



PONTIFICIA  
UNIVERSIDAD  
CATÓLICA DE  
VALPARAÍSO



**Vicente Westerhout Aliste**

# **Desarrollo de aplicaciones para cámaras basadas en eventos**

**Informe Proyecto de Título de Ingeniero Civil Electrónico**



**Escuela de Ingeniería Eléctrica  
Facultad de Ingeniería**

**Valparaíso, 26 de noviembre de 2021**



## Desarrollo de aplicaciones para cámaras basadas en eventos

Vicente Westerhout Aliste

Informe Final para optar al título de Ingeniero Civil Electrónico,  
aprobada por la comisión de la  
Escuela de Ingeniería Eléctrica de la  
Facultad de Ingeniería de la  
Pontificia Universidad Católica de Valparaíso  
conformada por

Sr. Esteban Vera Rojas  
Profesor Guía

Sr. Gonzalo Farias Castro  
Segundo Revisor

Valparaíso, 26 de noviembre de 2021

# Resumen

En los últimos años existe una gran preocupación a nivel mundial debido a la rapidez con la que crece la población de satélites y objetos orbitando la Tierra, sobre todo en orbitas cercanas (LEO). La preocupación surge del llamado Kessler effect el cual predice que debido a la finita posibilidad que existe de colisiones entre satélites, gran cantidad de basura espacial se puede llegar a generarse debido al efecto en cadena que esto ocurra. En el futuro, un cinturón de basura impedirá nuevos lanzamientos al espacio.

Por esto que la vigilancia espacial de satélites y basura espacial toma importancia. Mantener una vigilancia activa para tener un catálogo actualizado de objetos es la única solución viable en estos momentos, pero es una tarea tediosa debido a la gran cantidad de fragmentos de basura hoy en día.

Se propone en este documento un sistema que permita la detección de objetos en orbitas LEO, uniendo los beneficios que suponen las cámaras basadas en eventos, sensores novedosos con los cuales se cambia el paradigma en la obtención de imágenes.

La aplicación propuesta es capaz de detectar y referenciar en coordenadas ecuatoriales las mediciones realizadas y almacenarlas ocupando un mínimo de espacio en disco y capacidad de procesamiento, debido a las bondades de la cámara de eventos.

Por último, se presentan los resultados de las detecciones realizadas con la aplicación implementada con un telescopio además de presentar resultados en contraste con otro setup utilizando una cámara de alta resolución y un telescopio más grande.

Palabras claves: Vigilancia espacial, Kessler Effect, cámaras basadas en eventos, cámaras basadas a cuadros, LEO, basura espacial, catálogo de satélites.

# Abstract

In the past years, there is great concern worldwide due to the fast pace in which the population of satellites and objects orbiting Earth is growing, especially in Low Earth Orbits (LEO). The concern arise from the so called Kessler effect which predicts that due to the finite possibility of collisions between satellites, a large amount of space debris can be generated and then a chain effect made from this can occur. In the future, a debris belt will prevent further launches into space.

This is why the space surveillance of satellites and space debris takes on importance. Keeping an active watch for an up to date catalog of objects is the only viable solution at the moment, but it is a tedious task due to the large amount of debris in space.

In this document, a system that allows the detection of objects in LEO is proposed, combining the benefits of event based cameras, novel sensors with which the paradigm in obtaining images is changed.

The proposed application is capable of detect and reference the measurements made in equatorial coordinates and storing them occupying a minimum of disk space and processing capacity, due to the benefits of the event camera.

Finally, the results of detections made with the application implemented with a telescope are presented, and in addition results with another camera setup are presenting as well, to help illustrate the contrast between event base sensors and frame based cameras.

Key words: Space Surveillance, Kessler effect, event based cameras, frame based cameras, LEO, space debris, satellite catalogs.

# Índice general

Introducción .....	1
1 Antecedentes generales y propuesta.....	4
1.1 Antecedentes.....	4
1.1.1 Satélites en órbita.....	4
1.1.2 Estado actual de los lanzamientos de satélites .....	5
1.1.3 El Efecto Kessler y la basura espacial .....	7
1.1.4 Colisiones a hiper velocidad .....	8
1.1.5 Contexto social y económico .....	10
1.2 Problematización.....	12
1.3 Propuesta.....	12
1.4 Cámaras basadas en eventos.....	13
1.4.1 Cámara de eventos : Prophesee Evaluation Kit .....	16
1.4.2 Parámetros internos .....	18
1.4.3 Formato de archivos .RAW .....	20
2 Estado del arte .....	23
2.1.1 Pre-procesamiento del flujo de eventos .....	23
2.1.2 Detección de objetos .....	24
2.1.3 Rastreo basado en eventos.....	27
3 Referenciación de objetos en órbita.....	29
3.1 Sistema de coordenadas Earth-Centered Inertial (ECI) .....	29
3.1.1 Posición de un observador en ECI .....	30
3.1.2 Objetos en órbita en ECI .....	33
3.2 Estandar para mensajes de objetos en órbita.....	34
3.2.1 Formato Two Line Element .....	35
3.3 Stellarium .....	38
4 Aplicación para la detección de satélites .....	40
4.1 Librerías .....	40
4.1.1 Metavision::Device.....	40
4.1.2 Metavision::I_EventStream.....	41

4.1.3 Metavision::I_DeviceControl .....	41
4.1.4 Metavision::I_EventDecoder .....	41
4.1.5 Metavision::I_LL_Biases .....	42
4.1.6 Metavision::I_Decoder .....	42
4.2 Estructura del código .....	42
4.2.1 Nucleo de decodificación.....	43
4.2.2 Nucleo de visualización.....	44
4.2.3 Nucleo de detección .....	45
4.3 Marco teórico .....	47
4.3.1 Transformada de Hough .....	48
4.3.2 Conversión de coordenadas horizontales a coordenadas ecuatoriales.....	49
5 Implementación y resultados .....	51
5.1 Implementación .....	51
5.2 Resultados .....	54
Discusión y conclusiones.....	59
Bibliografía.....	62

# Introducción

En 1957 cuando se lanzó el primer satélite el Sputnik 1 se dio comienzo a la comúnmente llamada “Era Espacial”, fundamentada bajo objetivos políticos debido a la carrera espacial es sin duda uno de los hitos que motivaría la búsqueda y desarrollo de avances científicos en la época. Desde dicho momento es que múltiples satélites y sondas se han lanzado para orbitar o transitar el espacio cercano a la Tierra con diferentes objetivos, además de las múltiples misiones con tripulantes realizados por Estados Unidos y la Unión Soviética.

