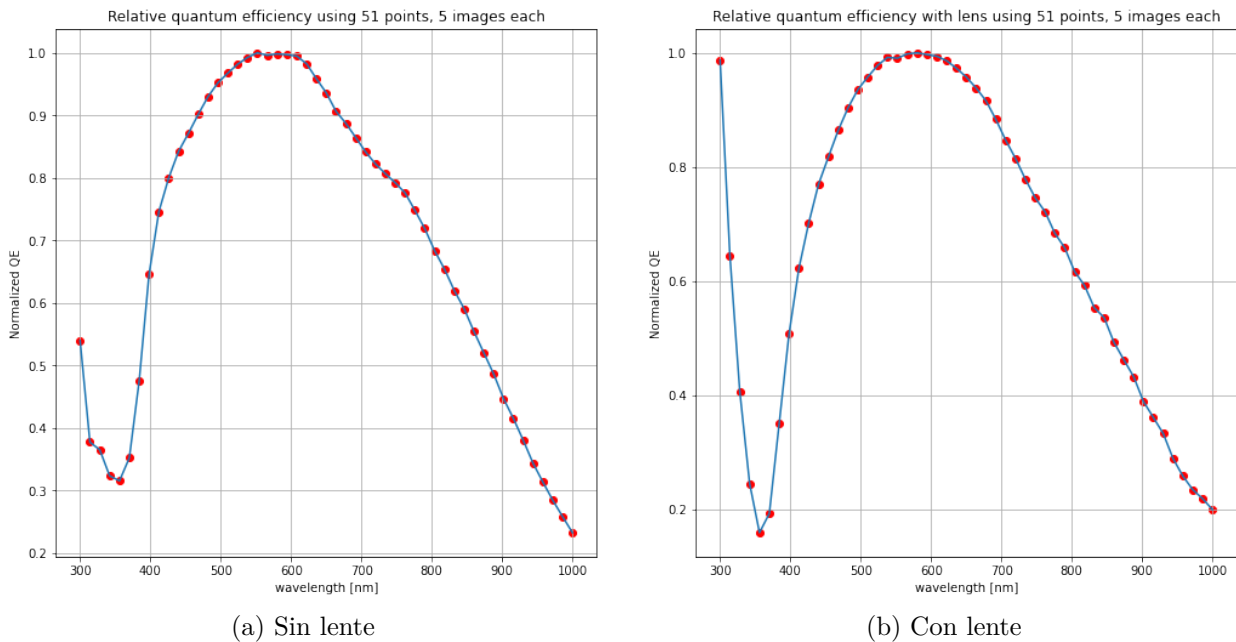


Figura 4.9: Eficiencia Cuántica relativa

4.6.2. Absoluta

Para estimar la eficiencia cuántica absoluta (QE_{abs}) del sensor se siguieron los siguientes pasos.

1. Fijar un rango de longitudes de onda para el monocromador y definir P valores equidistantes.
2. Fijar un tiempo de exposición tal que se evite tanto la saturación en el rango de mayor respuesta espectral como el efecto del ruido en zonas de menor respuesta.
3. Para cada longitud de onda tomar N imágenes y promediar K mediciones de de la potencia recibida por el foto-diodo (P_1).
4. Para cada conjunto de imágenes calcular la imagen media, $mean_img = \frac{1}{N} \sum_{n=1}^N img_n$.
5. Colocar el foto-diodo en la posición del sensor, medir la potencia recibida para cada longitud de onda (P_2) y calcular la proporción respecto al potencia medida en el paso 3. Se tiene el valor $pw_{ratio} = P_1/P_2$.
6. Calcular la QE_{abs} de acuerdo a la ecuación 4.10.

En este experimento se consideraron las mismas imágenes capturadas para el experimento de la eficiencia cuántica absoluta en el caso **con** el lente. Además, se realizaron mediciones de la potencia percibida por el fotodiodo en el mismo lugar donde estaba el sensor de la cámara. Con esto datos se estimó de manera experimental la QE_{abs} , cuyos resultados se muestran en en la figura 4.10.

