



Universidad de
Oviedo



ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN.

GRADO EN INGENIERÍA INFORMÁTICA EN TECNOLOGÍAS DE LA INFORMACIÓN

ÁREA DE LENGUAJES Y SISTEMAS INFORMÁTICOS

IMPLEMENTACIÓN DE TÉCNICAS DE ESTABILIZACIÓN PARA OBSERVACIONES ASTRONÓMICAS EN GLOBOS DE ALTA ALTITUD

D. SÁNCHEZ NOSTI, Pablo

TUTOR: D. MUÑIZ SÁNCHEZ, Rubén

COTUTOR: D. DÍEZ ALONSO, Enrique

FECHA: Julio, 2022

Índice.

1.	Hipótesis de partida y alcance.	7
1.1.-	INTRODUCCIÓN.....	7
1.2.-	HISTORIA DE LA ASTRONOMÍA ÓPTICA.....	8
1.2.1.-	La astronomía óptica en la antigüedad.	8
1.2.2.-	La astronomía óptica en la Edad Moderna.	9
1.2.3.-	La astronomía óptica en la actualidad.	9
1.3.-	OBSERVATORIOS ESPACIALES.	9
1.3.1.-	Historia de los observatorios espaciales.	10
1.3.2.-	Desventajas de los observatorios espaciales.....	11
1.4.-	ALCANCE.	14
1.5.-	EXPERIMENTOS CON TELESCOPIOS EN GLOBOS SONDA EN LA ACTUALIDAD.....	14
1.5.1.-	Globo con instrumentos científicos por el Pacífico.....	14
1.5.2.-	Proyecto ORISON.....	15
1.5.3.-	ASTHROS.....	15
1.5.-	CONDICIONES DEL EXPERIMENTO.....	16
1.6.-	PASOS PARA EL LANZAMIENTO DE UN GLOBO SONDA.	18
1.6.1.-	Características del experimento.....	18
1.6.2.-	Pruebas previas.....	20
1.6.3.-	Permisos necesarios.....	21
1.6.4.-	Lugar de lanzamiento.	21
1.7.-	ANÁLISIS DE ALTERNATIVAS.	23
1.7.1.-	Raspberry.....	24

1.7.2.-Cámara QHY 5-II monocolor.....	25
1.7.3.-Astroberry.....	26
1.7.4.-Stellarium.....	26
1.7.5.-OpenCV.....	28
1.7.6.-FireCapture.....	28
1.7.7.- Astrometry.net.....	29
1.7.8.- Sensor IMU.....	30
2. Objetivos concretos y relación con el estado actual.....	32
3. Metodología de trabajo.....	34
3.1.- DESCRIPCIÓN DE LOS VIDEOS TOMADOS.....	39
4. Trabajo realizado y resultados obtenidos.....	42
4.1.-HARDWARE.....	45
4.2.-SOFTWARE.....	46
5. Conclusiones y trabajo futuro.....	48
5.1.- CONCLUSIONES.....	48
5.2.- TRABAJO FUTURO.....	49
A. Anexo 1: OpenCV y software desarrollado.....	51
A.1.-ALTERNATIVA DE OPENCV.....	51
A.1.1.- Clasificador de cascada.....	51
A.1.2.- Contornos en imágenes.....	51
A.1.3.- Optical Flow.....	52
A.1.4.- Opción escogida.....	52
A.2.-EXPLICACIÓN DE LOS EJES DE COORDENADAS DE OPENCV Y MOVIMIENTO DEL SERVO.....	52
A.3.-ANÁLISIS DEL CÓDIGO.....	55

A.4.- CÓDIGO.....	60
6. Bibliografía.....	65

Índice de figuras.

Figura 1.1.- Telescopio Cosmos 215	11
Figura 1.2.- Localización de los puntos de Lagrange.....	13
Figura 1.3.- Fotografía del telescopio Hubble	13
Figura 1.4.- Preparativos del lanzamiento del globo en Nueva Zelanda	15
Figura 1.5.- Ilustración de un globo a alta altitud.....	16
Figura 1.6.- Ilustración de las capas de la atmósfera terrestre.....	17
Figura 1.7.- Vista de la Tierra desde la estratosfera	18
Figura 1.8.- Simulación de vuelo	22
Figura 1.9.- Fotografía de la Raspberry Pi 4 y el servomotor utilizados.....	25
Figura 1.10.- Fotografía de la cámara QHY 5-II usada.....	26
Figura 1.11.- Logo del programa Stellarium	27
Figura 1.12.- Vista del programa Stellarium	27
Figura 1.13.- Logo de OpenCV	28
Figura 1.14.- Logo de FireCapture	28
Figura 1.15.- Logo de Astrometry.net	29
Figura 1.16.- Funcionamiento de la herramienta.....	30
Figura 1.17.- Imagen de un sensor IMU.....	31
Figura 3.1.- Punto en una hoja de papel	34
Figura 3.2.- Captura de uno de los vídeos obtenidos desde Gijón	35
Figura 3.3.- Montaje para la cámara.....	36
Figura 3.4.- Captura de uno de los vídeos obtenidos desde Nava	37
Figura 3.5.- Fotografía tomada mediante el programa de prueba de la cámara QHY 5-I.....	38
Figura 3.6.- Resultado ASCII de la fotografía tomada.....	39
Figura 4.1.- Vista del escritorio del Astroberry	44
Figura A.1.- Explicación de los ejes de OpenCV.....	53
Figura A.2.- Movimiento vertical del servo	54
Figura A.3.-Movimiento horizontal del servo	55
Figura A.4.- Vista del rastreo del programa	57

Figura A.5.- Movimiento esperado del programa	58
Figura A.6.- Movimiento esperado en caso de rotación.....	58
Figura A.7.- Movimiento esperado en caso de rotación (2)	59

1. Hipótesis de partida y alcance.

1.1.-INTRODUCCIÓN.

La astronomía observacional es la parte de la astronomía que tiene como función observar, almacenar y recopilar información del universo. Dado que los objetos a estudiar se encuentran muy lejos, no es posible realizar experimentos directos tales como tomar muestras o interactuar con ellos. Este problema se compensa observando los diferentes fenómenos estelares. Por ejemplo, de nuestra estrella más cercana, el Sol, sabemos que está compuesto en su mayoría de hidrógeno sin haber tomado ninguna muestra física. Analizando la cantidad de energía recibida en las diferentes longitudes de onda se puede averiguar para qué color llega menos luz y con ello qué material o elemento es el que absorbe esa energía ausente. Estas observaciones se pueden realizar también con otras estrellas, denominadas estrellas variables, que con su variación del brillo en el tiempo se pueden medir distancias a galaxias o detectar otros elementos cercanos que la perturban, tales como otras estrellas, exoplanetas o agujeros negros [1] [2].

Por ello, la mayoría de las observaciones astronómicas se realizan mediante el espectro electromagnético. En función del objeto a observar, se utiliza una longitud de onda u otra. Con longitudes de onda grandes, como las de radio, se observa la radiación cósmica de microondas de fondo, esto es, la luz del Big Bang. Para objetos como planetas se usan frecuencias infrarrojas y para los procesos muy energéticos como agujeros negros o púlsares binarios se usan ondas con longitudes de onda muy pequeñas como rayos ultravioletas, rayos X y rayos gamma. La radiación ultravioleta tiene una longitud de onda que va desde los 100 a los 400 nm, los rayos X va desde los 10 nm a los 0,01 nm y los rayos gamma, que tienen la longitud de onda más pequeña, es de aproximadamente 100 picómetros o menos.

Por último, se tiene la astronomía óptica o visible, que es la que usa el rango de la luz visible y en el que entraría el experimento que se va a llevar a cabo [3]. La luz visible es una

región limitada del espectro electromagnético que coincide en su mayoría con la que el ojo humano puede detectar. Su longitud de onda va aproximadamente desde los 380 a los 750 nanómetros.

1.2.-HISTORIA DE LA ASTRONOMÍA ÓPTICA.

La astronomía óptica no es algo exclusivo de la actualidad, ya que simplemente con apuntar nuestros ojos al cielo ya se pueden apreciar muchos fenómenos astronómicos, por lo que desde tiempos muy pasados sin tecnología alguna se han realizado observaciones para diversas funciones [4] [5].

1.2.1.-La astronomía óptica en la antigüedad.

Es la parte de la astronomía que se lleva investigando más tiempo, ya que desde tiempos de la prehistoria se realizaban observaciones al Sol, la Luna y diferentes planetas con los que se consiguieron establecer horarios y calendarios para predecir las diferentes estaciones y con ello el movimiento de los animales que les servían de alimento o para cuándo sembrar y recoger las cosechas.

Más adelante, otras civilizaciones realizaron obras y observatorios como el de Stonehenge, el cual se utilizaba para averiguar fechas de los solsticios, predecir eclipses y determinar la posición de la Luna a lo largo del año. Los egipcios tenían ya un calendario muy acorde al año solar. Ellos usaban la posición de la estrella Sothis (Sirio en la actualidad) para calcular cuándo empezaba aproximadamente la crecida del río Nilo. Incluso hay teorías que afirman que las pirámides de Giza están alineadas con el cinturón de Orión, pudiendo determinar así el inicio de las estaciones. Estos son solo algunos ejemplos, ya que también babilonios, griegos, mayas, aztecas, indios, chinos, árabes y muchas otras civilizaciones tuvieron observatorios y gran tradición astronómica [6].

1.2.2.-La astronomía óptica en la Edad Moderna.

Ya bastante más adelante, a principios del siglo XVII, con la invención del telescopio por parte de Galileo Galilei y el desarrollo de las leyes del movimiento planetario de Kepler, se produjo una gran revolución en la astronomía, ya que se pudo hacer observaciones más precisas a objetos tales como planetas y sus lunas, así como calcular y predecir mejor sus órbitas.

1.2.3.-La astronomía óptica en la actualidad.

Entrados ya en el siglo XX se empezaron a construir grandes telescopios, como el telescopio Hooker o el Hale. Con estos avances se consiguió, entre otros ejemplos, medir el diámetro de una estrella lejana, calcular la distancia a otras galaxias o medir la expansión del Universo. Desde entonces, se han ido construyendo muchos telescopios similares, pero más modernos, alrededor de todo el planeta.

El siguiente paso fue el lanzamiento de observatorios o telescopios espaciales. Estos son satélites o sondas que toman imágenes o información mientras orbitan la Tierra u otros cuerpos y que supusieron un gran avance, ya que con estos artilugios se consigue evitar los principales inconvenientes, los cuales serán descritos más adelante, que tienen los situados en la superficie [7].

1.3.-OBSERVATORIOS ESPACIALES.

La atmósfera terrestre es fundamental para el desarrollo de la vida y para nuestra supervivencia, pero en el campo de la observación astronómica limita bastante las posibilidades. Primero, por el titileo de las estrellas [8], ya que según su luz atraviesa las distintas capas de la atmósfera, va cambiando de dirección debido a la refracción y a las distintas densidades y temperaturas de cada capa, por lo que nosotros lo percibimos como una

luz que cambia de posición rápidamente. Además, la atmósfera produce una importante distorsión a las imágenes que se conoce como aberración óptica, la cual se consigue solventar más o menos por medio de óptica adaptativa en los telescopios terrestres, pero no llega a acercarse a la resolución que se consigue en el espacio sin atmósfera de por medio. Otro inconveniente de la atmósfera es que absorbe gran parte del espectro electromagnético, por lo que ciertas observaciones en diferentes longitudes de onda del espectro electromagnético como rayos-X, gamma, ultravioleta o infrarrojo se hacen desde el espacio, como se explicará más adelante. También, en función del lugar desde el que se quiera observar entra en juego la variable del tiempo atmosférico, ya que si está nublado o hay viento puede dificultar bastante la observación. Por último, otro gran inconveniente es la contaminación lumínica producida sobre todo por los centros urbanos, viviendas o la iluminación de las carreteras.

Con el avance de nuestra civilización cada vez es más difícil huir de los núcleos urbanos que producen mucha luz en las noches. Para encontrar un buen lugar a veces hay que alejarse mucho a sitios difícilmente accesibles o peligrosos, como montes o llanuras alejadas. Lógicamente, en una órbita tampoco hay esta molestia [9].

1.3.1.-Historia de los observatorios espaciales.

El primer observatorio espacial fue el Cosmos 215, lanzado en 1968 por la Unión Soviética y que tuvo como misión observar la radiación de 36 estrellas diferentes [10]. El más conocido es el Telescopio Espacial Hubble, el cual funciona en la zona del espectro visible y el ultravioleta cercano. Orbita a unos 593 kilómetros de altura y lleva 32 años operativo, desde 1990, y se espera que funcione hasta 2026 aunque no se descarta que se prolongue su vida útil. Con él, se consiguió confirmar la existencia de los agujeros negros, descubrir miles de galaxias, resolver dudas sobre la creación del universo y su edad, observar colisiones de cometas en Júpiter, encontrar evidencias de la existencia de planetas orbitando otras estrellas y muchas otras cosas [11].

Hay muchos otros telescopios espaciales importantes, entre los que destacan el de rayos-X Chandra, el SOHO, Compton o Spitzer. Entre todos se consigue observar objetos tan variados como los cometas del sistema solar, la corona o la composición del Sol, supernovas, nebulosas o agujeros negros. Por último y siguiendo un orden cronológico, el último telescopio espacial lanzado es el James Webb, el que es el mayor telescopio jamás puesto en el espacio y que se espera que sea 100 veces más potente que el Hubble, pudiendo observar a distancias muy lejanas, consiguiendo así imágenes e información de momentos muy cercanos a la creación del universo y de las primeras estrellas y galaxias.



Figura 1.1.- Telescopio Cosmos 215

1.3.2.-Desventajas de los observatorios espaciales.