Progresivamente la capacidad de poner satélites en órbita no fue solo de estas grandes potencias ya que otros países se les unieron rápidamente como lo fueron Francia, Japón y China. Hoy en día una decena de países puede lanzar satélites y muchos más entran en la lista de poseer alguno gracias a lanzamientos en conjunto o lanzamientos de terceros, por otro lado, estados no son los únicos en tener satélites, sino que existe una gran cantidad de satélites comerciales propiedad de marcas y empresas internacionales. Esto se debe a que se reconocen aún como parte esencial para ciertos fines como lo son las comunicaciones, la meteorología, el reconocimiento (como la captura de imágenes terrestres), la navegación (sistemas como el GPS), la astronomía y propósitos militares varios.

Actualmente, aunque la exploración espacial no provoca la misma excitación que antaño el lanzamiento de satélites nunca ha tenido un mayor auge debido a la miniaturización como los CubeSat con dimensiones de 10x10x10 cm por lado y a proyectos codiciosos como las constelaciones de satélites como el proyecto Starlink de SpaceX que busca dar internet de banda ancha gratis al planeta entero. Mandar un satélite a órbita nunca ha sido más fácil y nunca ha habido tantos lanzamientos en tan corto periodo, SpaceX en tan solo el lapso de un mes (marzo 2021) ha lanzado alrededor de 300 satélites del mencionado Starlink.

Lamentablemente los satélites no son los únicos residentes de las órbitas terrestres resultantes de la Era Espacial ya que la mayoría de los cuerpos en órbita (Resident Space Objects - RSO) hoy en día corresponden a la llamada basura espacial. La basura espacial son el resultado de la corta, pero frecuente historia del ser humano en realizar lanzamientos espaciales, son los residuos provocados como partes de las etapas de propulsión, satélites inactivos y fragmentos de estos mismos. La generación de basura espacial pone en riesgo a los satélites en órbita además de dificultar futuros lanzamientos, y aunque la probabilidad de colisión entre objetos en órbita es pequeña colisiones han ocurrido y seguirán ocurriendo. Donald Kessler en 1978 propuso un

modelo que muestra como las continuas colisiones entre satélites o fragmentos en órbita generan más fragmentos, aumentando la probabilidad de colisiones y generando así una reacción en cadena que provocará un cinturón de basura espacial en órbita, imposibilitando futuras operaciones espaciales.

Considerando que en los primeros años de exploración espacial no se tomó ninguna precaución referente al problema de la basura espacial, es que actualmente es posible contar cientos de miles de objetos inactivos catalogados de un tamaño de 1cm o mayor, objetos que a pesar de su pequeño tamaño gracias a la gran velocidad con la que se orbita pueden fácilmente perforar las capas externas de los satélites en el mejor de los casos. Si se suma el continuo y constante aumento de lanzamientos, la cantidad de objetos en órbita y por tanto la probabilidad de colisiones aumenta exponencialmente, y aun que existe mayor visibilidad del problema latente en la basura espacial, una gran parte de los operarios de satélites suele ignorar o asumir el riesgo debido a razones culturales o económicas.

Debido a los puntos anteriores es que existe un gran énfasis en realizar un monitoreo constante de estos objetos en órbita ya que no existen métodos factibles de deshacerse de estos fragmentos.

En este proyecto de titulación se propone utilizar las llamadas cámaras basadas en eventos, las cuales poseen sensores que cambian el paradigma de la adquisición de imágenes. Con este tipo de dispositivos se pueden capturar los cambios en la intensidad de la luz en la escena y poseen variadas ventajas sobre las cámaras convencionales como una mayor resolución temporal, un gran rango dinámico, bajo consumo, entre otras. Se busca, por tanto, la creación de una aplicación para vigilancia espacial utilizando estas novedosas cámaras.

El diseño de aplicaciones para estos dispositivos no es tarea fácil, debido al corto tiempo que se han venido desarrollando estas cámaras los algoritmos son escasos y la exploración casi nula, esto se debe a lo diferentes que es la entrada de estas cámaras, un flujo continuo asincrónico. Es por esto que se busca investigar y probar métodos para el procesamiento de los flujos nativos creados por la cámara de eventos, para lograr este cometido se estudia el escaso estado del arte con respecto a aplicaciones similares a las que se quiere lograr, se describen el formato que cumplen los archivos creados por esta cámara y se comienza a explorar con las librerías disponibles.

Por otro lado, la observación astronómica que se ha venido perfeccionando hace siglos ha permitido la incorporación de procedimientos y estándares en lo que respecta a la captura y rastreo de cuerpos artificiales orbitando la Tierra. Se hace un estudio de los diferentes tipos de coordenadas existentes para poder referenciar los objetos detectados con el fin de poder luego emparejarlos con algún catalogo internacional existente. Esto permitirá conocer qué tipo de objeto ha sido detectado o si no se encuentra en el catálogo estándar.

Luego se propone la creación de la aplicación por medio del lenguaje C++. Esta herramienta debe ser capaz de capturar eventos mientras detecta objetos que pasen a través de la escena automáticamente, almacenar algún tipo de muestra y en última instancia proveer con los datos necesarios que permitan hacer un emparejamiento con objetos con trayectorias conocidas. Para esto último se utiliza un software de código abierto que permite la descarga de catálogos de las



fuentes que uno desee, pero además posee herramientas que ayudaran a la captura de efectiva al realizar grabaciones.

Por último, se realizan dos tipos de mediciones. La primera corresponde a grabaciones con un setup simple con el cual se cuenta la gran mayoría del tiempo, esta configuración considera la cámara de eventos conectada a un telescopio pequeño y montado en un trípode corriente. El segundo tipo de pruebas realizados se realiza montando la cámara y el telescopio de la primera configuración en conjunto con un telescopio más grande con una cámara de alta resolución, esto se realiza para visualizar el gran contraste que existe entre ambas tecnologías y que esta prueba dará a entender de forma interactiva.

### **Objetivos Generales**

Resumiendo, como objetivo general se plantea crear un sistema de vigilancia espacial de satélites y basura espacial en LEO utilizando una cámara de eventos.

Como objetivos específicos se identifican los siguientes:

- Capturar objetos orbitando en LEO con una cámara de eventos.
- Generar un catálogo propio con los objetos capturados.
- Poder comparar los objetos capturados con un catálogo internacional.
- Desarrollar una aplicación que procese los eventos y permita realizar los objetivos anteriores en tiempo real.

# 1 Antecedentes generales y propuesta

## 1.1 Antecedentes

Los satélites artificiales representan un avance importante en como el ser humano interactúa con el planeta, su relevancia radica principalmente en que múltiples tareas solo pueden ser realizadas por estos como lo son las telecomunicaciones, la meteorología, el monitoreo y la navegación. Además, en el día a día somos capaces de percibir las ventajas que estos nos ofrecen como lo es visualizar las imágenes satelitales de Google maps u ocupar el GPS incluido en cualquier smartphone. Por tanto, el mantener estos dispositivos funcionando y en condiciones integras es esencial para permitir que la comunidad en general siga recibiendo los mencionados beneficios.