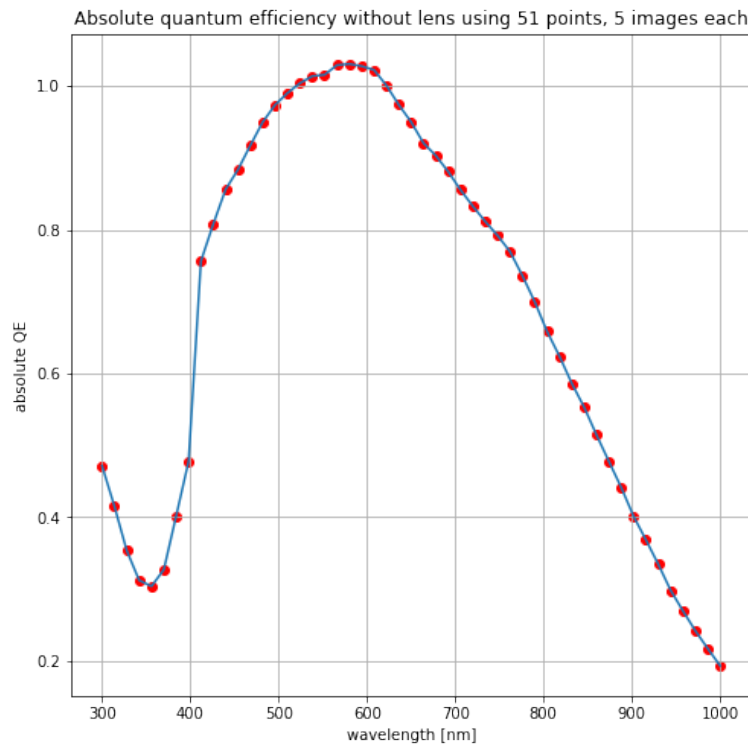


Figura 4.10: Linealidad del sensor en el rango de 1 a 500 ms

De acuerdo a los obtenido, se puede verificar que la forma de la curva de eficiencia es similar a la presentada en la obtenida para la QE_{rel} , sin embargo, se pueden apreciar valores inconsistentes de eficiencia en torno al máximo de respuesta alrededor de los 550nm. En esta porción de la curva la eficiencia pareciera indicar que sobrepasa el 100%, lo cual es imposible y se puede explicar por una sobre estimación del flujo de fotones captado por el fotodiodo puesto que, como se señaló anteriormente, la configuración experimental tuvo que ser adaptada para trabajar con este modelo de cámara por lo que posicionar el fotodiodo en la posición correcta y además bloqueando todo fuente de luz externa resultó ser un trabajo arduo y prácticamente imposible, por lo que realizó una estimación del flujo real que resulto ser aproximadamente correcta pero lo suficiente como para validar la curva entregada de respuesta por el fabricante.

4.7. Ruido de disparo

Para estimar el *Shot Noise* (σ_S) del sensor se siguieron los siguientes pasos.

1. Tomar N imágenes con el mínimo tiempo de exposición bloqueando cualquier posible fuente de luz que incida en el sensor.
2. Seleccionar una muestra de imágenes donde la temperatura del sensor sea la misma.
3. Calcular una imagen promedio de la muestra.
4. Usar la moda de la imagen media como señal.
5. El *Shot Noise* es la raíz cuadrada de la señal determinada.

En este experimento se tomaron $N = 100$ imágenes con $10 \mu s$ de tiempo de exposición. Luego se seleccionaron aquellas imágenes en las que sensor indica una temperatura de $32^\circ C$, contabilizando 82 muestras, y se procede según los pasos antes indicados. En la figura 4.11 se muestra el histograma normalizado calculado a partir de los valores de píxeles de la imagen media. La moda de los píxeles es de $99.04878[DN]$ lo que, según la ganancia del sensor en determinada para este número de cuentas, corresponde a $59[e^-]$, resultando en un *Shot Noise* igual a $\sigma_S = \sqrt{59} = 7.68[e^-]$.