Como se puede observar, estos observatorios espaciales suponen un gran avance y una gran mina de descubrimientos para la ciencia y la astronomía, pero también presentan grandes inconvenientes debido a su avanzada tecnología y lugar de uso. El coste de un artificio así es muy elevado, ya que la investigación y todo el proceso de desarrollo es muy laborioso, largo y caro, así como el lanzamiento. Por dar unos datos para ilustrar este problema, el lanzamiento con un cohete de tamaño medio puede alcanzar los 250 millones de dólares, mientras que en el

caso de usar un transbordador espacial ese número se puede hasta duplicar. El coste del telescopio Hubble ascendió a unos 2.800 millones de dólares estadounidenses, que teniendo en cuenta la inflación del 2,3% anual desde entonces, sería al cambio unos 5.700 millones de dólares estadounidenses [12]. Otro ejemplo es el del mencionado anteriormente James Webb, cuyo coste fue de unos 10.000 millones de dólares estadounidenses y su desarrollo y construcción duró desde 1996 hasta su lanzamiento el pasado mes de diciembre de 2021. Como se ve, son instrumentos muy complejos que además de un coste elevadísimo, requieren de un trabajo y un desarrollo que se suele alargar durante décadas.

Otro problema que tienen estos aparatos es el mantenimiento. Debido a su localización, orbitando la Tierra u otros cuerpos, es muy difícil arreglar o solventar cualquier problema que surja. De hecho, en este tipo de telescopios solo se ha realizado alguna tarea de mantenimiento en el telescopio Hubble, con un transbordador espacial. Se calcula que, añadiendo el coste de todas las tareas de mantenimiento, su precio total alcanza los 10.000 millones de dólares.

El Hubble pudo ser arreglado ya que orbita relativamente cerca de la superficie terrestre. Otros, como el Chandra mencionado ya, orbita a unos 140.000 km de altura, lo que por poner en contexto es algo más de un tercio del tamaño de la órbita que describe la Luna alrededor de la Tierra [13]. El James Webb lanzado recientemente se sitúa en uno de los puntos de Lagrange Tierra-Sol. Estos puntos son lugares del espacio donde el movimiento orbital de un cuerpo y las fuerzas gravitatorias se compensan entre sí. Con esto se tiene a un cuerpo de masa muy pequeña estable en un punto, como el propio telescopio, en órbita alrededor de dos cuerpos muy masivos como son la Tierra y el Sol. Existen cinco puntos de Lagrange y este telescopio se encuentra en el punto número dos (L2), el cual se encuentra opuesto al Sol y a una distancia aproximada de 1,5 millones de kilómetros [14]. Con estas distancias es prácticamente utópico que alguna misión se acerque a reparar o mantener estos telescopios en caso de que surja algún problema o inconveniente.

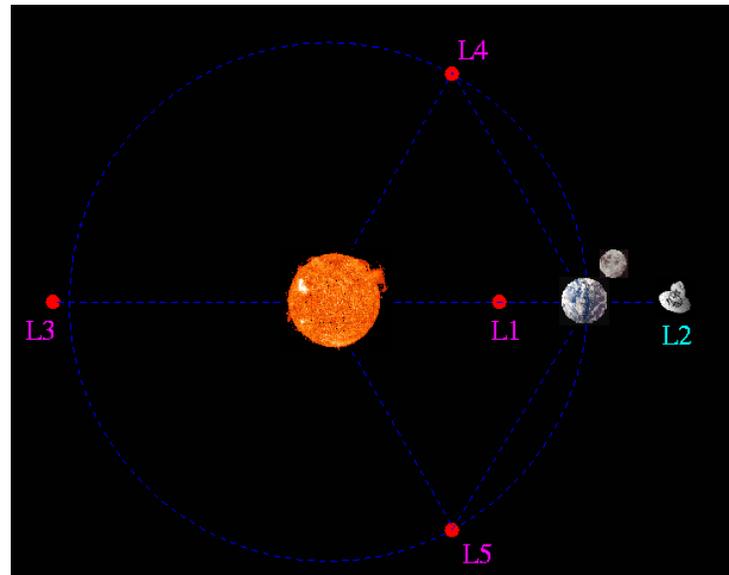


Figura 1.2.- Localización de los puntos de Lagrange

Por último, otro inconveniente que tienen estos telescopios es la limitada vida útil que tienen, ya que en el caso de los telescopios infrarrojos necesitan ser refrigerados con helio líquido y cuando este se termina no se puede llenar de nuevo.



Figura 1.3.- Fotografía del telescopio Hubble

1.4.-ALCANCE.

Una vez repasado para qué puede servir la observación astronómica y la evolución histórica de esta, se tiene que en la actualidad hay, principalmente, dos maneras de realizar observaciones astronómicas. Por una parte, se tienen las observaciones realizadas desde la superficie terrestre, ya sea por grandes observatorios o por telescopios más básicos que puede tener una persona individualmente en su casa o en su organización. Por otra, se encuentran los telescopios situados en una órbita. Como se vio antes, los primeros tienen el inconveniente de la atmósfera y sus consecuencias, y los segundos son extremadamente caros para un proyecto u organización pequeños.

Por lo tanto, con este proyecto se busca llegar a una solución algo intermedia, ser capaces a elevar un telescopio o cámara de un precio más estándar a cierta altitud con un globo meteorológico para evitar gran parte de los problemas de las capas atmosféricas más cercanas a nosotros y poder tomar mejores fotografías y observaciones de las realizadas aquí abajo, así como los costes elevados de un satélite.

1.5.- EXPERIMENTOS CON TELESCOPIOS EN GLOBOS SONDA EN LA ACTUALIDAD.

1.5.1.- Globo con instrumentos científicos por el Pacífico.

En julio de 2016, la NASA llevó a cabo un proyecto en el que se consiguió el vuelo más largo de un globo con instrumentos científicos hasta la fecha, durando la travesía por el océano pacífico 47 días, desde su lanzamiento en Nueva Zelanda hasta su aterrizaje en Perú. Este globo llevaba una carga científica llamada COSI, la cual es un instrumento para estudiar la energía y la distribución de los rayos gamma en el universo. [15]



Figura 1.4.- Preparativos del lanzamiento del globo en Nueva Zelanda

1.5.2.- Proyecto ORISON.

Este proyecto busca crear una infraestructura para la observación en la estratosfera mediante globos meteorológicos. Se busca llevar a cabo distintos proyectos y vuelos preparatorios y divulgativos. El primero de los vuelos fue realizado en Andalucía en abril del año 2016, durante una lluvia de estrellas, buscando grabar con gran calidad y en color diferentes meteoros. [16]

1.5.3.- ASTHROS.

Es un proyecto de la NASA que usará un telescopio de última generación con un tamaño de 2,5 metros. Se estima que será lanzado a finales de 2023 desde la Antártida., pasando tres semanas en las alturas encima de este continente. Este proyecto busca medir la velocidad y el movimiento del gas alrededor de estrellas dentro de nuestra galaxia, en una zona característica por la formación de nuevas estrellas. Además, también cuenta con crear

mapas detallados con el movimiento de ese gas para intentar entender la formación de estos astros. Por último, observará la galaxia Messier 83. [17]



Figura 1.5.- Ilustración de un globo a alta altitud

1.5.-CONDICIONES DEL EXPERIMENTO.

El globo meteorológico [18] o globo sonda [19] es un tipo de globo aerostático que es capaz de elevarse a la estratosfera para elevar instrumentos y obtener información de diferente tipo. Estos globos funcionan mediante un gas liviano, como puede ser el helio o el hidrógeno, ya que tienen menor densidad que el aire y esto hace que eleve el globo. Además, se tiene un elemento llamado tren de vuelo, que se encuentra unido por una cuerda al globo y es donde se encuentran los instrumentos que van a captar los datos e información. En el caso de que se quiera salvar la carga, como es el caso de este trabajo, se tiene un paracaídas para suavizar la bajada. Esto también sirve para tomar datos durante la bajada.

En cuanto a la zona de la atmósfera a la que llega el globo, la estratosfera [20] [21] [22], es una de las capas inferiores de nuestro planeta, el planeta Tierra. Se ubica por encima de la troposfera y por debajo de la mesosfera. En función de la zona del globo terráqueo en el que nos encontremos, esta capa empieza más abajo o más arriba. En las regiones polares, empieza entre los 6 y los 9 km de altura. En cambio, en la región ecuatorial, comienza a los 20 km de altura. La estratosfera acaba a los 50 km de altura. En esta capa se encuentra la capa de ozono y representa el 24% de la masa total de la atmósfera. En cuanto a la temperatura, por la zona inferior, también conocida como tropopausa, la temperatura es bastante homogénea y alcanza un valor promedio de -60°C . Luego, según aumenta la altura comienza a subir la temperatura debido a la capa de ozono, ya que la radiación solar se queda atrapada aquí. Por ello, la temperatura puede llegar a alcanzar los 0°C e incluso algo más.

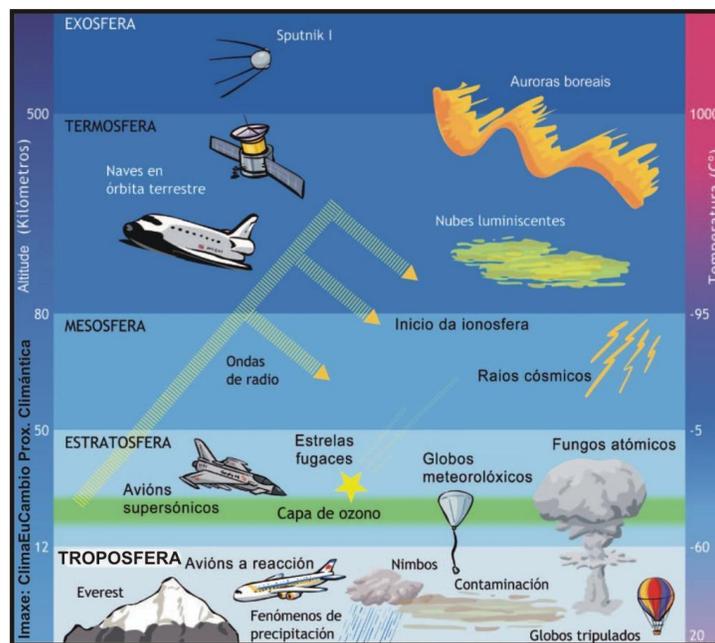


Figura 1.6.- Ilustración de las capas de la atmósfera terrestre

Más adelante se explicará cuál es el procedimiento para lanzar el experimento y cómo las condiciones de la estratosfera hacen que haya que plantearse diferentes opciones para los elementos a incluir en el globo sonda.

1.6.-PASOS PARA EL LANZAMIENTO DE UN GLOBO SONDA.

Para un lanzamiento de esta índole, hay que tener en cuenta muchas variables distintas, teniendo en cuenta las características de nuestro experimento, tanto tamaño, peso, densidad del aislante y duración, entre otras, así como las condiciones a las que se va a ver sometido en la estratosfera. También hay que tener en cuenta los diferentes permisos que hay que pedir a diferentes organizaciones, lo cual lleva unos plazos que alarga el lanzamiento y añade algunas complicaciones al experimento. A continuación, se procederá a explicar todo esto con más detalle.

1.6.1.-Características del experimento.

El globo usado sería uno de látex, relleno de helio. La subida tarda aproximadamente una hora y media y, pasado un tiempo, el globo explota y libera un paracaídas que controla la caída, la cual dura unos treinta minutos.



Figura 1.7.- Vista de la Tierra desde la estratosfera

Se espera llegar a una altura comprendida entre los 30 y 40 km de altura, esto es, la estratosfera. En función del peso, se puede estimar la altura a la que se va a llegar. Hay que

tener cuidado con las condiciones que se pueden encontrar aquí, ya que la temperatura puede llegar hasta los -50°C de temperatura y unos pocos milibares de presión. Con estas temperaturas tan bajas, las baterías aumentan su resistencia interna y disminuye su intensidad. Para luchar contra este problema se pueden usar dos soluciones.

La primera opción es desactivar alguno de los periféricos de la Raspberry, disminuyendo así la intensidad.

La segunda, usar un contenedor de poliestireno expandido, lo que consigue mantener la temperatura del interior a unos -10°C . No obstante, hay que tener cuidado, ya que un contenedor con gran aislamiento tiene mayor densidad, lo que hace que el peso aumente considerablemente. Hay que buscar un equilibrio y una buena relación entre poco peso y un aislamiento notable.

Hay distintos tipos de batería, pero uno interesante es la de tiónilo de litio, la cual no es recargable y tiene un tamaño AAA. Esto es una ventaja, ya que ocupa poco espacio y sobre todo tiene poco peso. Además, hay que valorar el consumo de toda la parte electrónica. Por ejemplo, la batería de Xiaomi, Powerbank, tiene una capacidad de 5000 mAh, lo cual es suficiente ya que aproximadamente la sonda gasta unos 300 mAh por hora de uso, aunque siempre hay que estimar algo más ya que como se comentó anteriormente en situaciones de frío el consumo aumenta. Hay que planificar que la corriente de salida por el puerto USB sea suficiente como para alimentar la Raspberry.

En añadido, hay que tener en cuenta que el sistema de localización tiene que seguir funcionando después del aterrizaje, hasta que se localice. El GPS tiene el problema de que tiene algo de margen de error y limitación de altura. El modelo europeo es más estricto, ya que en el caso de que se supere o una velocidad rápida o cierta altura, deja de funcionar. En este caso obviamente, la altura se vulnera. Sin embargo, el modelo chino es más permisivo, ya que tienen que cumplirse las dos condiciones para desconectarse. Con esto, se consigue trazar todo el recorrido en todo momento. La diferencia entre estos modelos es referida al firmware y a la

forma en la que vienen programada de fábrica dependiendo de dónde se programe o para qué mercado.

En resumen, hay que realizar un estudio económico e intentar minimizar el consumo eléctrico y el peso, sobre todo teniendo cuidado para ello con la batería y la densidad del contenedor.

1.6.2.-Pruebas previas.

Para comprobar todo esto antes de llevar a cabo el experimento, hay que hacer varias pruebas previas.

Primero, hay que probar que el aislamiento térmico funciona. Se mete el conjunto en un arcón para simular la temperatura que se va a encontrar en la estratosfera y viendo si la temperatura en el interior del contenedor se mantiene en unos valores óptimos. Hay empresas que pueden facilitarnos esta tarea con bidones de nitrógeno líquido. Así, podemos encontrar un balance adecuando entre aislamiento buscado y un peso mínimo del contenedor.