Una de las tareas necesarias para proteger la integridad de los satélites es conocer la ubicación y trayectorias de estos cuerpos en todo momento para así evitar posibles colisiones entre satélites u otros cuerpos. Dichas colisiones, aunque poco probables, han ocurrido y son cada vez más posibles, y por lo general estas resultan en perdidas catastróficas debido a la gran velocidad que alcanzan estos cuerpos al orbitar nuestro planeta. En las secciones siguientes se entregan antecedentes que dan cuenta del por qué estas colisiones son un peligro latente hoy en día y las razones por las cuales el panorama de cara al futuro puede llegar a empeorar de manera significativa.

### 1.1.1 Satélites en orbita

Como se mencionó anteriormente, las funciones y características de satélites son tan variadas como se puedan imaginar. Esto se intensifica si se considera que se han puesto en órbita satélites desde 1957 y la gran diferencia tecnológica entre esa fecha y la actual. Existen satélites tan grandes que llegan a un peso de un par de decenas de toneladas (satélites geosíncronos y estaciones espaciales) hasta tan pequeños con pesos aproximados de 1.5[Kg] (cubesats). Por otro lado, una de las características de diseño más importantes es la distancia a la cual se dejará un satélite orbitando, de esta girando todas las demás características de diseño. Por lo general, los operarios de satélites escogen orbitas específicas y por tanto, los satélites se tienden a clasificar dependiendo de la distancia a la que pertenecen dichas orbitas [1]:

- Órbita terrestre alta: Corresponde órbitas con altitudes mayores a 35790[Km] desde la superficie terrestre, esta distancia es especial ya que corresponde a la llamada Órbita

Geosíncrona Ecuatorial (GEO), una órbita geoestacionaria con inclinación y excentricidad igual a cero sobre el ecuador. Los satélites que orbitan en GEO poseen velocidad relativa nula con respecto a la Tierra y por tanto parece como si estuvieran suspendidos en un punto. Esto último es de suma importancia para tareas como la meteorología y la vigilancia, en las cuales se necesita mantener una zona en particular bajo monitoreo constante. Además, este tipo de orbitas es útil para las comunicaciones ya que en tierra es posible apuntar una antena a uno de estos satélites debido a que estos permanecen en un mismo lugar.

- Órbita terrestre media: Mientras más cerca están a la Tierra, los satélites orbitan con mayor velocidad. Este tipo de orbita se ubican entre los 2000[Km] y los 35790[Km]. Se pueden mencionar dos tipos de orbitas notables, la primera es la Órbita Semi-Síncrona, es una órbita de baja excentricidad con una altitud equivalente a 20200[Km] sobre la superficie terrestre. Satélites en esta orbita demoran aproximadamente 12 horas en dar la vuelta al globo, esta órbita es usada por los satélites para el Sistema Global de Posicionamiento (GPS). La segunda es la órbita Molniya, es una órbita de origen ruso útil para observar latitudes superiores, posee una extrema excentricidad lo que permite que satélites en esta orbita tomen menos tiempo en cursar la parte relativa al hemisferio sur, pero mucho tiempo más la parte relativa al hemisferio norte y dando la vuelta al mundo cada 12 horas al igual que la Semi-Síncrona.
- Órbita terrestre baja: Los satélites ubicados a una altitud menor a los 2000[Km] de la superficie terrestre se les considera en una Órbita terrestre baja (LEO). Satélites en LEOs demoran alrededor de una hora y media en dar la vuelta a la Tierra por lo que su velocidad es muy alta. No poseen limitantes como la órbita GEO por lo que se pueden encontrar orbitas con distintas inclinaciones como el satélite TRMM (Tropical Rainfall Measuring Mission) con 35° que monitorea lluvias en los trópicos o la órbita Solar-Síncrona que es un tipo de órbita polar ya que posee una inclinación de 98°, satélites en esta orbita se ocupan para detectar cambios en el tiempo de condiciones como la temperatura o el clima, esto se debe a que cada paso orbital ocurre a la misma hora.

### **1.1.2 Estado actual de los lanzamientos de satélites**

Hoy en día nunca ha sido más fácil hacer lanzamientos y poner en órbita satélites, esto se debe principalmente a dos causas, la miniaturización y las constelaciones.

El avance tecnológico en múltiples áreas ha permitido que la miniaturización de satélites hoy en día sea posible, y si bien un satélite “pequeño” se considera a cualquiera con un peso menor a 300[Kg], hoy se pueden encontrar satélites cuyas especificaciones de diseño no superan los 10[cm] por lado y 1.5[Kg] de peso, estos son los conocidos como CubeSat debido a su forma cubica. Los CubeSat comenzaron siendo un proyecto educacional para que las universidades pudiesen tener su propio programa espacial, la intención era proveer a los estudiantes con un diseño básicos que pudiesen desarrollar ellos mismos. Prontamente agencias gubernamentales y grupos comerciales adoptaron este tipo de tecnología reconociendo en la pequeña y

estandarizada plataforma de los CubeSat una forma eficiente de reducir los costos de lanzamiento, esto debido a que los CubeSat se lanzan en dispensadores que transportan más de una unidad de estos dispositivos (En la Figura 1-1 se muestran un CubeSat junto a un dispensador) [2].



Figura 1-1 Unidad CubeSat junto a un dispensador de 3 unidades (3U)

Por otro lado, las constelaciones de satélites no son un concepto nuevo ya que desde 1993 con la constelación de satélites norteamericanos para la navegación o mejor conocidos como GPS (Global Navigation System) son utilizados habitualmente incluso por la comunidad civil. El concepto es simple, debido a que enviar satélites a GEO es bastante costoso, la alternativa es colocar múltiples satélites en órbitas más cercanas generando un sistema que evite la pérdida de señal en la Tierra, es decir, que en todo momento por lo menos un satélite de la constelación este suministrando el servicio en específico. Así existen múltiples constelaciones de servicios de navegación en MEO (20200 [Km]), la conocida constelación de comunicación satelital Iridium en LEO (781 [Km]) y la novedosa Starlink.

Starlink es el proyecto ambicioso de SpaceX de proveer un servicio de internet de banda ancha satelital, para esto planean crear una constelación de 12000 satélites en órbitas a 550 [Km] de la superficie aproximadamente. Esto nuevamente es posible a la miniaturización de las tecnologías de los satélites, ya que los satélites de Starlink poseen un peso bastante pequeño de unos 227 [Kg] (3 veces menor que los satélites Iridium 689[Kg] que cumplen funciones relativamente parecidas), pero además por los esfuerzos de SpaceX en crear etapas de lanzamiento reutilizables que bajan los costos asociados. Los lanzamientos comenzaron en mayo del 2019 y cada lanzamiento pone en órbita 60 satélites, hasta la fecha 1844 están ya en órbita y una prueba beta del servicio de banda ancha ya está disponible en algunos lugares del mundo.