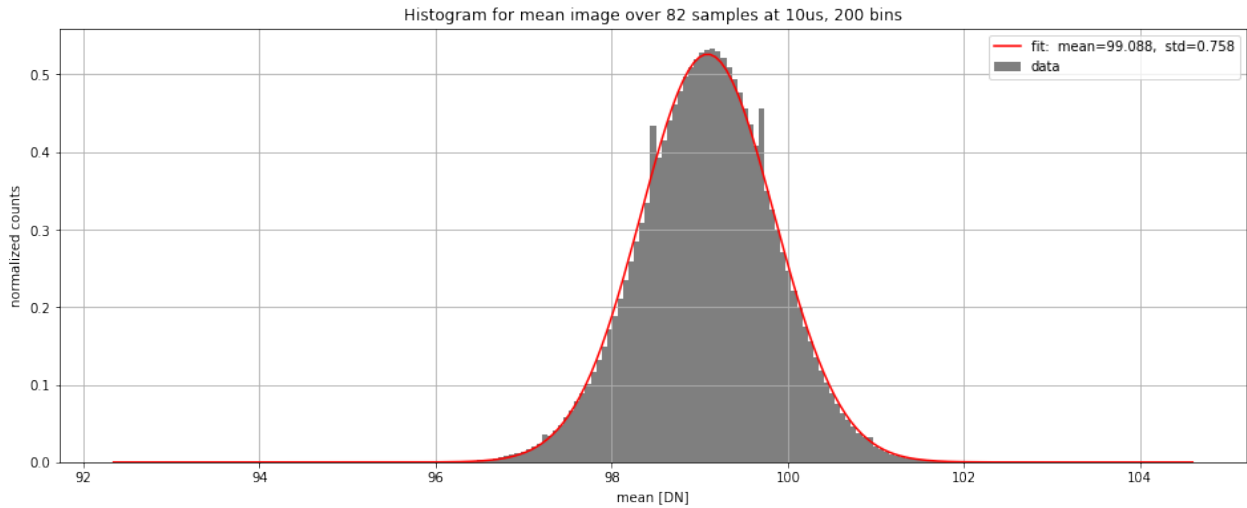


Figura 4.11: Histograma normalizado sobre píxeles de la imagen media

4.8. Point spread function PSF

En este experimento se utilizó el montaje mostrado en la figura para apuntar el láser de He/Ne y crear una fuente puntual sobre el sensor con el lente montado a la cámara. En la figura 4.12 se muestra la región de píxeles alrededor de aquel que recibe esta fuente de luz puntual, donde sólo el píxel central alcanza la saturación.

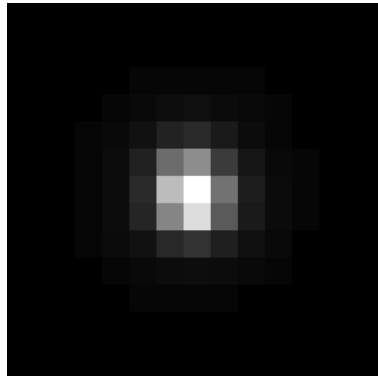


Figura 4.12: Región de incidencia de la fuente puntual

A partir de esta imagen se calcula un ajuste gaussiano en ambos ejes. En la figura 4.13.a se muestra el perfil gaussiano por fila, donde se obtiene un ancho de 2.56 píxeles (valor de FWHM), mientras que en la figura 4.13.b se muestra el perfil gaussiano por columna, donde se obtiene un ancho de 2.21 píxeles (FWHM).