Segundo, una prueba de impacto para simular el aterrizaje con el paracaídas. Como aproximación, el impacto sería como tirar la sonda desde un segundo o tercer piso. Para ello, se simula un peso similar en la caja y se prueba a lanzar desde una altura parecida para ver si lo resiste bien.

Por último, una tercera prueba para la electrónica y la duración de la batería. Se pone en marcha el experimento en condiciones atmosféricas normales y se cronometra cuánta autonomía tiene la batería. Después, se repite la prueba en unas condiciones más próximas a la temperatura de la estratosfera para ver cómo afecta esto a la duración de la batería y si con la que tenemos nos alcanzará para el tiempo estimado que se alargará el experimento real.

1.6.3.-Permisos necesarios.

Hay que tener en cuenta que el Enaire (<https://www.enaire.es/home>) es el organismo que legisla el espacio aéreo español. La altura a la que empieza este espacio varía en función del relieve, pero aproximadamente se encuentra a partir de los 100 metros de altura desde la superficie. Superada esta cifra, hay que pedir permisos. En nuestro caso, en Asturias, se piden al aeropuerto militar de la Virgen del Camino. Es necesario dar toda la información posible de nuestro experimento, color, tamaño, aspecto físico, etc. Con ello se emite una nota a todos los pilotos que vayan a volar cerca. El peso limitado es de 4 kilogramos, ya que a partir de esa cifra se requieren más permisos, los cuales llevan más tiempo y retrasan el lanzamiento.

También hay una normativa para la cuerda que sujeta la sonda, ya que tiene que cumplir unas condiciones de tensión para romper a partir de esa cifra.

Cuando comunican la fecha del lanzamiento, no se sabe las condiciones climatológicas que habrá. El mismo día del lanzamiento hay que valorar estas condiciones. En el caso de que sean pésimas, hay que retrasar el lanzamiento y repetir el proceso. Igualmente hay que volver a llamar al aeropuerto militar de la Virgen del Camino para que autoricen el lanzamiento en ese preciso momento. Además, hay que tener un seguro de responsabilidad civil por lo que pueda pasar.

1.6.4.-Lugar de lanzamiento.

Una vez se tenga todo, hay que buscar un buen lugar de lanzamiento. Hay que tener en cuenta que, con las diferentes rachas de viento y movimientos atmosféricos, el experimento puede llegar a aterrizar a unos 100 km de distancia del lugar de lanzamiento. Por lo tanto, en Asturias es difícil por el poco margen existente, teniendo mar al norte y una cordillera montañosa, la cordillera Cantábrica, al sur, pudiendo acabar el experimento en alguno de estos enclaves. Existen simuladores de vuelo que, aunque sea muy difícil predecir todo esto, pueden estimar con el peso total y el lugar de lanzamiento dónde puede acabar. Un ejemplo es el

encontrado en la página <https://predict.habhub.org/>. Se indican las coordenadas de lanzamiento, en este caso, desde el Aulario Sur de la EPI, así como otros parámetros como la fecha, hora, velocidad de ascenso o descenso y altura de la explosión. Cuanto más cercano se realice esta simulación de la fecha del lanzamiento la predicción será más fiable, ya que tendrá en cuenta pronósticos meteorológicos más acertados. Estos parámetros sirven como aproximación para ver cómo funciona este simulador. Como se puede ver en la siguiente imagen, Asturias no sería un buen lugar para el lanzamiento, ya que lanzando el globo desde Gijón acabaría en pleno mar Cantábrico, dificultando bastante su rescate.

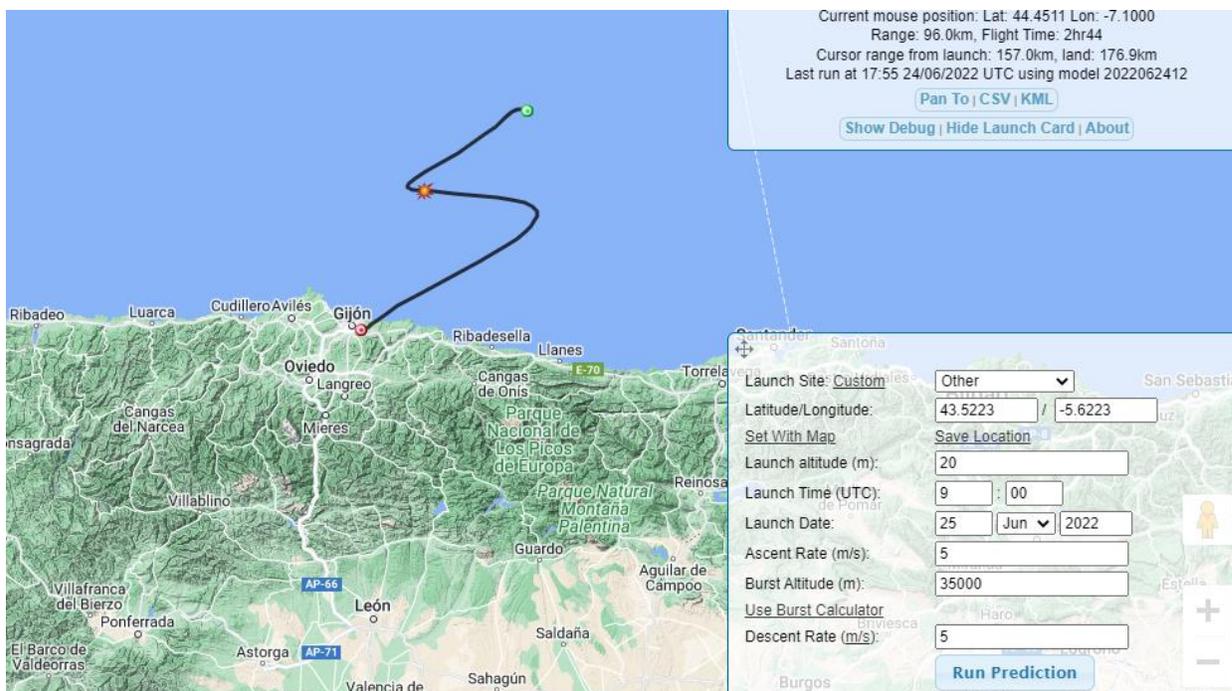


Figura 1.8.- Simulación de vuelo

1.7.-ANÁLISIS DE ALTERNATIVAS.

Para el diseño del proyecto, tanto su parte hardware como software, hubo que tomar varias decisiones para la elección del material, programas o librerías a usar.

En el caso del hardware y como se verá más adelante, la decisión está claramente condicionada por la situación actual geopolítica y la escasez de componentes electrónicos, ya que hace que se encarezca muchísimo o incluso no se disponga de la opción deseada. Por lo tanto, dada la escasez de suministros, la única Raspberry Pi 4 disponible era la proporcionada por los tutores del trabajo con un servo motor conectado. La cámara guía conectada a la Raspberry será una QHY 5-II monocolor, también suministrada por los tutores del trabajo.

En cuanto al software, se optó por realizar el programa en el lenguaje Python. Para el planetario para las pruebas, Stellarium fue la opción elegida ya que es libre y además aparece en el portal del software corporativo de la universidad. En cuanto a la librería usada para el seguimiento de las estrellas, se consideró que OpenCV es la más indicada. Se procedió a instalar la distribución Astroberry en la Raspberry para el manejo de su instrumentación. Para la toma de imágenes se optó por FireCapture, ya que se adapta de forma adecuada a la cámara de la que se dispone.

Uno de los interrogantes que plantea el proyecto es una vez llegado el globo a la altura deseada cómo indicar al programa y la cámara a dónde apuntar. Aunque esta parte no va a ser desarrollada por no tener la posibilidad de probarlo de forma eficiente, se va a plantear alguna alternativa para llevarlo a cabo. Para el reconocimiento de la zona a la que se está apuntando inicialmente, una opción es usar astrometry.net. Esta herramienta permite detectar y reconocer la zona del cielo nocturno mediante una simple fotografía.

Una vez el programa localice dónde se sitúa y hacia dónde apunta, se puede mover el sistema hacia la zona buscada.

Otra opción sería realizar esta tarea mediante un dispositivo móvil que detecte las coordenadas y un sensor IMU, ya que, al contar con magnetómetros y acelerómetros, puede saber hacia dónde se está apuntando.

El sensor IMU también sirve para medir la altura a la que se encuentra el globo, pudiendo controlar con este dato cuándo iniciar el programa en sí.

1.7.1.-Raspberry.

Es un ordenador de formato compacto y coste reducido para hacer llegar la informática a todos los usuarios, permitiendo crear pequeños prototipos de informática y electrónica [23].

Es de software libre y suele utilizar sistemas operativos libres basados en Linux. Se pueden instalar diferentes sistemas operativos, pero el desarrollado para este dispositivo es el Raspberry Pi OS o Raspbian.

El funcionamiento de una Raspberry es igual que el de cualquier ordenador, ya que dispone de los mismos elementos. Tiene varios puertos USB, conector Jack y salida de audio, entre otros. También tiene un lector de tarjetas microSD, en la que se instala el sistema operativo y se encuentran los diferentes archivos.

En este caso, el modelo Pi 4, posee un conector Gigabit Ethernet y una tarjeta con WiFi y Bluetooth.

Desde el instalador del operativo, Raspberry Pi Imager, se puede habilitar el uso de SSH por parte de la Raspberry, así como pasarle los datos de la red WiFi a la que está conectado el ordenador desde el que se realiza la instalación. También permite configurar el usuario y la contraseña de la Raspberry, así como la dirección del dispositivo, que por defecto es raspberrypi.local.

Una vez configurada, se puede acceder a la terminal del dispositivo mediante un cliente SSH. En mi caso, utilicé PuTTY. Para poder transferir archivos desde el ordenador a la Raspberry o viceversa se usa un cliente SFTP que use SSH como puede ser WinSCP.

El servomotor es un pequeño motor que va instalado en una plataforma encima de la Raspberry y mueve un soporte en los dos ejes, horizontal y vertical, unos 180 grados en ambas direcciones.



Figura 1.9.- Fotografía de la Raspberry Pi 4 y el servomotor utilizados.

1.7.2.-Cámara QHY 5-II monocolor.

La cámara usada es la QHY 5-II monocolor [24]. Las especificaciones de esta son importantes como se verá más adelante en la explicación del propio código. Tiene una resolución útil de 1280 x 960 píxeles activos, un tamaño de píxel de 3.75 micras, una distancia focal de 10,4 milímetros y un tiempo de exposición que se puede variar entre 20 microsegundos y 10 minutos. Además, se conecta mediante una interfaz USB 2.0.



Figura 1.10.- Fotografía de la cámara QHY 5-II usada

1.7.3.-Astroberry.

Es un sistema para controlar los equipos de astronomía conectados a la Raspberry, tales como un telescopio o cámara. Es un software de código abierto y funciona con Linux. Se puede acceder de forma remota desde un navegador web o un cliente SSH como si de un escritorio remoto se tratase. Permite una gran personalización en todas las partes del sistema. Se puede acceder a planetarios como el propio Stellarium para controlar las cámaras y apuntar a las partes deseadas del cielo nocturno.

1.7.4.-Stellarium.

Es un planetario de código abierto que muestra el cielo tal cual se ve en la vida real a simple vista o con un telescopio. Muestra la posición de estrellas, planetas, galaxias, constelaciones, satélites y muchos otros elementos del cielo. Además, simula ciertos fenómenos astronómicos tales como eclipses o lluvias de meteoros. También permite fijar la ubicación, pudiendo mostrar el cielo según se ve en diferentes partes del globo terráqueo. Otra gran utilidad que tiene es la posibilidad de fijar una fecha y hora, pudiendo simular cómo se

vio o verá el cielo en cualquier instante de tiempo. Permite detener la reproducción en tiempo real del cielo o acelerarlo para que vaya más rápido [25].



Figura 1.11.- Logo del programa Stellarium



Figura 1.12.- Vista del programa Stellarium

1.7.5.-OpenCV.

Es una librería destinada para la visión artificial e inicialmente fue desarrollada por Intel [26] [27] [28]. Es ampliamente usada para el análisis y tratamiento de imágenes mediante diferentes algoritmos. Por mencionar alguno de los usos más importantes, OpenCV es usado para la detección de rostros, de gestos, rastreo de objetos, interacción persona-computadora, estereoscopía o realidad aumentada.

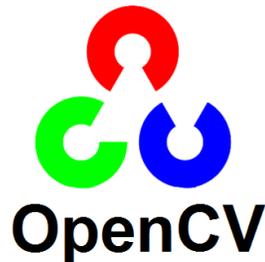


Figura 1.13.- Logo de OpenCV

1.7.6.-FireCapture.

Es un programa gratuito específico para la astronomía que sirve para obtener vídeos e imágenes. Tiene diferentes herramientas para la captura de diferentes astros, como planetas, el Sol y la Luna, así como una gran compatibilidad con gran diversidad de cámaras especialmente dedicadas a la astronomía. Este programa permite cambiar el tiempo de exposición y la cantidad de luz absorbida, entre otros tantos parámetros [29].



Figura 1.14.- Logo de FireCapture

1.7.7.- Astrometry.net.

Es una plataforma que posee una especie de mapa astronómico formado por una gran cantidad de fotografías enviadas, permitiendo así reconocer y detectar rápidamente en qué coordenadas se sitúa cualquier fotografía que se tenga y no se sepa de dónde es. Tiene sobre todo dos formas útiles para esta detección.

Primero, un servicio web [30] al que se sube la fotografía en cuestión.

Segundo, un programa descargable que lleva consigo unos índices. Estos son unos archivos que contienen puntos de referencia del cielo nocturno para ayudar al identificador de imágenes a realizar su tarea. Hay diferentes índices en función del tamaño en minutos de arco de nuestra imagen.