### 1.1.3 El Efecto Kessler y la basura espacial

En 1978, D. Kessler expuso los riesgos que supondría mantener desatendida la situación referente a todos los objetos inactivos en la órbita terrestre como fragmentos de cohetes o partes de etapas de lanzamiento, pedazos de satélites averiados o incluso satélites que cuya vida útil ya ha caducado, todos estos objetos que no poseen utilidad y que siguen orbitando el planeta se les conoce como basura espacial. Kessler creó un modelo probabilístico que demuestra como existe una finita posibilidad de que satélites en funcionamiento choquen generando fragmentos de basura espacial, estos fragmentos a su vez elevan el riesgo de nuevas colisiones generando así un ciclo vicioso que rápidamente supondría la creación de un cinturón de basura espacial alrededor de la Tierra. Similar a como existen cinturones de asteroides alrededor de planetas, este cinturón de basura espacial provocaría una limitante importante en las misiones espaciales futuras como también en el lanzamiento de nuevos satélites y la autonomía de los ya se encuentran en órbita, sin mencionar claro que las colisiones también ponen en riesgo a la población en general si un trozo importante de esta basura lograra ingresar a la atmósfera. Este fenómeno se le conoce como el efecto Kessler o síndrome Kessler [3].

Tanto los CubeSat como las constelaciones descritas anteriormente, contribuyen al aumento en la población de satélites, sobre todo en LEO. Esto directamente aumenta el riesgo de que el efecto Kessler proliferé ya que la probabilidad de colisiones incrementa.

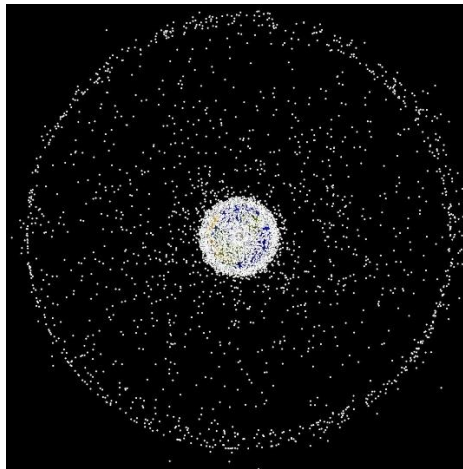


Figura 1-2 Representación que muestra la concentración de basura espacial en GEO (exterior del anillo) y en LEO (puntos cercanos a la Tierra).

La basura espacial es un problema mayor si se considera que fragmentos muy pequeños son difícilmente detectables, a la fecha más de 36500 fragmentos de un tamaño mayor a 10[cm] se hayan catalogados y se les mantiene constante vigilancia y se cree que el número de partículas de tamaños entre 1 y 10 [cm] es de un millón, pero la cantidad estimada de tamaños entre 1[mm] y 1[cm] supera los 330 millones [4]. Estas grandes cantidades pueden no sorprender debido a su tamaño, pero cuando la velocidad a la que viajan entra en la ecuación es que se entiende que una de estas partículas sea capaz de perforar las capas exteriores de las naves espaciales o destruir un satélite con facilidad. La velocidad con la que viajan estos fragmentos es de unos 8[Km/s], pero

en impactos con otros objetos en órbita pueden incluso llegar a  $15\text{[Km/s]}$  [5]. En la Figura 1-3 se evidencia el daño que puede causar un pequeño fragmento de basura espacial al impactar, en este caso, un transbordador espacial, este no solo perforo el casco, sino que al hacerlo se desintegro en pedazos menores los cuales provocaron múltiples daños en otras partes de la nave incluyendo la capa que controla la temperatura interior [6].

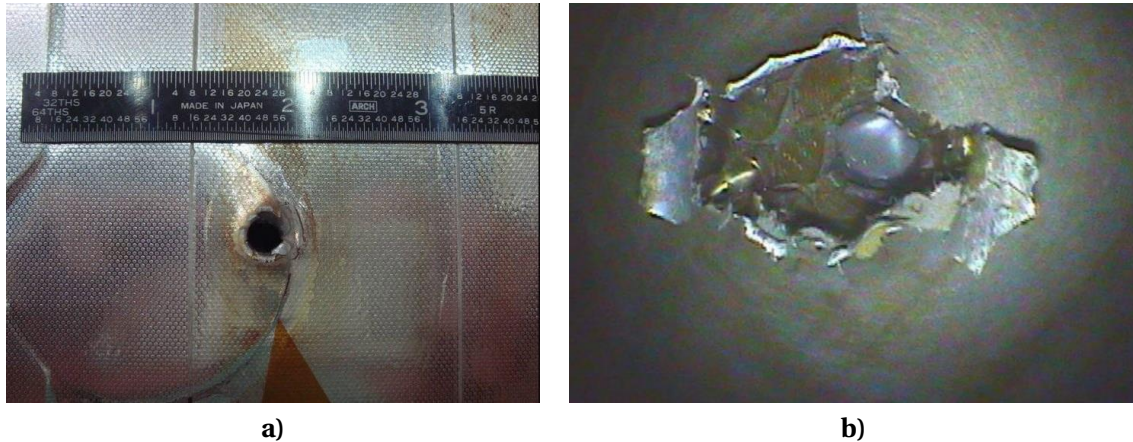


Figura 1-3 Impacto de basura espacial en el radiador del transbordador espacial Endeavour. A izquierda el impacto exterior y a la derecha el agujero fotografiado por el interior.

#### 1.1.4 Colisiones a hiper velocidad

El 2009 el mundo presencio por primera vez una colisión a hiper velocidad entre dos satélites, el primero, perteneciente a la constelación de satélites de telefonía, el Iridium 33 fue impactado por un satélite de comunicaciones ruso el Kosmos 2251 de una tonelada inutilizado desde 1995, el impacto fue a una velocidad relativa de  $11.6\text{[Km/s]}$  y una altura de  $770\text{[Km]}$ . Este evento generaría gran preocupación debido a la gran cantidad de basura espacial que disperso, ya que considerando las características del choque los fragmentos generados son lanzados y se distribuyen en múltiples órbitas, incluso haciendo que algunos trozos de satélite reingresen a la atmosfera poniendo en riesgo la población en general.