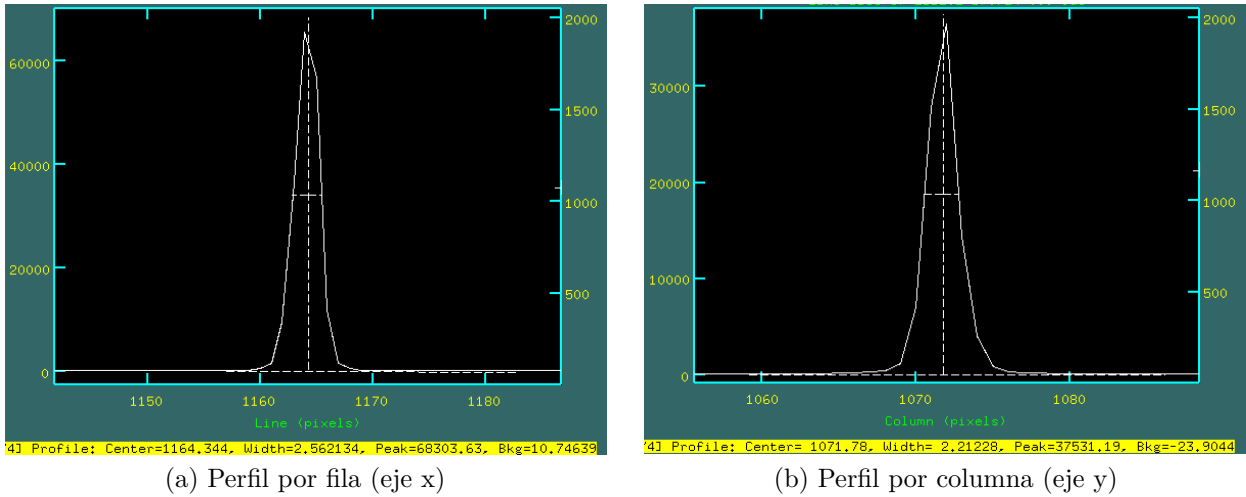


Figura 4.13: Perfil gaussiano sobre la región de incidencia

Por otro lado, se calcula también un ajuste gaussiano en dos ejes (3D). En la figura 4.14 se muestra un gráfico del ajuste, donde se obtiene un radio de 2.7 píxeles aproximadamente de dispersión para la fuente de luz puntual incidente sobre el píxel central.

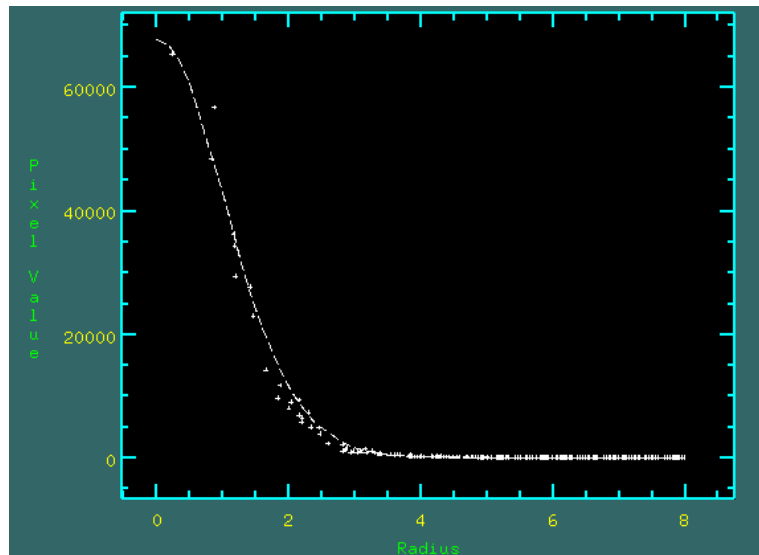


Figura 4.14: Región de incidencia de la fuente puntual

4.9. Calibración

Siguiendo el procedimiento para la calibración detallado en la sección experimental, se obtuvo un conjunto de imágenes sobre el cual se eligió detectar un patrón de círculos de tamaño 9 por 6, logrando encontrar dicho patrón en 25 imágenes por medio del algoritmo de detección de círculos. Con esta cantidad de datos es posible estimar los parámetros de calibración con suficiente confianza estadística.

En la figura 4.15 se muestra la distorsión radial en píxeles sobre la región abarcada por

el tamaño del sensor (13.5 por 13.5 mm), notando que los mayores efectos de distorsión son en las esquinas de las imágenes y que la distorsión es positiva, es decir, hacia afuera, sin embargo, el efecto de la distorsión radial es prácticamente despreciable ya que su máxima magnitud en los 4 píxeles de las esquinas es de apenas 0.078 píxeles.