Figura 1.15.- Logo de Astrometry.net

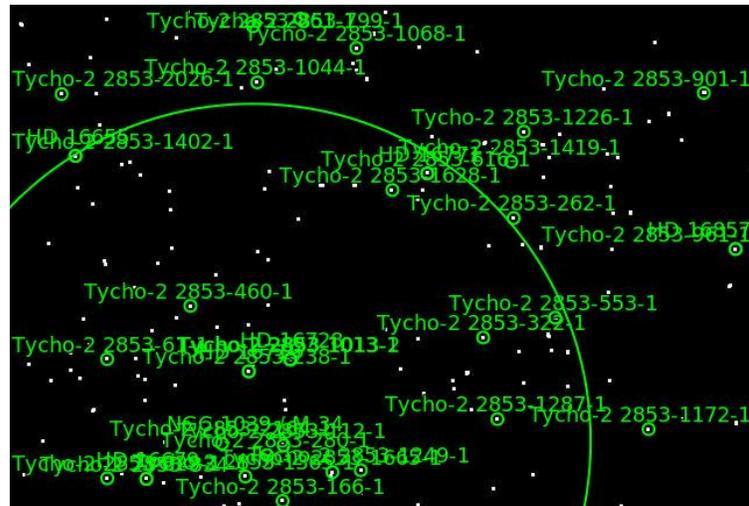


Figura 1.16.- Funcionamiento de la herramienta

1.7.8.- Sensor IMU.

Una Unidad de Medición Inercial o IMU [31] es un dispositivo electrónico que mediante el uso de giroscopios y acelerómetros mide la aceleración y la rotación. Uno de sus múltiples usos es en los drones, pudiendo calcular la altitud o su posición. En el caso de este proyecto, otra vez la situación de los componentes limita esta parte. Solo se dispone de un IMU con sensor en dos ejes, por lo que solo se puede saber su movimiento en el eje horizontal y vertical. Sin embargo, puede ser útil para dos aspectos. Primero, para la propia orientación a la que se acaba de hacer referencia y, segundo, para saber en qué momento se ha llegado a la altura deseada. La IMU sabe en todo momento la aceleración, en este caso, vertical, a la que está sometido el globo. En una situación de supuesta estabilidad en la altura se puede calcular cuál es la aceleración esperada y, una vez llegados a ese valor, iniciar el programa.



Figura 1.17.- Imagen de un sensor IMU

2. Objetivos concretos y relación con el estado actual.

Planteada esa hipótesis de partida, el objetivo final será montar un sistema en un globo de alta altitud para obtener mejores imágenes que en tierra firme y abaratar costes con respecto a las cifras mareantes de un satélite.

Por motivos de tiempo, tanto como por no poder hacer una cantidad abundante de pruebas debido a la meteorología que hubo en el Principado de Asturias estos últimos meses de primavera, como por la dificultad que entraña el diseño y los permisos necesarios para el lanzamiento del globo, la extensión de este trabajo consistirá en investigar sobre las posibilidades que ofrece el desarrollo de un prototipo terrestre que mantenga fija una cámara apuntando al cielo nocturno, moviéndose en función del avance del cielo nocturno u otros movimientos del propio sistema.

En último caso el sistema consistirá en dos cámaras unidas a una Raspberry Pi 4. La primera cámara serviría como guía capturando las imágenes para rastrear si se ha producido algún movimiento del sistema con respecto al cielo a observar, ya sea debido al viento u otra alteración de la posición del globo, o a la propia rotación terrestre y avance del cielo nocturno. Para ello, se implementa un programa en Python usando la librería OpenCV, el cual se encarga de trazar unos puntos concretos, en este caso, estrellas, en los frames capturados en tiempo real de la cámara guía. El programa sigue esos puntos y calcula los píxeles y el ángulo que se movió cada punto para transmitir el grado de movimiento necesario para seguir apuntando al mismo sitio con la Raspberry. Como cámara guía se utilizará la QHY 5-II mencionada previamente.

La segunda cámara sería la que toma las imágenes de ciencia que se van a analizar posteriormente. Al ir conectada a todo el mecanismo, se mantendría estable con respecto al objetivo a observar, sirviendo esto, por ejemplo, para tomar fotografías de larga exposición.

Este prototipo es una primera aproximación de un proyecto a más largo plazo y que cuenta con diversas limitaciones impuestas por la situación geopolítica actual. La escasez de componentes electrónicos, tales como procesadores, hace que las diferentes alternativas se hayan encarecido bastante o directamente no haya. Por ejemplo, las Raspberry han triplicado o cuadruplicado el precio. Por tanto, ante esta situación, se optó por usar los componentes con los que ya se contaba, en este caso, la Raspberry Pi 4. Más adelante se explicará otras alternativas o mejoras posibles para este sistema mencionado.

3. Metodología de trabajo.

La metodología de trabajo usada para este experimento ha sido ir realizando diferentes pruebas para ver que el programa se iba comportando de la forma esperada. Se comenzó con el diseño del programa, el cual obviamente se fue mejorando y perfeccionando con el paso de las diferentes pruebas y necesidades ocurridas. Para empezar, se programó un código capaz de seguir un punto simple. Para ello la primera prueba consistió en tener en un papel un punto pintado con un boli y moverlo manualmente para ver que el programa sigue el punto.

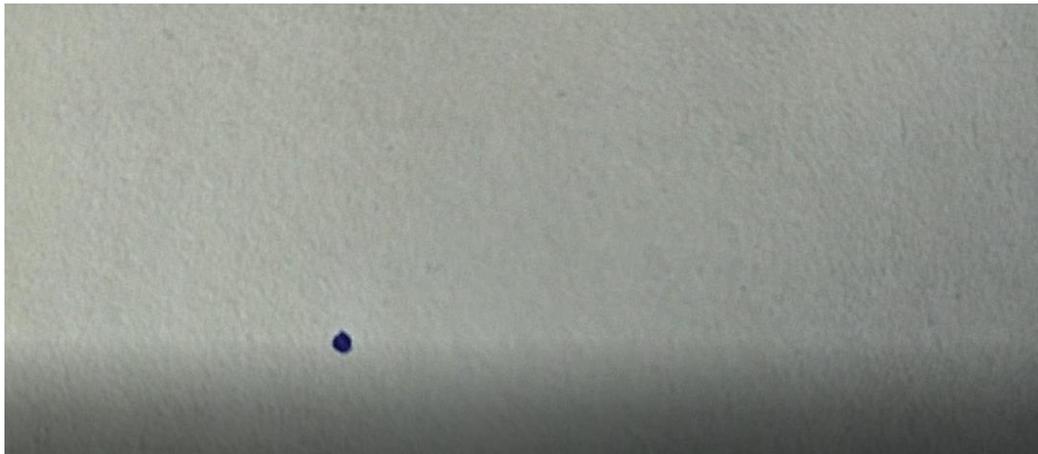


Figura 3.1.- Punto en una hoja de papel

Para ir subiendo el grado de realismo de las pruebas, utilicé el programa Stellarium. Como se mencionó anteriormente, la posibilidad de acelerar las grabaciones del cielo resulta muy útil ya que, aunque en el entorno real la velocidad de movimiento del cielo será la normal, la que se puede apreciar con el ojo cualquier noche, para hacer pruebas y tener un movimiento considerable y trazable haría falta capturar un vídeo muy largo y pesado. Además, el movimiento acelerado sirve para simular movimientos rápidos del globo provocados por el viento o rotaciones por inercia.

Pasando ya a una prueba real, se realizaron pruebas y grabaciones con una cámara en un cielo nocturno. La cámara usada es la QHY 5-II monocolor.

Estos vídeos fueron tomados con el programa FireCapture, probando diferentes configuraciones de enfoque y tiempos de exposición para conseguir las imágenes y vídeos deseadas.

Primero, se tomaron vídeos desde el centro de Gijón. Debido a la contaminación lumínica y a que esa noche había luna llena, los vídeos no fueron de la calidad esperada. Solo se ven unas pocas estrellas tenues.

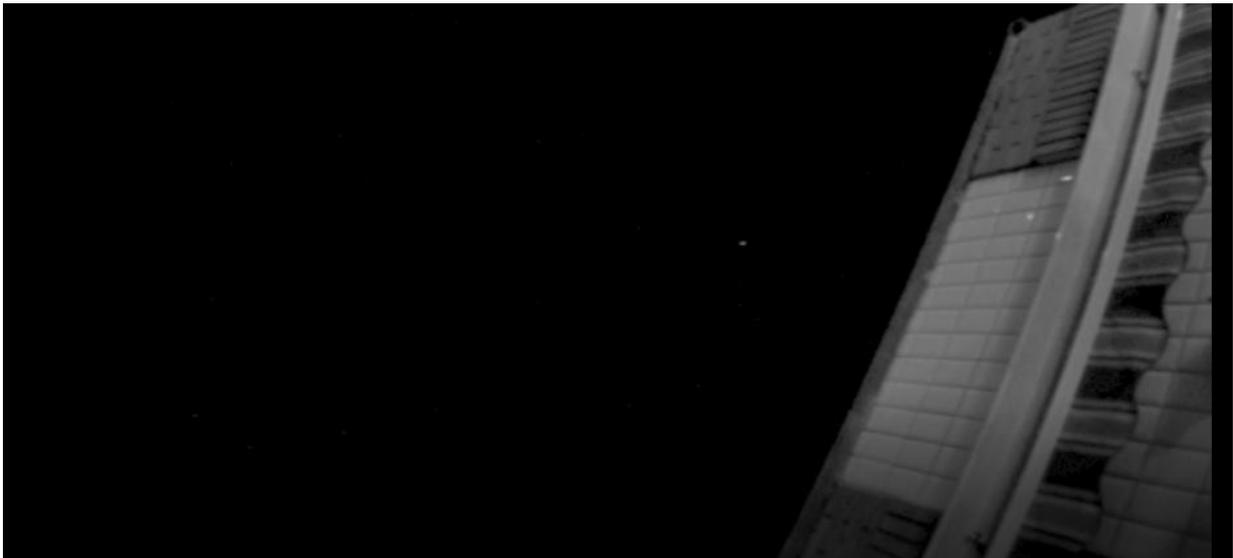


Figura 3.2.- Captura de uno de los vídeos obtenidos desde Gijón

Hubo que esperar varias semanas a poder repetir mejores pruebas, ya que el tiempo atmosférico no acompañó nada, con noches muy nubladas. Para poder seguir realizando pruebas, a la espera de que el tiempo mejorase, se optó por buscar algún vídeo en la red que pudiese ser de ayuda. Se usó un vídeo subido a la plataforma Vimeo [32] en el que se muestra una grabación con una cámara de ojo de pez con una focal de 2.7mm. La siguiente noche despejada, a finales de junio, se hicieron pruebas a unos kilómetros de Nava, alejado de los núcleos urbanos que, aunque no son muy grandes, producen suficiente luz como para inducir

contaminación lumínica muy molesta. Además, esa noche no había Luna (estaba casi nueva y ya se había ocultado), por lo que la visibilidad era inmejorable. Se sujetó la cámara con un palo a la tierra para mantenerla estable y se grabaron varios vídeos, con tiempos de exposición de 1 y 2 segundos y moviendo suavemente el palo con la cámara para simular los movimientos del globo.



Figura 3.3.- Montaje para la cámara

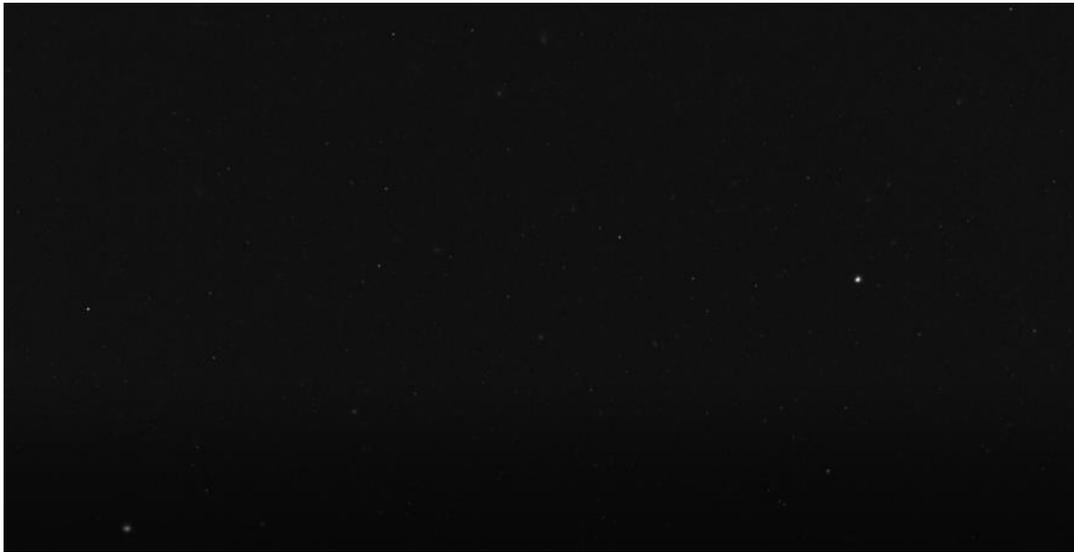


Figura 3.4.- Captura de uno de los vídeos obtenidos desde Nava

Los siguientes pasos fueron ya en la Raspberry. Tras instalar el Astroberry mencionado, se detectó un problema de incompatibilidad en la instalación de la cámara, por lo que se decidió descartar Astroberry e instalar el SO por defecto, Raspbian.

Para empezar, se hicieron varias pruebas con el servomotor para entender su movimiento y cómo se puede controlar. Estos resultados son analizados más adelante en el anexo. Después, se probó a ejecutar el programa creado dentro de la Raspberry sin ninguna cámara conectada, tomando como referencia uno de los vídeos grabados y el servomotor se movía adecuadamente.

Por último, la prueba final consiste en conectar la cámara a la Raspberry para realizar la tarea en tiempo real y transmitiendo ese movimiento al servo de la Raspberry.