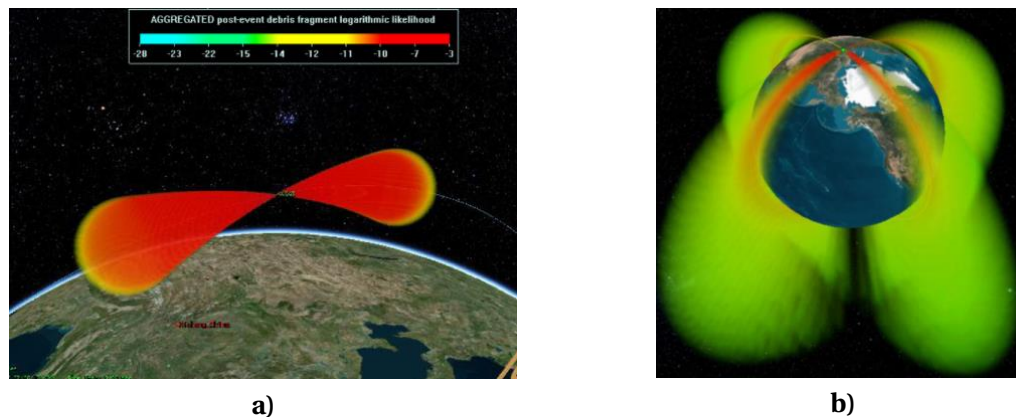


Figura 1-4 Dispersión estimada de los fragmentos producto de la destrucción del Fengyun 1C (Izquierda) en los primeros 4 minutos y del choque Iridium-Kosmos (Derecha) en las primeras 3 horas.

En la Figura 1-4 se puede ver el área de dispersión que alcanzaron estos fragmentos solo en los primeros minutos. Este evento accidental es uno de los que se atribuye la gran cantidad de escombros presente en el espacio, en particular en LEO, pero no es el único. Solo dos años antes, y en una acción totalmente premeditada, el gobierno chino en una prueba de su sistema antisatélite (ASAT) interceptó el Fengyun-1C, un satélite meteorológico de 750 [Kg], haciéndolo estallar y generando basura espacial en el acto. Los restos de ambos eventos hasta el día de hoy son rastreados y siguen siendo catalogados para evitar cualquier incidente que involucre una nueva colisión.

En la Figura 1-5 se presenta una estimación de la tasa anual existente de una colisión entre los pares de objetos activos e inactivos orbitando la Tierra realizada en base a los datos de objetos catalogados hasta el 2019. El cálculo se realizó a razón de 25[Km] y se asumieron radios de impacto combinado (Combined Hard Body Radius - CHBR) para cada par y órbita estudiada. El gráfico muestra la mayor probabilidad de que exista una colisión en LEO, sobre todo en órbitas alrededor de los 500 y 900[Km] de altitud. Se puede notar, además, la gran posibilidad de choques de cuerpos inactivos (en gris) que existe en las clases correspondientes a 775[Km] y 850[Km], las cuales justamente se corresponden con la colisión Iridium-Kosmos y a la prueba del ASAT chino con el Fengyun 1C respectivamente.

Estas probabilidades dan cuenta de la importancia de los sistemas de SSA para mantener un catálogo actualizado de la basura espacial que pueda brindar información a los operarios de satélites con el fin de evitar colisiones, sobre todo en las órbitas LEO mencionadas. Otras técnicas podrían ser hacer lanzamientos de satélites a órbitas no tan saturadas, pero esto tiene implicancias técnicas y económicas involucradas ya que mientras más alta la órbita en la que se busca dejar un satélite más costoso es el lanzamiento. Por otro lado, dejar satélites muy cerca de la superficie también provoca inconvenientes debido a que la gravedad dificulta mantener el satélite en órbita, sin mencionar que la altura es uno de los parámetros que se debe tener en cuenta en el diseño del satélite y por tanto escoger una altura puede no ser una posibilidad. Por otro lado, que los operarios mantengan y compartan la información referente al desplazamiento de sus satélites puede llegar a ser una buena práctica a realizar, aunque esto en la realidad puede no constituir algo beneficioso para los dueños de satélites debido a intereses propios en juego (como podrían ser intereses militares o comerciales) [7].

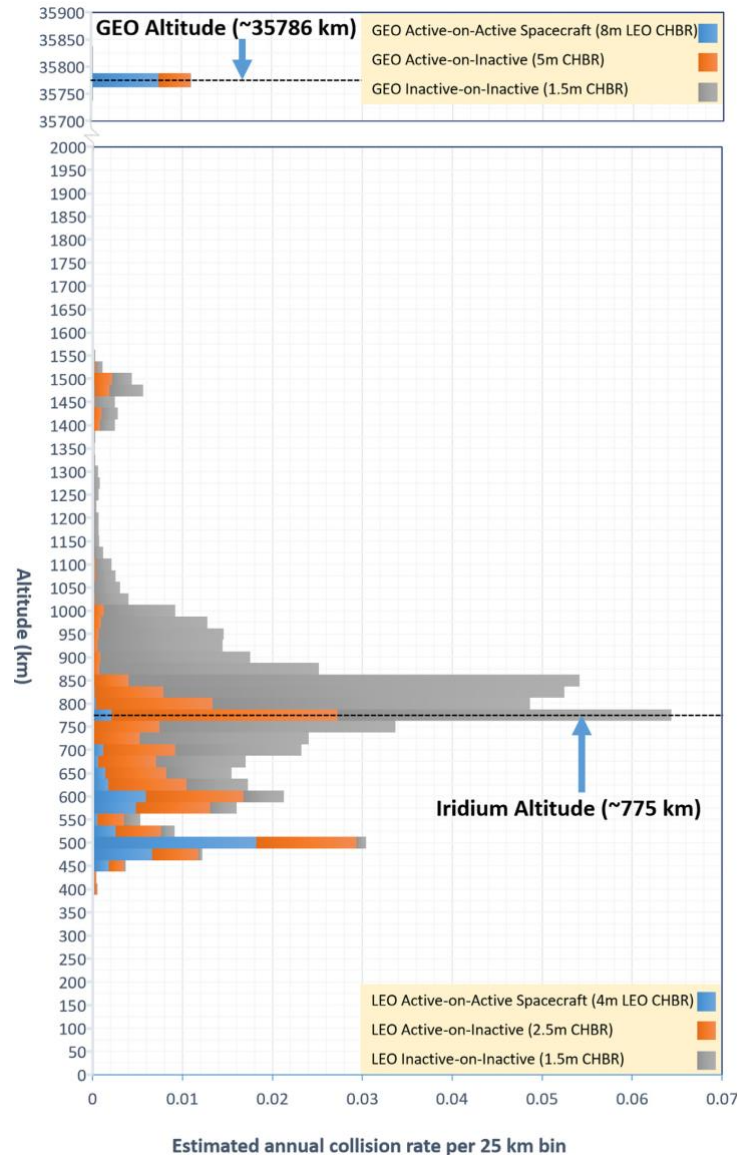


Figura 1-5 Representación de la tasa de colisiones al año en función de la altitud y el tipo de objeto orbitando. Datos actualizados hasta 2019.