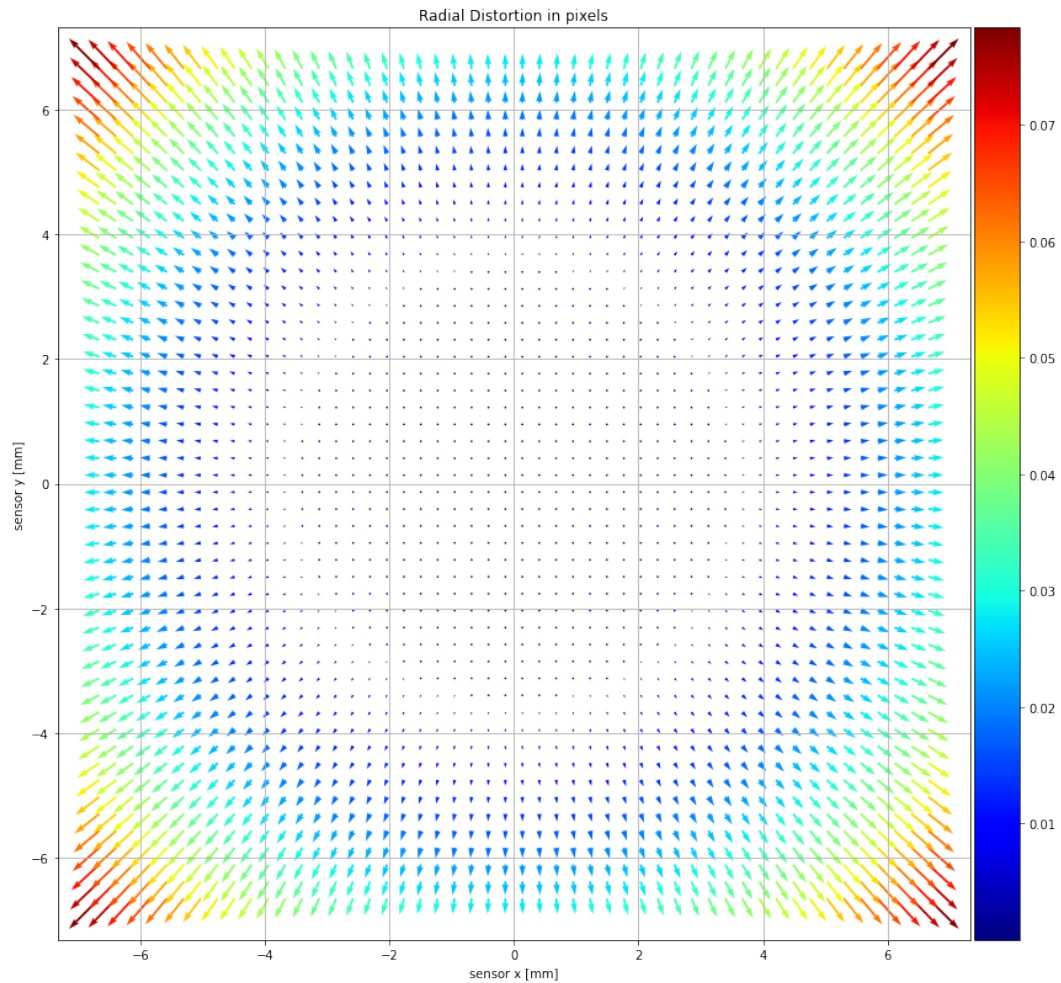


Figura 4.15: Distorsión radial en píxeles

Por otro lado, en la figura 4.16 se muestra la distorsión tangencial sobre la misma región, observando nuevamente que los mayores efectos se dan en las esquinas de la imagen. En este caso, la magnitud de la distorsión es mayor donde, por ejemplo, en píxel superior izquierdo se tiene una magnitud de 0.43 píxeles de distorsión, lo cual se mantiene dentro de rango aceptables considerando que la distorsión máxima entregada por el fabricante del lente es de un 0.05 %, es decir, de aproximadamente 1 píxel con un sensor de tamaño 2048x2048.