Tras descartar la opción de usar Astroberry por problemas de compatibilidad y usar en su defecto Raspbian, se consiguió usar la cámara en la Raspberry. Para ello, hubo que instalar un SDK, en este caso se usó la versión 22.03.11, descargada desde el propio sitio web de la cámara (https://www.qhyccd.com/html/prepub/log_en.html#!log_en.md). Para comprobar que todo funciona de manera satisfactoria, en la propia página hay un par de ejemplos en varios

lenguajes, entre ellos, Python. Esta prueba inicializa la cámara y captura algunos frames, guardándolos tanto en formato .bmp, pudiendo abrirse con un editor de fotos, como transformando la imagen a caracteres ASCII y mostrando esa imagen transformada por consola. La siguiente fotografía es un ejemplo tomado con el programa de prueba mencionado. Desde este programa se puede ajustar el tiempo de exposición y el brillo, entre otros parámetros.



Figura 3.5.- Fotografía tomada mediante el programa de prueba de la cámara QHY 5-II

Y esta sería la transcripción que hace el programa de la anterior fotografía a código ASCII.

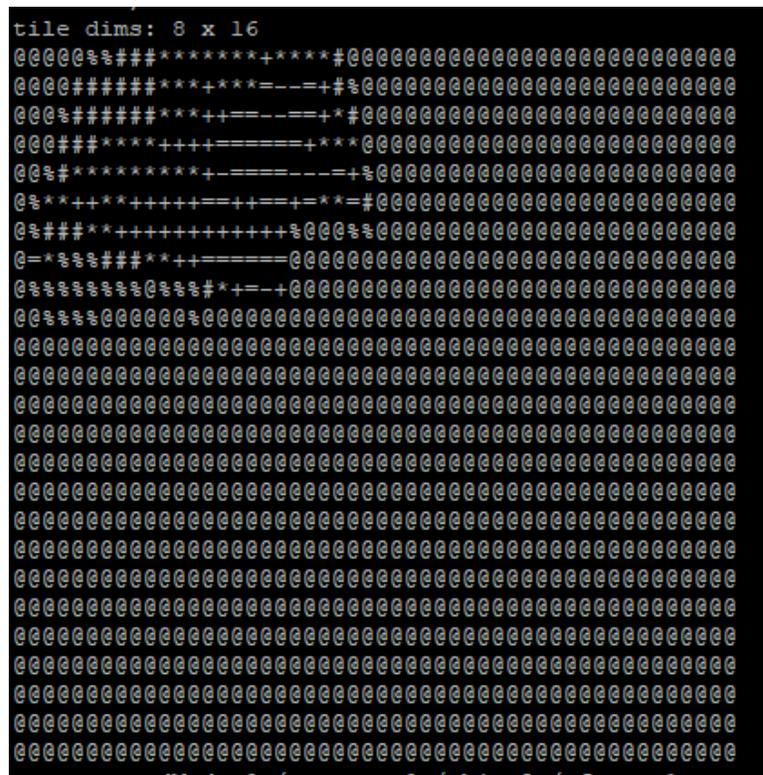


Figura 3.6.- Resultado ASCII de la fotografía tomada

Sin embargo, a la hora de juntar este código de prueba, con el que se ve que se toman las imágenes, y el código ya existente de rastreo de estrellas, se producen algunos errores. Tras investigar, otros usuarios reportaron problemas similares de compatibilidad entre esta cámara en concreto y la librería OpenCV. No se encontró una solución sencilla, ya que esta cámara necesita su SDK propio.

3.1.- DESCRIPCIÓN DE LOS VIDEOS TOMADOS

A continuación, se hará una breve descripción de los distintos vídeos tomados como prueba:

- Prueba inicial: es el archivo prueba_inicial.mp4, un vídeo grabado con mi propio dispositivo móvil moviendo una hoja con un punto de boli pintado en ella.

- Stellarium: los vídeos tomados mediante el planetario Stellarium.
 - 45_grados.mp4: es un vídeo tomando como campo de visión un campo cuadrado de 45°, siendo el único movimiento apreciable el de las propias estrellas moviéndose.
 - 45_poco_movimiento.mp4: es un vídeo similar al anterior, pero añadiendo cierto movimiento extra simulando el balanceo o movimiento del globo
 - 45_mucho_movimiento.mp4 y 45_mucho_movimiento_2.mp4: son dos vídeos similares a los anteriores, pero añadiendo movimientos más bruscos para ver cómo se comporta el programa
 - prueba_horizontal.mp4: es un vídeo horizontal que, como el de 45_grados.mp4, solo tiene como movimiento apreciable el propio de las estrellas.
- Pruebas con la cámara QHY 5-II: todos los vídeos tienen una resolución de 640 x 480 píxeles.
 - Desde Gijón: son los vídeos que se llaman Sistema_pruebas, desde el 1 al 6, en formato .avi, que es el que graba la cámara. En estos vídeos no solo se ve el cielo si no algunos elementos más, por lo que el programa aquí también detecta puntos fuera del cielo. Todos estos vídeos son una consecuencia de imágenes de un segundo de exposición cada una.
 - Desde Nava: son vídeos en los que únicamente se ve el cielo nocturno.
 - 1s_movimiento.avi: es un vídeo que tiene un tiempo de exposición de un segundo y tiene un pequeño movimiento añadido aparte del propio de las estrellas.
 - 1s_mucho_movimiento.avi: igual que el anterior, pero con movimientos más bruscos.
 - 2s_movimiento.avi: este vídeo tiene un tiempo de exposición de dos segundos y también presenta un movimiento añadido.
 - video_largo.avi: son tres vídeos con ese nombre y numerados hasta el 3. Son grabaciones algo largas sin movimiento añadido. Al no estar acelerado como las pruebas del Stellarium, apenas hay cambios en todo

el vídeo, pero sirve para hacerse una idea de cómo puede ser una grabación real.

- Prueba tomada de internet: timelapse.mp4, como ya se especificó anteriormente, es un vídeo descargado de Vimeo en el que se ve un cielo estrellado y bastante acelerado su movimiento. Se recortó el vídeo descargado originalmente para dejar únicamente el cielo nocturno. No se incluye en la entrega ya que es un archivo muy pesado.

4. Trabajo realizado y resultados obtenidos.

Por lo tanto, se ha conseguido desarrollar un programa desarrollado en lenguaje Python con la librería OpenCV que es multiplataforma. Este programa detecta unos puntos al inicio de un vídeo o grabación y los va rastreando en sucesivos frames, calculando la diferencia entre los mismos. Esa diferencia resulta ser un ángulo y se transmite al servomotor de la Raspberry para moverse y compensar ese desplazamiento.

Mediante las diferentes pruebas, se fue perfeccionando el programa de rastreo. Primero como se indicó, hubo una primera prueba con un punto en un papel. Al mover el papel, se simuló una “estrella gorda” y se comprobó que el programa cumplía su función. Después, se probó con una grabación creada en el Stellarium. Inicialmente, con un vídeo estático en el que simplemente se capta el movimiento natural del cielo nocturno (algo acelerado, como se explicó anteriormente). Lógicamente, en un globo de altitud hay un balanceo producido por el estado de suspensión, así como del viento. El programa debe estar preparado también para estos inconvenientes. Para ello, el siguiente vídeo grabado se le introdujo algo de movimiento. Con las teclas dentro del programa se desplaza el campo de visión por el planetario virtual, así, con pequeños movimientos se puede simular ese balanceo.

Una vez se comprobó que para estos vídeos simulados el programa funcionaba, se pasó a pruebas más realistas con la cámara mencionada anteriormente. Hubo bastantes problemas y se hicieron menos pruebas de las esperadas, sobre todo debido al tiempo atmosférico que hubo en Asturias durante todos estos meses, ya que apenas hubo noches despejadas con buenas condiciones para tomar vídeos útiles. Se tomaron varios vídeos desde Gijón, así como algún vídeo encontrado en la red para el que se usó la misma cámara. Los resultados tras probar el programa con estos vídeos es que funciona correctamente, rastreando bien las estrellas. El mayor inconveniente encontrado fue al pasar alguna nube o avión, que ahí sí que el programa

fallaba, perdiendo la referencia a las estrellas o confundiendo a los puntos detectados con la distinta velocidad de esos objetos, pero se considera un problema menor, ya que a la altura en la que se va a realizar este experimento no existen esos obstáculos.

Posteriormente y como se indicó en el apartado anterior, se hicieron nuevas pruebas en una noche con mejores condiciones meteorológicas desde los alrededores de Nava, obteniendo unos resultados positivos.

Además, se llevó a cabo la instalación de Astroberry en la Raspberry. Para ello, hizo falta el utilitario balenaEtcher [33], el cual sirve para escribir archivos .iso entre otros para tarjetas SD o dispositivos USB. Para poder configurar y usar la Raspberry desde el ordenador, hubo que conectarse a la propia conexión Wi-Fi de la Raspberry. Una vez realizado esto, permite acceder al dispositivo desde el navegador como si de un escritorio remoto se tratase. Sin embargo, al estar conectado a su red Wi-Fi, el ordenador se desconecta de la red anterior. Para solucionar esto, desde el escritorio de la Raspberry se conecta a la red Wi-Fi disponible, como la de cualquier hogar. Así, pasa a ser un dispositivo más al que el router le asigna una dirección IP dentro de la red mediante DHCP y permite conectarse simplemente desde el navegador poniendo la IP en la barra de búsqueda.

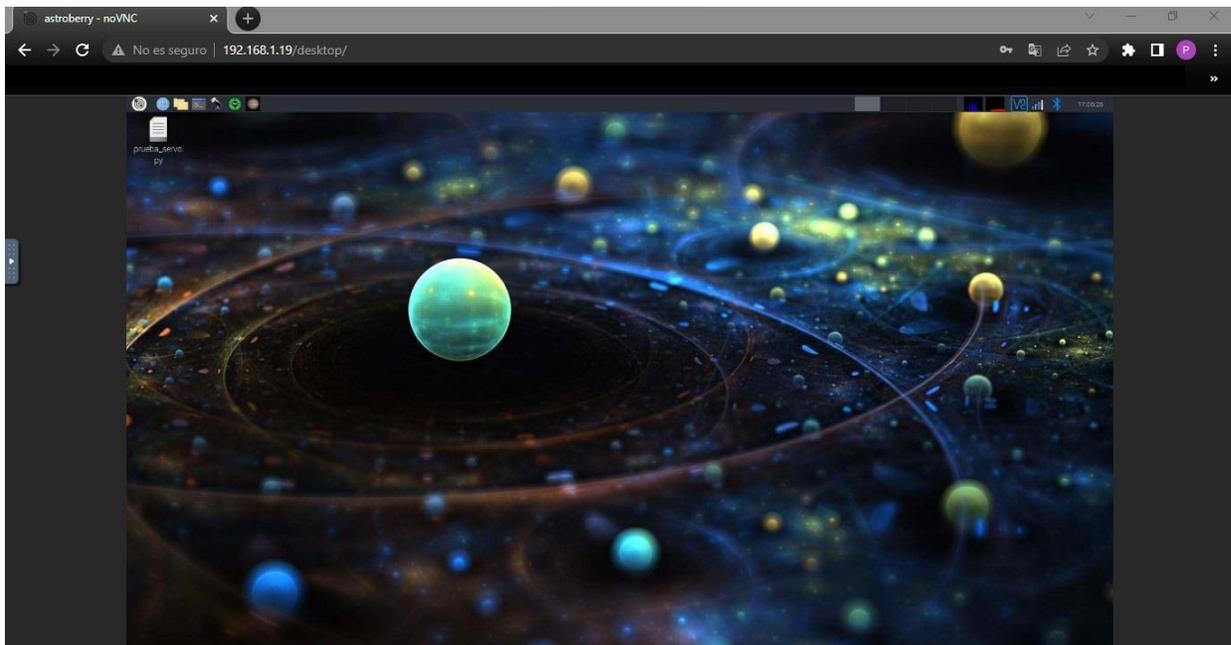


Figura 4.1.- Vista del escritorio del Astroberry

Sin embargo, como se mencionó en el apartado anterior, esta distribución Astroberry resultó no ser compatible con la cámara usada, por lo que se procedió a instalar el sistema operativo por defecto de Raspberry.

Se investigó el movimiento del servomotor y cómo poder controlarlo. Además, se indagó en el funcionamiento de la cámara de la que se disponía. Se consiguieron tomar fotografías desde la propia Raspberry tras instalar el SDK propio de la cámara y con un programa de prueba de su propia página. Pero, por desgracia, este SDK tiene incompatibilidades con OpenCV.

La gran mayoría de cámaras al conectarlas a la Raspberry se configuran de forma automática, sin instalar un SDK. Por ello, el proyecto queda limitado por una cámara que causa problemas con Linux.

4.1.-HARDWARE.

Este proyecto ha sido realizado en un ordenador personal con las siguientes características:

- **CPU:** Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz
- **RAM instalada:** 8,00 GB
- **Sistema operativo:** Windows 11 Home

La Raspberry usada tiene estas características [34]:

- **Procesador:** ARM Cortex-A72 1,5 GHz
- **GPU:** VideoCore VI
- **RAM instalada:** 2,00 GB
- **Conectividad:** WiFi y Bluetooth

La cámara usada es una QHY 5-II monocolor con las siguientes características:

- **Resolución:**
 - 30 FPS a 1280 x 960
 - 44 FPS a 1024 x 768
 - 75 FPS a 800 x 600
 - 106 FPS a 640 x 480
 - 200 FPS a 320 x 240
- **Tamaño de píxel:** 3.75 micras
- **Distancia focal:** 10,4 milímetros
- **Tiempo de exposición:** desde los 20 microsegundos a los 10 minutos
- **Conectividad:** USB 2.0
- **Gama de colores:** blanco y negro

4.2.-SOFTWARE.

Para el desarrollo del programa se ha utilizado el lenguaje Python en su versión 3.9, con el IDE PyCharm 2021.2.3. Se han usado estas librerías:

- **Numpy:** librería diseñada para el cálculo numérico.
- **OpenCV (cv2):** librería desarrollada para visión artificial. Realiza un tratamiento de las imágenes con diferentes algoritmos.
- **Pantihat:** librería para controlar el servomotor de la Raspberry de forma independiente en sus dos ejes.