### 1.1.5 Contexto social y económico

El problema de la basura espacial lleva un tiempo encima de la mesa y múltiples organizaciones han visibilizado y puesto cartas en el asunto como la ESA en su programa de Vigilancia Espacial (SSA por sus siglas en inglés) o el 18th Space Control Squadron (18SPCS) el cual mantiene y maneja el programa de vigilancia espacial de Estados Unidos. Estos programas mantienen actualizados catálogos de objetos y entregan la información que consiguen de forma gratuita acerca de la mayoría de los satélites que rastrean. Además, satélites actuales son diseñados con su propia forma de manejar la basura generada cuando este queda obsoleto, como la mencionada Starlink en la cual sus satélites orbitan a una altitud relativamente baja para que cuando la vida



útil se agote estos satélites sean atraídos a la atmosfera para que se desintegren en su reingreso, ya que están hechos con materiales pensados para que eso ocurra.

Lamentablemente en general aún existen malas prácticas acerca del cuidado del ambiente espacial, ya que la mayoría de los tratados, guías y estándares acerca de la sostenibilidad y mitigación de la basura espacial no posee regulaciones, monitoreo o refuerzo en sus medidas. Es así como siguen y seguirán ocurriendo casos donde el criterio económico es más importante que este tipo de cuestiones, sobre todo si consideramos que la mayoría de los satélites en órbita actualmente pertenece a organizaciones comerciales en donde posibles malas noticias pueden desplomar las acciones y desincentivar las inversiones.

A modo de ejemplo se puede mencionar el preocupante incidente en abril del 2021 cuando la tripulación de la capsula Crew Dragon que se dirigía a la ISS (International Space Station) tuvo que permanecer alerta debido a que un fragmento de basura espacial se dirigía hacia ellos. Afortunadamente no hubo colisión alguna, aunque horas más tarde portavoces de SpaceX se dirigirían a los medios afirmando que no hubo ninguna amenaza puesto que se trataba de una falsa alarma [8]. Dos lecturas se desprenden de este caso la primera es que, si la amenaza fue real, SpaceX actualmente en los inicios de su gran y codicioso programa espacial no querría que inversores ni posibles clientes conozcan que el peligro es latente, y la segunda es que con un mejor sistema de vigilancia espacial la alerta hubiese podido ser hecha mucho antes que los 20 minutos de este caso o incluso pensar que no hubiese sido necesario debido a la previsión del evento. Casos como este invisibilizan el verdadero riesgo que presenta la basura espacial en el lanzamiento de satélites o misiones espaciales.

Además, existen casos donde operarios debido a razones financieras o la falta de regulaciones decide ignorar y no estimar la seguridad espacial de los satélites, ya que asumir los riesgos que conlleve perder el aparato es mucho más económico que el trabajo que implica realizar un diseño previo y meticuloso con respecto a estos temas. Esto último, se debe sobre todo a la facilidad de lanzar un satélite y los bajos costos de manufactura asociados, sobre todo considerando la redundancia de las constelaciones, la miniaturización y estandarización de lanzamientos (sección 1.1.2).

Consecuentemente, la falta de responsabilidad sigue siendo una de las mayores causantes y generadoras de basura espacial, ejemplos de esto se pueden mencionar la prueba del ASAT indio en marzo de 2019, aunque similar a la prueba china su impacto no tuvo las mismas consecuencias debido a la baja altura relativa en la que se condujo, la de unos 300[Km] . Aunque la mayoría de los escombros pudo reingresar con éxito a la atmosfera, existen algunos que fueron propulsados a alturas incluso de 1000[Km] [9]. En otro contexto el cuerpo del cohete chino Long March 5B que reingreso a la atmósfera de manera descontrolada y no se sabía el lugar exacto donde caerían sus restos en mayo de 2021, da cuenta que en varios de los lanzamientos hoy en día los operarios no se hacen cargo de las etapas de lanzamiento (distintos propulsores que ponen en órbita un objeto), lo que era una práctica común en la época temprana de la era espacial.

## 1.2 Problematización

Los puntos anteriores dan evidencia del latente peligro que significa la basura espacial en las órbitas alrededor de la Tierra y como mantener este problema desatendido podría llegar a impedir nuevas misiones espaciales. El estado de los escombros en órbita ya es preocupante considerando algunos eventos recientes y el descuidado principio de la era espacial donde los satélites obsoletos, trozos de cohetes de lanzamiento y fragmentos de ambos objetos sobre los 700[Km] tenderán a mantenerse en órbita durante cientos de años más. Además, ya que no existen formas factibles aún de reingresar estos objetos a la atmosfera con el fin de que se desintegren la probabilidad de que colisiones ocurran solo es posible que aumente, y más aún con la gran frecuencia con la que satélites son lanzados actualmente.

De las órbitas mencionadas, las LEO figuran como las más atestadas de basura espacial debido, sobre todo a los incidentes ocurridos en la primera década del siglo XXI. Además, junto con ser las más frecuentes para alojar satélites debido al menor costo de lanzamiento asociado a su posición. Por tanto, es en LEO donde más apremia una vigilancia exhaustiva para poder evitar nuevas colisiones.

Por último, los sistemas de vigilancia actualmente son insuficientes para caracterizar la cantidad total de objetos orbitando y el hecho de que vayan a aumentar es un asunto inquietante [7]. Programas codiciosos como Space Surveillance Network de los Estados Unidos busca la colaboración de varias partes del mundo en su esfuerzo para mantener la más completa red de vigilancia espacial, aunque para cubrir la totalidad del cielo aún queda camino por recorrer. Instalaciones comunes para realizar vigilancia espacial involucran tradicionalmente rastreo por radar las cuales tienen buenos desempeños sobre todo en la detección de objetos menores a los 10[cm], pero el costo de implementar una central de este tipo es muy alto y en general se ha visto una tendencia de relevar el trabajo de la vigilancia a los telescopios, pero considerando que estos dispositivos no están pensados para estas tareas es que existen oportunidades para mejorar estos sistemas sobre todo en términos de captura y detección.

Por estos motivos en general es que se busca innovar continuamente en el área de vigilancia espacial con el objetivo de cada vez poseer un ambiente más controlado donde se puedan predecir efectivamente posibles colisiones y que estas no ocurran. En la sección siguiente se hace una propuesta de solución para resolver algunos de los problemas anteriores.

## 1.3 Propuesta

Con el propósito de aportar a los esfuerzos en la vigilancia y detección de objetos en órbita se propone implementar un sistema de vigilancia espacial en base a una cámara de eventos, lo que supone traspasar todas las ventajas de este tipo de sensores a la búsqueda y rastreo de cuerpos en órbita, esto quiere decir la gran resolución temporal, el bajo ancho de banda y la maximización de la información con respecto a los datos guardados en la escena. Esto se hace pensando en las oportunidades que presenta utilizar esta cámara en una escena fundamentalmente “*sparse*” con lo que podría lograrse una aplicación eficiente en términos de procesamiento.