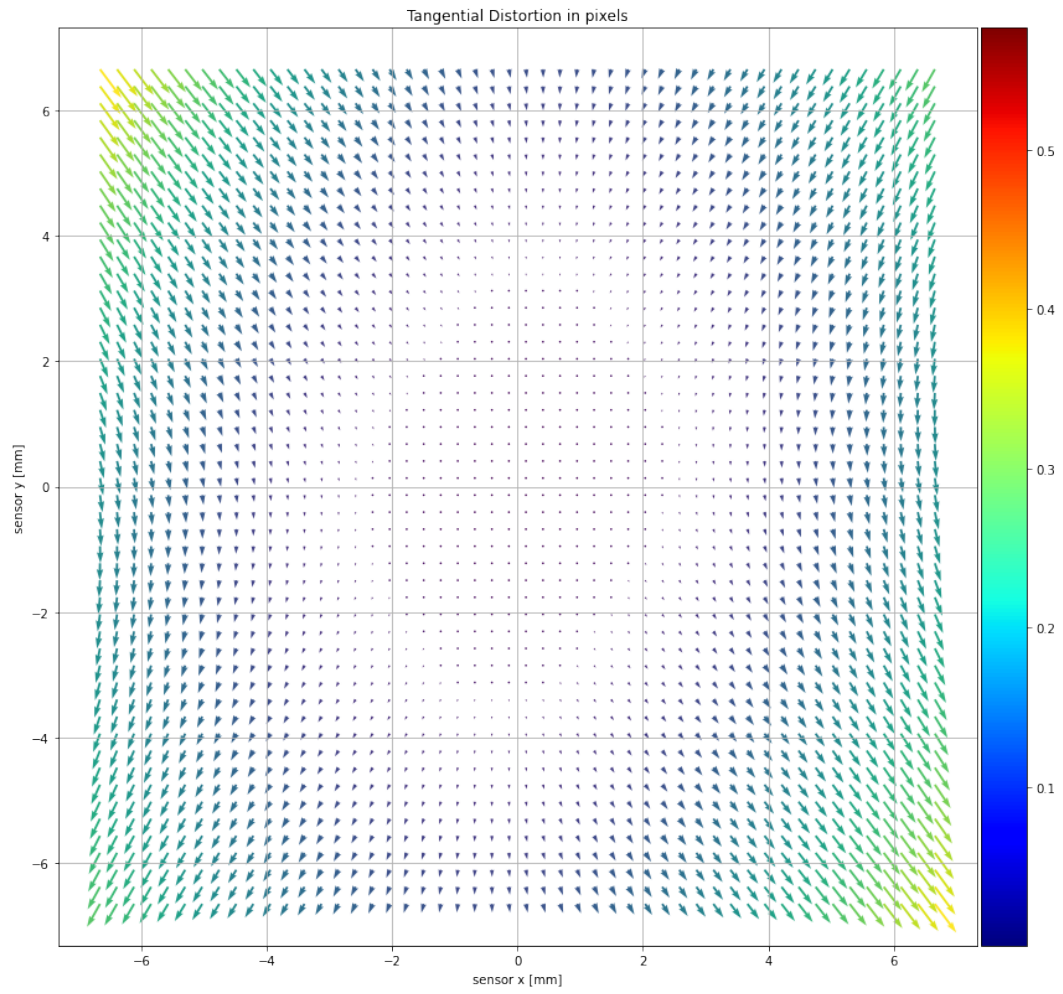


Figura 4.16: Distorsión tangencial en píxeles

Capítulo 5

Conclusiones

5.1. Integración de la cámara

A pesar de que no se pudo acoplar el switch USB para regular el encendido y apagado, ésta aún se puede operar bajo las restricciones de consumo de energía impuestas por OBC y la energía disponible en la batería. De esta manera se montó la cámara en su estructura especialmente diseñada para mantenerla alineada según los ejes que se muestran en la figura 2.1, estructura que no se aprecia en el esquema y que fue impresa en material Windform. La cámara se conectó directamente por USB a la placa del STT y se ha comprobado su correcto encendido y apagado a través de ésta. Se concluye entonces que se ha cumplido con el objetivo de integrar la cámara físicamente al SUCHAI 2.

La versión del SFS desarrollada específicamente para el *payload* conjunto PCO+STT fue instalada en la RPI0W embebida en la placa del STT. Se han ejecutado y comprobado todos los comandos presentados en la sección 3, verificando así que es posible comandar la cámara de manera remota y realizar las tareas esenciales requeridas, entre ellas, captura de imágenes y generación de metadatos, compresión de imágenes, procesamiento de imágenes, manejo de la base de imágenes, extracción de información en forma de histogramas y envío de imágenes a tierra. Se concluye entonces que se ha cumplido con el objetivo de integrar la cámara al SUCHAI 2 a nivel de software.

Además, se han elaborado documentos que describen las fases de integración de la cámara, detallando la partes mecánicas involucradas, las conexiones eléctricas y los requerimientos de energía, y también guías de operación de la cámara que describen el uso de los comandos, la función que desempeña cada uno, los parámetros recibidos y los valores retornados. Se ha cumplido con documentar en detalle todo el trabajo, lo que constituirá material de estudio y revisión para futuros trabajos y misiones.

5.2. Caracterización del sensor

Aunque la caracterización del sensor de la cámara no juegue un rol preponderante en las fases de integración de la cámara, es importante validar sus características para el posterior análisis de la imágenes. Tener conocimiento certero del proceso de captura de la luz incidente que será objeto de estudio es vital para la corrección de errores inherentes al proceso de captura y en especial para entender la naturaleza de las fuentes de contaminación lumínica.

En esta línea, de acuerdo a los resultados presentados en la sección 4 se determina que la cámara cumple con las características técnicas y especificaciones entregadas por el fabricante. Es de interés en particular para esta misión caracterizar la eficiencia cuántica del sensor ya que uno de los objetivos de la misión es estudiar el impacto de las luces LED blancas sobre la contaminación lumínica en el norte de Chile, verificando que este modelo en particular de cámara será de utilidad. Por otro lado, la información obtenida sobre la PSF del sensor será de utilidad para comprender el proceso de formación de imágenes en la cámara a partir de la luz incidente en el sensor proveniente de fuentes de luz en tierra dentro del rango espectral de captura, y así comprender la naturaleza de dichas fuentes para determinar su intensidad, su forma y su posición a partir de las imágenes recibidas en la estación terrena.