Para el programa de prueba localizado en la página web de la cámara, se usaron estas librerías para la toma de fotografías:

- **Os:** es una librería para acceder a distintas funciones del sistema operativo, en particular las referentes a los directorios y manipular, leer y escribir archivos.
- **Time:** es una librería para funciones relativas al tiempo, como obtener la fecha actual.
- **PIL:** es una librería diseñada para manipular y guardar formatos de archivo imágenes.
- **Astropy:** es una librería desarrollada para la investigación astronómica.
- **Datetime:** otra librería para manipular medidas de tiempo.
- **Qcam:** la propia librería de la cámara con sus funciones.

Otros programas usados:

- **Astroberry:** sistema para controlar diferentes instrumentos astronómicos, tanto como los propios telescopios como algunos planetarios como Stellarium.
- **FireCapture:** programa para obtener vídeos e imágenes y orientado a la observación astronómica.
- **Stellarium:** planetario de código abierto que permite buscar diferentes cuerpos, en cualquier localización y momento.

- **Astrometry.net:** herramienta que permite localizar a qué zona del cielo nocturno pertenece una fotografía.

5. Conclusiones y trabajo futuro

5.1.- CONCLUSIONES.

Echando la vista atrás y viendo las condiciones con las que se comenzó este proyecto y las diferentes dificultades a las que me fui enfrentando en todo este tiempo, creo firmemente que la experiencia y el trabajo realizado ha sido muy positivo.

Ha sido muy enriquecedor, ya que hasta la fecha no había realizado un proyecto de esta envergadura, tanto por dificultad y por duración como por combinar diferentes disciplinas como pueden ser la astronomía y la informática.

Elegí este trabajo porque lo vi como una oportunidad de juntar el conocimiento adquirido durante estos años estudiando este grado con una de mis aficiones fuera de los estudios como es la astronomía.

Desde pequeño siempre me llamó la atención todo lo relacionado con nuestro planeta y lo que sucede fuera de él. Tanto las estrellas, los planetas, las galaxias, así como los diferentes eventos astronómicos y meteorológicos que tienen lugar habitualmente. En mis ratos libres, suelo estar pendiente de lo que sucede, tanto para observarlo con mi telescopio de aficionado, como incluso fotografiar lo que se puede con mi teléfono móvil. Por lo tanto, en cuanto vi la posibilidad de llevar a cabo este proyecto no dudé ni un instante en que debía realizarlo.

Además, a lo largo del proyecto he trabajado con diferentes componentes o programas a los que no me había enfrentado en mi vida. Por poner un ejemplo, hasta ahora nunca había usado una Raspberry y desconocía sus posibles utilidades. Ha sido interesante entender y usar componentes así. Por otra parte, el uso de OpenCV ha sido enriquecedor para tener una visión de cómo se puede lograr tener un programa que rastree o reconozca ciertos objetos.

También estuvo interesante el momento de realizar las pruebas, ya que como dije anteriormente muchas veces las realizaba ya en mis ratos libres de como afición. Con esto, conseguí entender y aprender ciertos conceptos que desconocía acerca de la materia.

Quiero añadir también que, aunque no se llegó a utilizar e investigar más en detalle por diferentes problemas la distribución Astroberry, sí que llegué a entender el potencial que puede tener de cara a controlar la diferente instrumentación y cargar planetarios para orientarlos de forma rápida y eficiente.

Considero todos estos conocimientos adquiridos importantes, ya que me da pie a poder continuar o realizar algún proyecto similar en un futuro, tanto en mis ratos libres por pura afición como, por qué no, como una salida profesional.

En cuanto al resultado obtenido, fue un lastre la imposibilidad de realizar más y mejores pruebas por el tiempo atmosférico. Durante el invierno sí hubo noches despejadas, pero ni se disponía aún de la cámara utilizada y la temperatura era extremadamente baja. Cuando llegó la primavera y las buenas temperaturas no hubo apenas noches sin nubosidad.

Aun así, se consiguieron pruebas suficientes como para comprobar que el programa funcionaba de forma correcta, rastreando de forma eficaz las estrellas, calculando el desplazamiento entre frames y trasmitiéndolo al servomotor de la Raspberry.

5.2.- TRABAJO FUTURO.

Como se explicó anteriormente, la situación geopolítica actual, con la escasez de componentes que existe en todo el mundo, limitó bastante las posibilidades del hardware a usar. Un ejemplo es el servomotor y su poca precisión en el movimiento ya que solo permite moverse de grado en grado. En un movimiento normal del cielo pasa una cantidad de tiempo considerable hasta que se llega a mover un grado completo, por lo que es algo brusco estar sin

ningún movimiento durante algunos minutos y de golpe moverse un grado en alguno de los dos ejes. Sería recomendable usar algún motor o elemento que tenga una precisión mayor en el giro. Algo similar ocurre con el sensor IMU, ya que del que se dispone solo cuenta con dos ejes, mientras que existen en el mercado algunos modelos con tres ejes.

Como se indicó anteriormente, estos sensores son usados para calcular la aceleración a la que está sometido el globo y por tanto poder estimar la altura. Aquí me gustaría hacer mención de que existe otro TFG relativo a este tema.

También está el problema de la compatibilidad entre la cámara y OpenCV. Lo ideal sería haber usado una cámara que hubiese facilitado más esta tarea, pero fue imposible con las limitaciones existentes.

A. Anexo 1: OpenCV y software desarrollado.

A.1.-ALTERNATIVA DE OPENCV.

Como ya se ha explicado, para este programa se usará la librería OpenCV. Esta contiene infinidad de algoritmos y posibilidades. Tras una investigación previa, se encontraron varias opciones que podrían funcionar para el objetivo buscado.

A.1.1.- Clasificador de cascada.

Mediante este sistema se pueden detectar objetos gracias al aprendizaje automático. Se entrena la función con imágenes positivas y negativas, esto es, que contienen o no el objeto a detectar. Al entrenar el clasificador, la salida es un 1 si hay una alta probabilidad de que se muestre el objeto y un 0 si no. Se puede usar para objetos más complejos que las estrellas, como coches o normalmente rostros de personas. [35] [36]

A.1.2.- Contornos en imágenes

Los contornos son las líneas que unen los puntos que conforman una figura. OpenCV tiene una función llamada findContours la cual detecta los contornos que forman una imagen. Esto podría ser usado para detectar el contorno de las diferentes estrellas a detectar, pero normalmente esto es usado en objetos más grandes que las estrellas en las capturas o vídeos [37] [38].

A.1.3.- Optical Flow.

Detecta un patrón de movimiento entre dos frames seguidos, tanto si se produce por el movimiento de la cámara o del propio objeto a observar. Para ello, asume varias premisas. Primero, que la intensidad de los píxeles del objeto no cambia entre frames consecutivos y segundo, que los píxeles de su alrededor tienen un movimiento similar. Es por esto por lo que es usado en estabilización de vídeo entre otros usos, ya que permite estimar la distancia que se desplaza el objeto [39] [40].

A.1.4.- Opción escogida.

Vistas las diferentes opciones, la que mejor se adapta a este trabajo es Optical Flow, ya que sirve para objetos simples como son las estrellas, en las que los píxeles y objetos a su alrededor, tanto el propio vacío como las otras estrellas, tienen un movimiento similar.

A.2.-EXPLICACIÓN DE LOS EJES DE COORDENADAS DE OPENCV Y MOVIMIENTO DEL SERVO.

En OpenCV, los puntos están formados por sus coordenadas en el eje X y en el eje Y. Estas coordenadas representan los píxeles del frame. Sin embargo, no siguen el sistema cartesiano, en el que el eje X aumenta hacia la derecha y el Y hacia arriba. En este caso el eje horizontal también aumenta según nos desplazamos hacia la derecha, pero el eje vertical aumenta al bajar en la imagen. Esto hace que, ante un movimiento normal de las estrellas, esto es, hacia arriba y a la derecha según lo vemos nosotros como observadores como se ve en la siguiente imagen, las coordenadas del punto aumentan en el eje X, pero disminuyen en el eje Y.

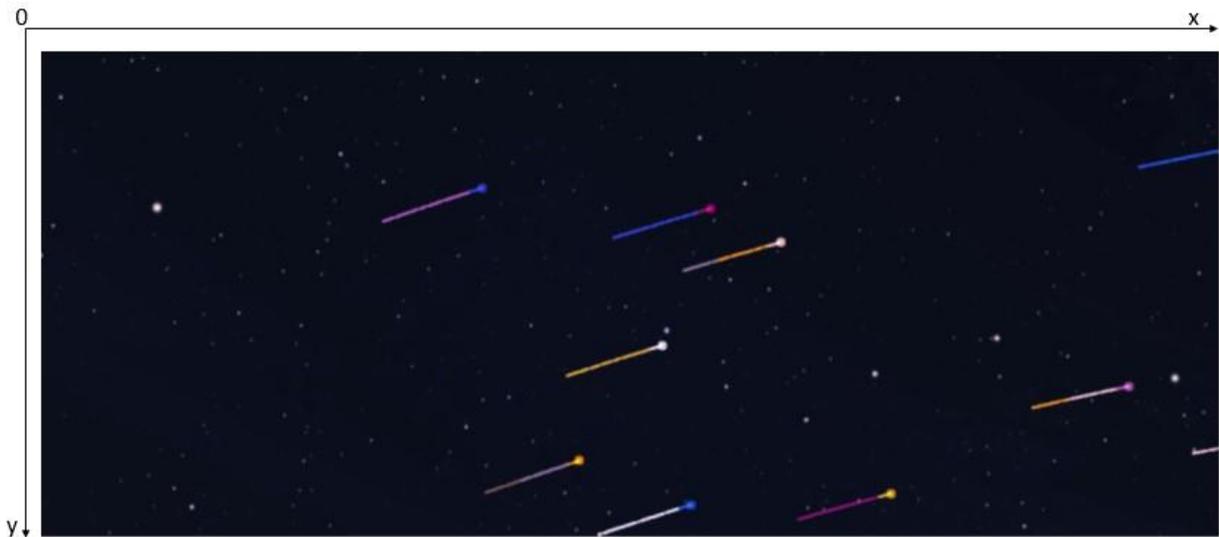


Figura A.1.- Explicación de los ejes de OpenCV

En cuanto al movimiento del servomotor, este gira en los dos ejes mediante las funciones pan y tilt, a las que se les pasa un entero entre -90 y 90, siendo este la posición a la que se mueve. Tilt hace referencia al movimiento en el eje Y. Como se aprecia en la siguiente foto, los valores negativos mueven el servo hacia arriba, mientras que los valores positivos lo sitúan mirando hacia abajo.

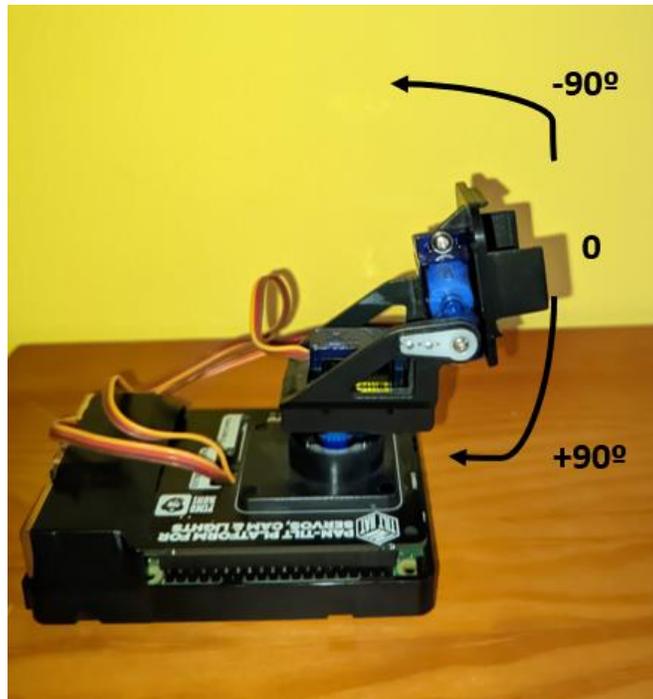


Figura A.2.- Movimiento vertical del servo

Para el movimiento horizontal, se tiene la función pan. En la foto a continuación se ve que, en este caso, los valores negativos hacen que el servo gire hacia la derecha, mientras que los valores positivos lo mueven hacia la izquierda.

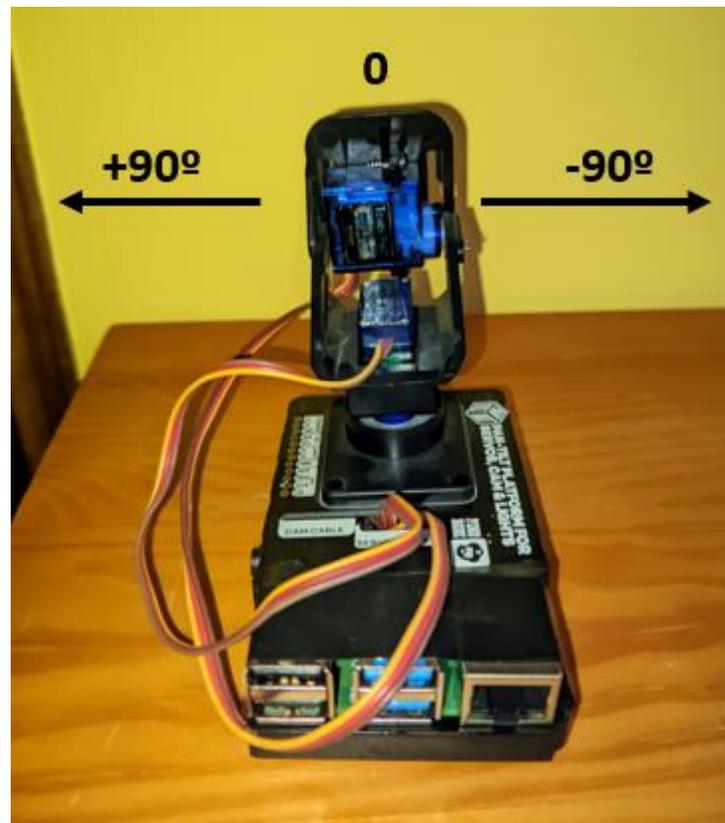


Figura A.3.-Movimiento horizontal del servo

Hay que tener estos conceptos en cuenta para más adelante cuando se explique el código en detalle. Así se entiende mejor la forma en la que se calculan las diferencias de coordenadas en los frames y la forma en la que se pasan los ángulos al servo.