Para este fin será necesario unir una cámara de eventos a un telescopio compatible con el cual pueda hacer mediciones de objetos en LEO, las órbitas de satélites más habitadas y por tanto donde existen más probabilidades de eventos desastrosos. Cuando se menciona la vigilancia de objetos en órbita se hace referencia a la detección de satélites con el fin de verificar su trayectoria, y la posible captura de fragmentos de basura espacial catalogados o no en alguna base de datos. Para esto será necesario comprender y estudiar los estándares para la referenciación de los objetos en órbita, de esta forma se podrán comparar las mediciones adquiridas con catálogos internacionales con el fin de identificar satélites o escombros catalogados, o en un caso ideal encontrar fragmentos de basura espacial que no se encuentren en el catálogo.

Dado lo anterior, es necesario el desarrollo de una plataforma digital que permita el procesamiento y guardado de los datos, esto supone la implementación de un software compatible con la cámara de eventos. Esto es un gran desafío debido a la diferencia sustancial que poseen este tipo de cámaras con las cámaras a cuadros tradicionales, ya que el cambio paradigmático es gigantesco. A priori un par de dificultades que se encontrarán en el camino respecto a la creación del software serán de qué forma procesar el gran flujo de eventos generado por una cámara de este tipo y como limitar o quitar el ruido de disparo.

#### **1.4 Cámaras basadas en eventos**

Con el objetivo de comprender la propuesta de este proyecto se detalla en qué consisten las cámaras basadas en eventos, el principal elemento con el cual se construirá una aplicación de detección de satélites.

Las cámaras basadas en eventos poseen sensores que cambian la forma de conseguir imágenes, estos sensores están inspirados en el comportamiento de la retina biológica (de aquí que otro nombre que se les da es “silicon retina”) y no siguen el paradigma general de las cámaras tradicionales de muestrear siguiendo un proceso discreto en el tiempo, sino que estos píxeles continuamente muestrean la luz incidente cuando se genera un cambio en el nivel de iluminación de la escena, es decir, son sensores de contraste que emiten “eventos” cada vez que existe una diferencia de contraste.

La detección de contraste (CD) se hace por medio de un rápido foto-receptor logarítmico en cada uno de sus píxeles, los cuales además funcionan de forma asincrónica entre sí. Los eventos que se generan pueden ser de tipo ON si el cambio ocurrió desde un nivel menor de iluminación a uno mayor o de tipo OFF si el cambio fue a la inversa y esto ocurre solo si la intensidad supera cierto umbral que puede ser ajustado por el operario Figura 1-6.

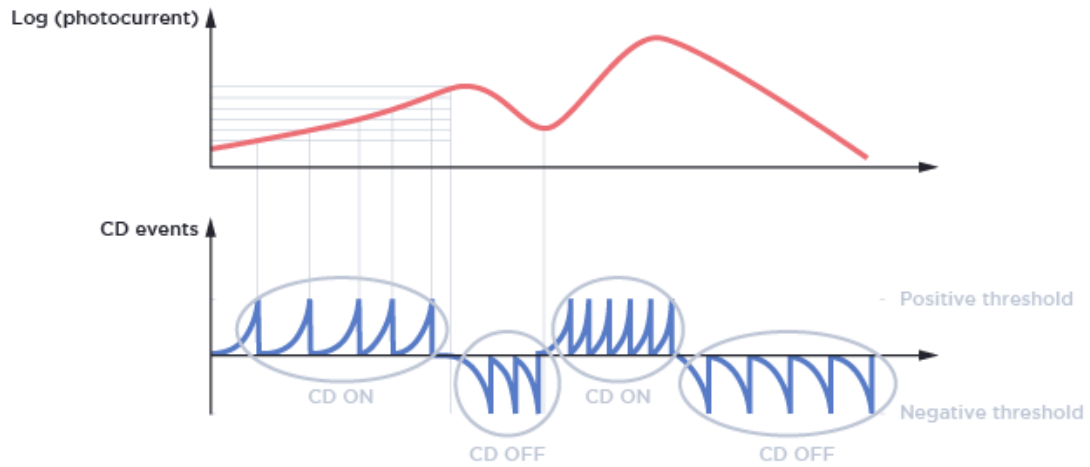


Figura 1-6 Funcionamiento de la captura de eventos. Arriba se ve el nivel de luz (o logaritmo de la fotocorriente) en rojo, cada vez que se excede un cierto umbral (sea incrementando o decreciendo el nivel de luz) un evento se genera ON u OFF (en azul en la imagen de abajo).

En contraste con las cámaras convencionales que capturan imágenes completas utilizando la total cantidad de los píxeles en sus sensores a una frecuencia fija especificada por un reloj interno, las cámaras basadas en eventos capturan la dinámica de la escena de forma asíncrona e independiente en cada píxel, luego estos eventos son etiquetados con el siguiente formato:

- Un arreglo  $(x, y)$  que representa la posición del píxel que genero el evento en el sensor.
- Un valor de polaridad  $[-1, 1]$ , -1 para un evento OFF y 1 para un evento ON.
- Una marca de tiempo expresada en  $\mu s$  (microsegundos).

Un flujo de eventos, que es lo que se suele encontrar en los archivos RAW obtenidos de estas cámaras, suelen contener n-cantidad de eventos uno tras otro con el formato anteriormente mencionado, la forma matemática de expresar este flujo de eventos está dada por la siguiente ecuación donde  $i$  corresponde al  $i$ -ésimo evento del flujo:

$$e_i = (x_i, y_i, t_i, p_i) \quad (1-1)$$

Para comprender la diferencia sustancial entre una cámara convencional y una cámara basada en eventos en la Figura 1-7 se muestran las salidas de ambas cámaras ante un mismo impulso. En este caso un disco rotando en su eje se ve como una cantidad finita de cuadros a la salida de una cámara estándar, imágenes que si no se ven en secuencia pareciesen no tener ninguna cohesión, en cambio, para la cámara de eventos el mismo movimiento es representado en el tiempo como un flujo continuo de eventos. Esto también explica por qué en las cámaras de eventos se preserva de mejor manera la dinámica de la escena capturada.