Respecto al sistema sensor+lente, se han estimado los efectos de distorsión en las imágenes debido al proceso de captura y se concluye que en la zonas de mayor distorsión geométrica se mantiene dentro de rango aceptables, aun así, esta información permitirá mejorar el proceso de corrección de imágenes. Además, se han documentado todos los resultados obtenidos y detallado los experimentos llevados a cabo con el fin de caracterizar el sensor, lo que constituirá material de estudio tanto para el SPEL como para el fabricante PCO. Se concluye entonces que se cumple con el objetivo de caracterizar el sensor.

5.3. Trabajo futuro

A la fecha de concluido este trabajo el SUCHAI 2 se encuentra el espacio a la espera de ser puesto en su órbita designada, luego comenzará la fase de operación del satélite donde se pondrá a prueba todo el trabajo desarrollado. Se espera poder a extraer datos dentro de los días siguientes al inicio de operaciones y comprobar finalmente la captura de fotos desde es espacio. Los primeros pasos serán determinar la calidad de las fotos a bordo mediante el cálculo y envío de histogramas y mediante el envío de miniaturas a tierra, y luego, cuando se determine cuales fotos contiene información relevante, se enviarán a tierra par comenzar la fase de estudio de la contaminación lumínica.

Por otra parte, se continuará con el desarrollo de los procesos de georeferenciación y geolocalización, necesarios para estudio de las imágenes y que se verá beneficiado al contar con datos reales, es decir, imágenes *raw* que deberán ser corregidas y sus fuentes de luz debidamente localizadas.

Bibliografía

- [1] F. Hölker, C. Wolter, E. K. Perkin, K. Tockner, *et al.*, “Light pollution as a biodiversity threat.,” *Trends in ecology & evolution*, vol. 25, no. 12, pp. 681–682, 2010.
- [2] R. Chepesiuk, “Missing the dark: health effects of light pollution,” 2009.
- [3] “Cubesat design specification.” <https://static1.squarespace.com/static/5418c831e4b0fa4ecac1bacd/t/5f24997b6deea10cc52bb016/1596234122437/CDS+REV14+2020-07-31+DRAFT.pdf>. Accessed: 2021-05-05.
- [4] D. Selva and D. Krejci, “A survey and assessment of the capabilities of cubesats for earth observation,” *Acta Astronautica*, vol. 74, pp. 50–68, 2012.
- [5] L. Kosmidis, I. Rodriguez, Álvaro Jover, S. Alcaide, J. Lachaize, J. Abella, O. Notebaert, F. J. Cazorla, and D. Steenari, “Gpu4s: Embedded gpus in space - latest project updates,” *Microprocessors and Microsystems*, vol. 77, p. 103143, 2020.
- [6] F. Falchi, P. Cinzano, D. Duriscoe, C. C. Kyba, C. D. Elvidge, K. Baugh, B. A. Portnov, N. A. Rybnikova, and R. Furgoni, “The new world atlas of artificial night sky brightness,” *Science advances*, vol. 2, no. 6, p. e1600377, 2016.
- [7] T. Toutin, “Review article: Geometric processing of remote sensing images: models, algorithms and methods,” *International Journal of Remote Sensing*, vol. 25, no. 10, pp. 1893–1924, 2004.
- [8] T. Toutin, “State-of-the-art of geometric correction of remote sensing data: a data fusion perspective,” *International Journal of Image and Data Fusion*, vol. 2, no. 1, pp. 3–35, 2011.
- [9] A. Hackeloeer, K. Klasing, J. M. Krisp, and L. Meng, “Georeferencing: a review of methods and applications,” *Annals of GIS*, vol. 20, no. 1, pp. 61–69, 2014.
- [10] Princeton Instruments, “cientific cmos (scmos) cameras: The basics — educational notes.” <https://www.princetoninstruments.com/learn/camera-fundamentals/scmos-the-basics>, 2022. [Online; accessed 9-April-2022].
- [11] J. Ohta, *Fundamentals of CMOS image sensors*. CRC Press, 2020.
- [12] M. Bigas, E. Cabruja, J. Forest, and J. Salvi, “Review of cmos image sensors,” *Microelectronics Journal*, vol. 37, no. 5, pp. 433–451, 2006.
- [13] Princeton Instruments, “Camera gain — educational notes.” <https://www.princetoninstruments.com/learn/camera-fundamentals/gain>, 2022. [Online; accessed 9-April-2022].
- [14] Princeton Instruments, “Full well capacity and pixel saturation — educational notes.” <https://www.princetoninstruments.com/learn/camera-fundamentals/full-well-cap>