A.3.-ANÁLISIS DEL CÓDIGO.

El programa inicialmente usa la función VideoCapture para coger el vídeo desde una fuente, ya sea un archivo del proyecto o una cámara conectada al ordenador o Raspberry. El primer caso fue el usado para las pruebas con los vídeos del Stellarium, pasando la ruta o el nombre del archivo si se encuentra dentro del propio proyecto. El segundo es más realista, ya que coge los frames en tiempo real de la propia cámara. Por el problema de compatibilidad

con la cámara no se añade la forma de acceder a la QHY 5-II. En su defecto, se pasa el parámetro 0, que es la cámara por defecto, en el caso de mi ordenador personal, su webcam. Para acceder a la cámara QHY 5-II, en el código de prueba se usa la función Qcam de su propia librería.

Después, se declaran los parámetros para la detección de las esquinas o puntos. Al comienzo del código se declararon también los parámetros de los puntos a detectar y de la distancia focal y tamaño del píxel de la cámara, en este caso, los ya mencionados de la QHY 5-II.

El detector de esquinas o puntos Shi-Tomasi es la evolución de uno anterior llamado Harris. Con este método, OpenCV usa la función `goodFeaturesToTrack` con ciertos parámetros para detectar las esquinas. Estos parámetros son los siguientes [41]:

- **maxCorners:** número máximo de esquinas a detectar.
- **qualityLevel:** es un nivel de calidad por debajo del cual se descartan todas las esquinas
- **minDistance:** es la distancia mínima euclidiana entre las distintas esquinas
- **blockSize:** tamaño de los alrededores de la esquina

El método diferencial Lucas-Kanade usa los cuadrados mínimos para calcular el flujo óptico de un punto. Igual que en el otro método, este algoritmo usa unos parámetros: [42]

- **winSize:** tamaño de la ventana de búsqueda
- **maxLevel:** nivel máximo de la pirámide o matriz
- **criteria:** expresa los criterios para finalizar el algoritmo de búsqueda

A continuación, se coge el primer frame del vídeo y se marcan las esquinas en él. Se inicia un bucle que está activo hasta que no haya más frames, esto quiere decir, que se acabó el vídeo. En él se va tratando cada frame, pintando la diferencia entre los diferentes puntos o esquinas tanto visualmente en el vídeo con una línea como por consola, mostrando la diferencia en píxeles como en ángulo. Para calcular este ángulo, se usa esta fórmula.

$$\theta = 206.25 \frac{L}{F}$$

Siendo θ los segundos de arco del ángulo directamente proporcional a L , el tamaño en micras del desplazamiento, esto es, el tamaño del píxel por el desplazamiento en píxeles, e inversamente proporcional a la focal de la cámara en milímetros.

Tras varias pruebas, se llegó a la conclusión que el ángulo que hay que mover el motor tiene que ser la media de las diferencias de los distintos puntos que se tienen. Obviamente, solo se tiene un motor y se le tiene que pasar un valor único. En el caso de una prueba simple como las realizadas con el Stellarium o con una cámara fija, todos los puntos se mueven de forma prácticamente similar en el eje X e Y.

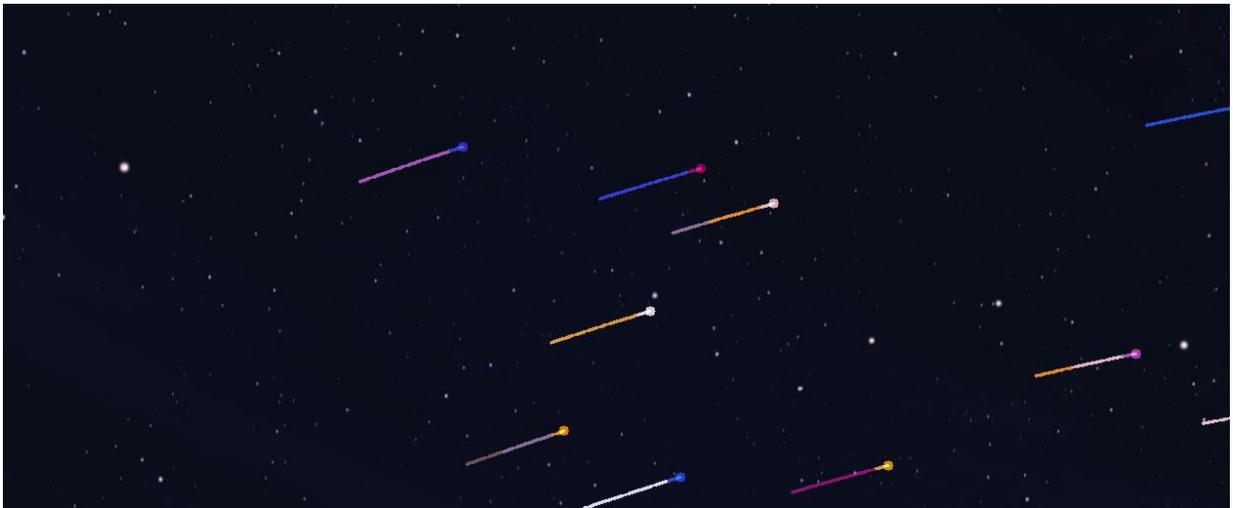


Figura A.4.- Vista del rastreo del programa

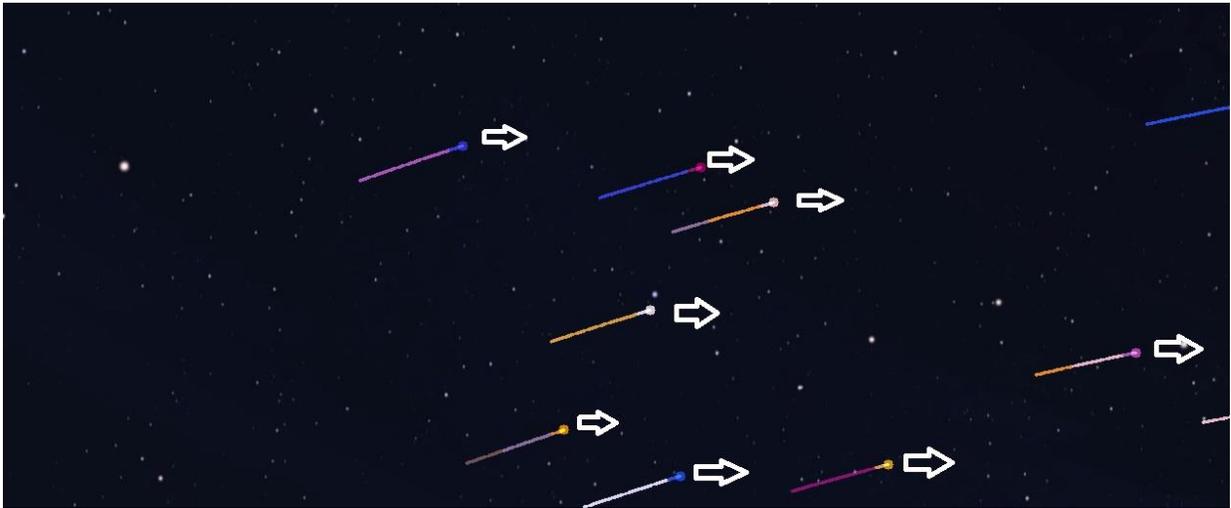


Figura A.5.- Movimiento esperado del programa

Sin embargo, en el entorno real del globo, se puede dar el caso de movimientos más complejos, como por ejemplo una rotación. En este caso, suponiendo que la imagen se rotaría teniendo como centro el punto señalado en verde, los puntos de los exteriores de la imagen sí tienen un desplazamiento significativo en los dos ejes, pero un punto central o cercano al punto de rotación, apenas experimentarían movimiento. Por tanto, lo más razonable es aplicar una media para paliar de alguna forma este problema.

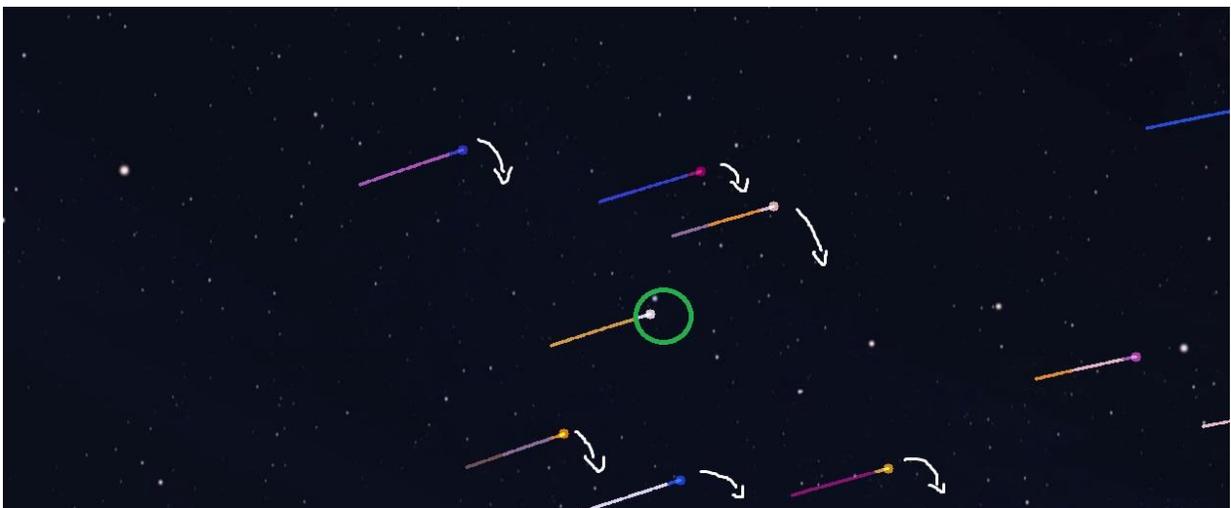


Figura A.6.- Movimiento esperado en caso de rotación

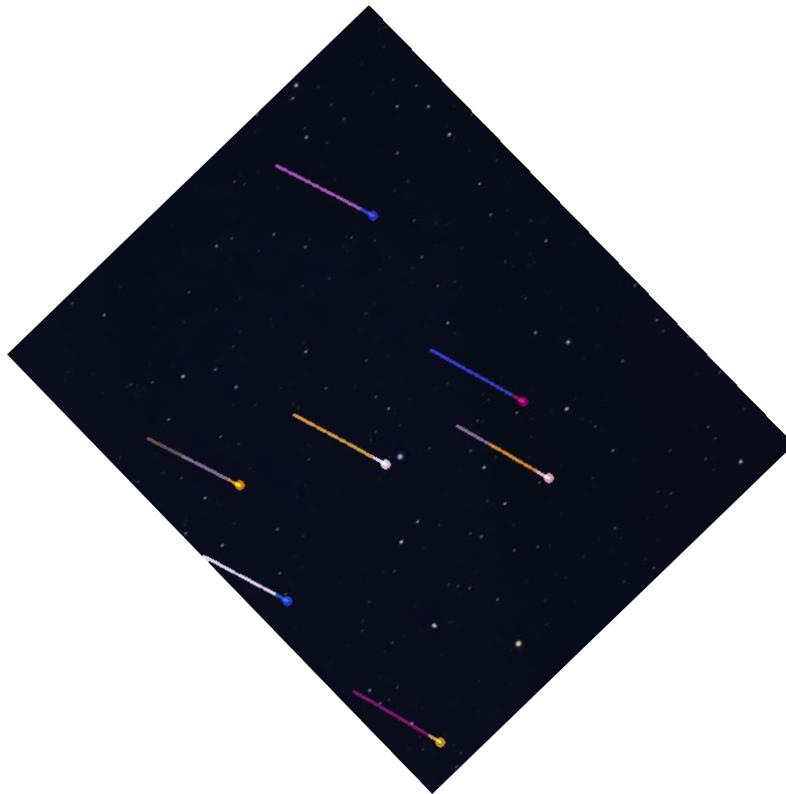


Figura A.7.- Movimiento esperado en caso de rotación (2)

Una vez se calcula esa media se acumula en una variable. Este es un valor decimal, pero a la función del movimiento se le debe pasar un entero. Se convierte la variable, dejando siempre guardado el valor acumulado. Así, aunque en la gran mayoría de iteraciones no se mueva el servomotor porque no se llega al siguiente entero, sí se van acumulando estos pequeños incrementos de movimiento.

Para el desplazamiento, la diferencia de las coordenadas es la resta del punto nuevo y el viejo. Como se explicó anteriormente, esto hace que en el eje X salga un valor positivo, pero en el eje Y salga negativo. Esto tiene mucha importancia, ya que se busca que el servomotor, en un movimiento normal, se mueva hacia la derecha y hacia arriba, lo que significa que los valores esperados deben ser ambos negativos. Por lo tanto, en el caso del ángulo desplazado de forma horizontal, se debe pasar con un signo negativo delante para adaptarse al sistema de movimiento del servomotor.

Antes de repetir el bucle, hay una espera. El valor pasado está en milisegundos y debe coincidir con el tiempo de exposición del vídeo o cámara. En el caso, por ejemplo, de que el tiempo de exposición sea de un segundo, como ocurría en alguna de las pruebas, el valor pasado como parámetro debía ser de 1000 milisegundos.

Además, cuando el programa dura mucho tiempo ejecutándose, todos los puntos detectados inicialmente llegados a cierto momento saldrán del marco de la imagen, dejando al programa de alguna manera ciego al no tener puntos de referencia que seguir ni que calcular. Por lo tanto, en cada iteración del bucle se comprueba si el número de puntos actuales es menor de la mitad de los puntos iniciales para volver a detectar puntos útiles.

A.4.- CÓDIGO.

```
import numpy as np
import cv2 as cv
import pantilthat as pt

#parametros de la camara usados
focal = 10.4
tam_pixel = 3.75

#puntos a seguir
puntos = 10

#funcion que calcula los grados desplazados
def arco_pixel(f, tam, num):
    return (206.26*((num*tam) / f) ) / 3600

#ubicacion o nombre del fichero
filename = 'stelarium.mp4'

#Si se usa un archivo
cap = cv.VideoCapture(filename)

#Si se usa una cámara
```

```
#cap = cv.VideoCapture(0)

# parametros para la deteccion de esquinas ShiTomasi
feature_params = dict( maxCorners = puntos,
                       qualityLevel = 0.1,
                       minDistance = 1,
                       blockSize = 5 )

# parametros para el flow optico lucas kanade
lk_params = dict( winSize = (15, 15),
                  maxLevel = 2,
                  criteria = (cv.TERM_CRITERIA_EPS |
cv.TERM_CRITERIA_COUNT, 10, 0.03))

# crea los colores para las lineas de rastreo
color = np.random.randint(0, 255, (100, 3))

# obtiene el primer frame y localiza las esquinas
ret, old_frame = cap.read()
old_gray = cv.cvtColor(old_frame, cv.COLOR_BGR2GRAY)
p0 = cv.goodFeaturesToTrack(old_gray, mask = None, **feature_params)

# crea una mascara de imagen para dibujar los puntos
mask = np.zeros_like(old_frame)

# variables para acumular el ángulo que se mueve
arco_pan_double = 0
arco_tilt_double = 0

# variable que controla cuantos puntos hay en cada momento
# se inicializa con el doble del valor ya que el tamaño del vector de
puntos es siempre el doble del indicado
puntos_actuales = puntos*2

while(1):

    # se lee un frame
    ret, frame = cap.read()
    if not ret:
        print('No hay frames leídos')
        break

    # si el numero de puntos actuales baja de la mitad de los
    # iniciales, se vuelven a buscar puntos
    if puntos_actuales < puntos:
        p0 = cv.goodFeaturesToTrack(old_gray, mask=None,
**feature_params)
```

```
#se calcula el flujo optico
frame_gray = cv.cvtColor(frame, cv.COLOR_BGR2GRAY)
p1, st, err = cv.calcOpticalFlowPyrLK(old_gray, frame_gray, p0,
None, **lk_params)

# se eligen nuevos puntos
if p1 is not None:
    good_new = p1[st==1]
    good_old = p0[st==1]
    print(good_new.size)
    puntos_actuales = good_new.size
    print(good_old.size)

# variables para calcular la suma de los movimientos de los
puntos
aux_x = 0
aux_y = 0

# bucle que recorre todos los puntos
for i, (new, old) in enumerate(zip(good_new, good_old)):

    # Nuevo punto
    print(new)

    # Punto anterior
    print(old)

    print("Diferencia en pixeles en el eje X del punto:")
    print(new[0]-old[0])

    print("Diferencia angulo en eje x")

    # diferencia en el eje x
    arco_pan = arco_pixel(focal, tam_pixel, new[0]-old[0])
    print(arco_pan)

    print("Diferencia en pixeles en el eje Y del punto:")
    print(new[1]-old[1])
    print("Diferencia angulo en eje y")

    # diferencia en el eje y
    arco_tilt = arco_pixel(focal, tam_pixel, new[1]-old[1])
    print(arco_tilt)

    aux_x = aux_x + arco_pan
    aux_y = aux_y + arco_tilt
```

```
# se dibujan las lineas

a, b = new.ravel()
c, d = old.ravel()
mask = cv.line(mask, (int(a), int(b)), (int(c), int(d)),
color[i].tolist(), 2)
frame = cv.circle(frame, (int(a), int(b)), 5,
color[i].tolist(), -1)

# se calcula la media
arco_pan_movimiento_media_x = aux_x / (good_new.size / 2)
arco_tilt_movimiento_media_y = aux_y / (good_new.size / 2)

print("ARCO PAN movimiento media x")
print(arco_pan_movimiento_media_x)
print("ARCO TILT movimiento media y")
print(arco_tilt_movimiento_media_y)

# Movimiento del servo en horizontal (eje x)

arco_pan_double = arco_pan_double +
arco_pan_movimiento_media_x
arco_pan_int = int(round(arco_pan_double))

print("ARCO PAN double")
print(-arco_pan_double)
print("ARCO PAN int")
print(-arco_pan_int)
pt.pan(round(-arco_pan_int))

# Movimiento del servo en vertical (eje y)

arco_tilt_movimiento_y = arco_pixel(focal, tam_pixel,
good_new[0][1] - good_old[0][1])
print("ARCO TILT movimiento y")
print(arco_tilt_movimiento_y)
arco_tilt_double = arco_tilt_double +
arco_tilt_movimiento_media_y
arco_tilt_int = int(round(arco_tilt_double))

print("ARCO TILT double")
print(arco_tilt_double)
print("ARCO TILT int")
print(arco_tilt_int)
pt.tilt(round(arco_tilt_int))
```

```
#se muestra la imagen
img = cv.add(frame, mask)
cv.imshow('frame', img)

#para la prueba con Stellarium
k = cv.waitKey(30) & 0xff
#1000 ms para la prueba con la camara
#k = cv.waitKey(1000) & 0xff
if k == 27:
    break
# se actualizan los puntos
old_gray = frame_gray.copy()
p0 = good_new.reshape(-1, 1, 2)

cv.destroyAllWindows()
```

6. Bibliografía

«Astronomía observacional,» [En línea]. Available:

- 1] https://es.wikipedia.org/wiki/Astronom%C3%ADa_observacional. [Último acceso: 4 Junio 2022].

«Astronomía: la ciencia más mágica,» [En línea]. Available:

- 2] <https://www.elmundo.es/ciencia/2015/09/11/55eff8d122601d995a8b4572.html>. [Último acceso: 5 Junio 2022].

«Astronomía visible,» [En línea]. Available:

- 3] https://es.wikipedia.org/wiki/Astronom%C3%ADa_visible. [Último acceso: 5 Junio 2022].

«Observatorio,» [En línea]. Available: <https://es.wikipedia.org/wiki/Observatorio>.

- 4] [Último acceso: 4 Junio 2022].

«Astronomía óptica,» [En línea]. Available: https://enciclopedia_universal.es-academic.com/8297/Astronom%C3%ADa_%C3%B3ptica.

- 5] [Último acceso: 2022 Junio 6].

«Astronomía primitiva,» [En línea]. Available:

- 6] <https://www.astromia.com/historia/astromagia.htm>. [Último acceso: 5 Junio 2022].

«Breve historia de la observación astronómica,» [En línea]. Available:

- 7] <http://simplementeeluniverso.blogspot.com/2012/05/breve-historia-de-la-observacion.html#:~:text=El%20primero%20que%20le%20dio,conocidos%20como%20%E2%80%9Csat%C3%A9lites%20galileanos%E2%80%9D>. [Último acceso: 5 Junio 2022].

«¿Por qué titilan las estrellas?,» [En línea]. Available:

- 8] https://elpais.com/elpais/2014/08/06/actualidad/1407347167_923598.html. [Último acceso: 7 Junio 2022].

«Observatorio espacial,» [En línea]. Available:

- 9] https://es.wikipedia.org/wiki/Observatorio_espacial. [Último acceso: 6 Junio 2022].

- «Cosmos 215,» [En línea]. Available: https://es.wikipedia.org/wiki/Cosmos_215.
10] [Último acceso: 7 Junio 2022].
- «El Telescopio espacial Hubble,» [En línea]. Available:
11] <https://www.astromia.com/astrofotografia/telescopiohubble.htm>. [Último acceso: 7 Junio 2022].
- «Las 8 misiones espaciales más caras de la historia,» [En línea]. Available:
12] <https://computerhoy.com/listas/life/misiones-espaciales-mas-caras-de-la-historia-590205>. [Último acceso: 7 Junio 2022].
- «Los principales observatorios espaciales que exploran el universo,» [En línea].
13] Available: <https://es.digitaltrends.com/espacio/principales-observatorios-espaciales/>. [Último acceso: 7 Junio 2022].
- «Los Puntos de Lagrange,» [En línea]. Available:
14] <https://josevicentediaz.com/astrofotografia/los-puntos-de-lagrange/>. [Último acceso: 1 Julio 2022].
- «Explorar el universo a bordo de un globo,» [En línea]. Available:
15] <https://www.elespectador.com/ciencia/explorar-el-universo-a-bordo-de-un-globo-articulo-649576/>. [Último acceso: 5 Julio 2022].
- «El proyecto ORISON estudia el cosmos desde globos estratosféricos,» [En línea].
16] Available: <https://www.dicyt.com/noticias/el-proyecto-orison-estudia-el-cosmos-desde-globos-estratosfericos>. [Último acceso: 5 Julio 2022].
- «La misión de la NASA que estudiará el cosmos con un globo estratosférico.,»
17] [En línea]. Available: <https://www.mdsc.nasa.gov/index.php/2020/07/27/la-mision-de-la-nasa-que-estudiara-el-cosmos-con-un-globo-estratosferico/>. [Último acceso: 5 Julio 2022].
- «Globo meteorológico,» [En línea]. Available:
18] https://es.wikipedia.org/wiki/Globo_meteorol%C3%B3gico. [Último acceso: 16 Junio 2022].
- «Cómo funcionan los "globos sonda" que se elevan a diario y ayudan a los
19] pronósticos meteorológicos,» [En línea]. Available:

- <https://www.bbc.com/mundo/noticias-56825880>. [Último acceso: 16 Junio 2022].
- «Estratosfera,» [En línea]. Available: <https://es.wikipedia.org/wiki/Estratosfera>.
20] [Último acceso: 16 Junio 2022].
- «Estratosfera,» [En línea]. Available: <https://concepto.de/estratosfera/>. [Último
21] acceso: 16 Junio 2022].
- «Estratosfera,» [En línea]. Available:
22] <https://www.capasdelatierra.org/estratosfera/>. [Último acceso: 16 Junio 2022].
- «Raspberry Pi: Crea proyectos DIY por muy poco dinero,» [En línea]. Available:
23] <https://www.profesionalreview.com/2021/07/18/que-es-raspberry-pi/>. [Último acceso: 21
Junio 2022].
- «<https://www.astropolar.es/139-camara-qhy5l-ii-monocroma.html>,» [En línea].
24] Available: <https://www.astropolar.es/139-camara-qhy5l-ii-monocroma.html>. [Último
acceso: 23 Junio 2022].
- «Stellarium,» [En línea]. Available: <https://es.wikipedia.org/wiki/Stellarium>.
25] [Último acceso: 16 Junio 2022].
- «OpenCV,» [En línea]. Available: <https://opencv.org/>. [Último acceso: 23 Junio
26] 2022].
- «OpenCV,» [En línea]. Available: <https://es.wikipedia.org/wiki/OpenCV>. [Último
27] acceso: 23 Junio 2022].
- «¿Qué es OpenCV? Instalación en Python y ejemplos básicos,» [En línea].
28] Available: <https://www.inesem.es/revistadigital/informatica-y-tics/opencv/>. [Último
acceso: 23 Junio 2022].
- «FireCapture, la herramienta de captura definitiva,» [En línea]. Available:
29] [https://www.planetografia.es/astrofotografia-planetaria/curso-de-astrofotografia-
planetaria/tutorial-firecapture-astrofotografia-planetaria/](https://www.planetografia.es/astrofotografia-planetaria/curso-de-astrofotografia-planetaria/tutorial-firecapture-astrofotografia-planetaria/). [Último acceso: 23 Junio 2022].
- «About Astrometry.net,» [En línea]. Available: <https://nova.astrometry.net/>.
30] [Último acceso: 7 Julio 2022].
- «Cómo elegir la IMU adecuada para un UAV,» [En línea]. Available:

- 31] <https://www.embention.com/es/news/la-imu-adecuada-para-un-uav/#:~:text=La%20Unidad%20de%20Medici%C3%B3n%20Inercial,del%20veh%C3%ADculo%20a%C3%A9reo%20no%20tripulado..> [Último acceso: 7 Julio 2022].
«All Sky Monochrome Time-Lapse on Night of Sun. 10th October 2021. Kelling Heath Star Party,» [En línea]. Available: https://vimeo.com/633709881?embedded=true&source=vimeo_logo&owner=10155138. [Último acceso: 8 Julio 2022].
«Etcher,» [En línea]. Available: <https://es.wikipedia.org/wiki/Etcher>. [Último acceso: 25 Junio 2022].
«Raspberry Pi 4 es oficial,» [En línea]. Available: <https://www.xataka.com/ordenadores/raspberry-pi-4-caracteristicas-precio-ficha-tecnica>. [Último acceso: 7 Julio 2022].
«Detección de rostros de OpenCV mediante el clasificador en cascada de Haar,» [En línea]. Available: <https://www.delftstack.com/es/howto/python/opencv-haar-cascade/>. [Último acceso: 25 Junio 2022].
«Cascade Classifier,» [En línea]. Available: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html. [Último acceso: 25 Junio 2022].
«Find and Draw Contours using OpenCV | Python,» [En línea]. Available: <https://www.geeksforgeeks.org/find-and-draw-contours-using-opencv-python/#:~:text=Contours%20are%20defined%20as%20the,the%20contours%20from%20the%20image..> [Último acceso: 25 Junio 2022].
«Find contours in your image,» [En línea]. Available: https://docs.opencv.org/3.4/df/d0d/tutorial_find_contours.html. [Último acceso: 25 Junio 2022].
«Optical Flow,» [En línea]. Available: https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html. [Último acceso: 26 Junio 2022].
«Optical Flow in OpenCV (C++/Python),» [En línea]. Available:

40] <https://learnopencv.com/optical-flow-in-opencv/>. [Último acceso: 26 Junio 2022].

«Detector de esquina Shi-Tomasi y buenas características para rastrear,» [En

41] línea]. Available: <https://unipython.com/detector-esquina-shi-tomasi/>. [Último acceso: 27 Junio 2022].

«Método Lucas-Kanade,» [En línea]. Available:

42] https://es.wikipedia.org/wiki/M%C3%A9todo_Lucas%E2%80%93Kanade. [Último acceso: 27 Junio 2022].