- [acity-pixel-saturation](#), 2022. [Online; accessed 9-April-2022].
- [15] B. A. Fowler, A. El Gamal, D. X. Yang, and H. Tian, “Method for estimating quantum efficiency for cmos image sensors,” in *Solid State Sensor Arrays: Development and Applications II*, vol. 3301, pp. 178–185, International Society for Optics and Photonics, 1998.
- [16] Princeton Instruments, “Scientific camera noise sources — educational notes.” <https://www.princetoninstruments.com/learn/camera-fundamentals/scientific-camera-noise-sources>, 2022. [Online; accessed 9-April-2022].
- [17] J. J. Braat, S. van Haver, A. J. Janssen, and P. Dirksen, “Chapter 6 assessment of optical systems by means of point-spread functions,” vol. 51 of *Progress in Optics*, pp. 349–468, Elsevier, 2008.
- [18] Z. Zhang, “A flexible new technique for camera calibration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.
- [19] OpenCV, “Camera calibration.” https://docs.opencv.org/3.4/dc/dbb/tutorial_py_calibration.html, 2022. [Online; accessed 9-April-2022].
- [20] G. Bradski, “The OpenCV Library,” *Dr. Dobb’s Journal of Software Tools*, 2000.
- [21] C. E. Gonzalez and C. J. Rojas and A. Bergel and M. A. Diaz, “Suchai flight software repository on gitlab.” <https://gitlab.com/spel-uchile/suchai-flight-software>, 2022. [Online; accessed 9-April-2022].
- [22] C. E. Gonzalez, C. J. Rojas, A. Bergel, and M. A. Diaz, “An architecture-tracking approach to evaluate a modular and extensible flight software for cubesat nanosatellites,” *IEEE Access*, pp. 1–1, 2019.
- [23] Refactoring GURU, “Command design pattern.” <https://refactoring.guru/design-patterns/command/>, 2022. [Online; accessed 9-April-2022].
- [24] Wikipedia contributors, “Unified modeling language — Wikipedia, the free encyclopedia.” https://en.wikipedia.org/w/index.php?title=Unified_Modeling_Language&oldid=1077269626, 2022. [Online; accessed 9-April-2022].
- [25] Wikipedia contributors, “Usb 3.0 — Wikipedia, the free encyclopedia.” https://en.wikipedia.org/w/index.php?title=USB_3.0&oldid=1078818281, 2022. [Online; accessed 9-April-2022].
- [26] 8086, “8086 products.” <https://www.8086.net/products>, 2022. [Online; accessed 9-April-2022].
- [27] PCO, “PCO development tools.” <https://www.pco.de/software/development-tools/>, 2022. [Online; accessed 9-April-2022].

Anexo A

Escala de Bortle

A.1. Escala de Bortle















Ratio to natural brightness	Artificial brightness ($\mu\text{cd}/\text{m}^2$)	Approximate total brightness (mcd/m^2)	Color	
<0.01	<1.74	<0.176	Black	
0.01–0.02	1.74–3.48	0.176–0.177	Dark gray	
0.02–0.04	3.48–6.96	0.177–0.181	Gray	
0.04–0.08	6.96–13.9	0.181–0.188	Dark blue	
0.08–0.16	13.9–27.8	0.188–0.202	Blue	
0.16–0.32	27.8–55.7	0.202–0.230	Light blue	
0.32–0.64	55.7–111	0.230–0.285	Dark green	
0.64–1.28	111–223	0.285–0.397	Green	
1.28–2.56	223–445	0.397–0.619	Yellow	
2.56–5.12	445–890	0.619–1.065	Orange	
5.12–10.2	890–1780	1.07–1.96	Red	
10.2–20.5	1780–3560	1.96–3.74	Magenta	
20.5–41	3560–7130	3.74–7.30	Pink	
>41	>7130	>7.30	White	

Figura A.1: Escala de Bortle