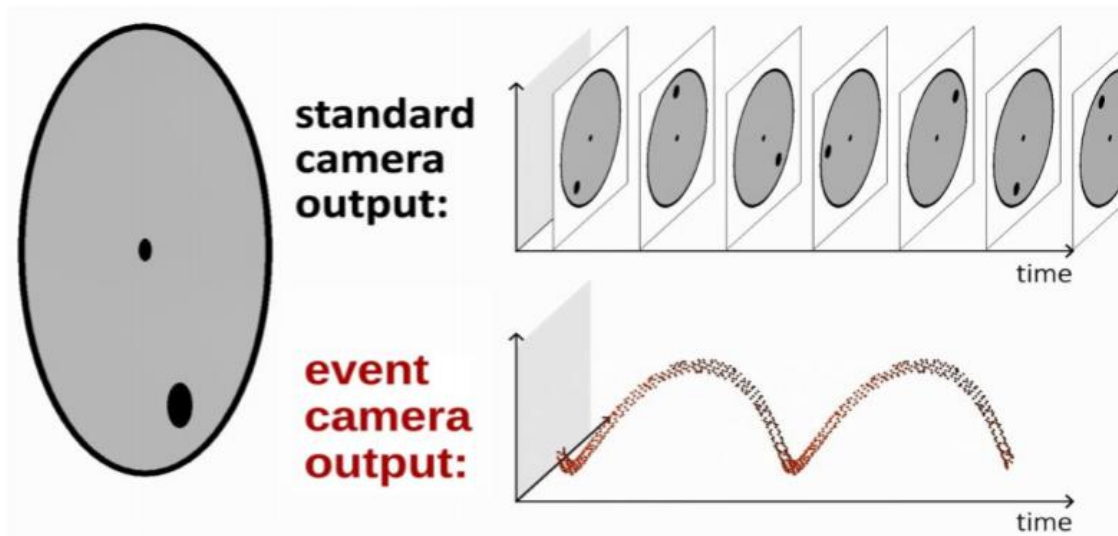


Figura 1-7 Salida de una cámara tradicional a cuadros (arriba) y una cámara de eventos (abajo) ante un mismo estímulo, un disco rotando con un punto negro dibujado en él.

Las principales ventajas que en las cámaras basadas en eventos podemos encontrar:

- Gran resolución temporal, la cual corresponde a  $1\mu s$  que no solo supera en gran medida a las cámaras basadas en cuadros de alta rapidez ( $\sim 1000$  cuadros por segundo,  $1ms$ ) pudiendo captar movimientos mucho más veloces, sino que también, debido a la característica asincrónica de sus píxeles y que están captando continuamente diferencias en la escena, es posible capturar la dinámica de la escena de una forma más fiable, sin la discretización que desacopla el movimiento al separar en cuadros.
- Gran rango dinámico, debido a que los foto-receptores de los píxeles operan en escala logarítmica es posible para estas cámaras trabajar en condiciones extremas de iluminación desde a plena luz del día hasta con una muy baja cantidad de luz. Superan ampliamente los 60[db] que poseen las cámaras convencionales llegando a tener más de 120[db] algunos modelos.
- Baja latencia, debido a que cada píxel se comunica independientemente y no existe la necesidad de esperar a un tiempo global de exposición, cuando ocurre el evento es transmitido, lo que genera tiempos muy bajos de latencia, menor al rango de milisegundos.
- Bajo consumo, siguiendo la misma idea anterior como la activación de cada píxel ocurre cuando existe un cambio de brillo y, por tanto, solo se activa un píxel cuando ocurre esto, la potencia consumida es muy baja, menor al orden de 10[mW].
- Bajo ancho de banda, ya que los datos capturados son esencialmente “sparse” (alta cantidad de información en relativamente pocos datos) y considerando que no captura simultáneamente en todos los píxeles, entonces se tiene que los archivos conseguidos y transmitidos en la captura tienen una cantidad menor de datos, pero más información ya

que los datos redundantes son excluidos (zonas de la imagen que no hay cambios de brillo, es decir, zonas donde no ocurre movimiento).

Existen desafíos en el uso de estas cámaras debido principalmente a que los eventos se capturan de forma considerablemente distinta a como se capturan los cuadros, sobre todo por el carácter asincrónico de sus píxeles y que estos no guardan valores acerca de la intensidad, solo sus cambios. Es por lo que el principal reto en ocupar cámaras basadas en eventos actualmente es diseñar métodos novedosos que puedan obtener los datos y extraer la información con el objetivo de desbloquear el potencial de estas cámaras. Se pueden describir específicamente tres grandes barreras a superar:

- Diferencia espaciotemporal de la salida: Eventos y cuadros son fundamentalmente distintas representaciones, los primeros son asíncronos y espacialmente “sparse” en cuanto los segundos son síncronos y densos, esto hace que los algoritmos y diseñados para el procesamiento de cuadros no sean aplicables a la representación por eventos.
- Diferencia de la magnitud medida: En contraste a la información monocromática que una cámara convencional entrega (valores absolutos de iluminación), los cambios en los niveles de luz que las cámaras de eventos miden no solo depende de la polaridad del evento, sino del valor pasado del punto espacial y el movimiento relativo entre la escena y la cámara.

Ruido y efectos dinámicos: Todos los sensores tienen que lidiar con el ruido (ruido de disparo, en este caso) y este no es la excepción, pero el problema es mayor sobre todo porque aún no ha sido completamente caracterizado [10].

#### **1.4.1 Cámara de eventos : Prophesee Evaluation Kit**

El modelo de cámara utilizado en este proyecto corresponde al kit de evaluación de Prophesee, este kit entrega una solución completa para la visualización en tiempo real y captura de eventos. El sensor CSD3MVCD que captura los cambios de contraste en la escena viene asegurado en una carcasa metálica la cual posee una montura CS donde se colocan los lentes, una interfaz USB 3.0 con la cual el kit realiza completamente la tarea de empaquetar y enviar el flujo de datos, además en la parte inferior posee un espacio donde se puede atornillar un pequeño trípode. Una imagen de la cámara y la carcasa se pueden ver en la Figura 1-8, en tanto, las especificaciones técnicas del sensor se muestran en la Tabla 1-1.



Figura 1-8 Fotografía del Kit de evaluación de Prophesee (a la izquierda) e imagen superior de la cámara mostrando los pines (a la derecha).

Tabla 1-1 Especificaciones del sensor modelo CSD3MVCD

Características del sensor	
Tamaño del arreglo	640 x 480
Tamaño pixel	15 $\mu m$
Tamaño de sensor	3/4"
Latencia promedio	200 $\mu s$
Resolución temporal	1 $\mu s$

De manera análoga, en la parte superior de la carcasa la cámara posee 7 pines con los cuales el kit soporta la captura de triggers externos los que se empaquetan de forma nativa al flujo de eventos, también por medio de estos pines se puede generar una salida con pulso y periodo programables o utilizar dos de estos como I2C master, la función y orden de estos pines se muestra en la Tabla 1-2.

Tabla 1-2 Orden y función de pines del kit de evaluación

Interfaz para conexiones externas	
#Pin	Función
0	Trigger Out / Ext Sync CLK Out (1.8V)
1	GND
2	Ext Sync CLK In (1.8V)
3	GND
4	Trigger In P / opto - coupled
5	Trigger In N / opto - coupled
6	I2C (SDA)
7	I2C (SCL)

A modo de discusión se puede comparar las grandes diferencias entre los sensores CCD convencionales que usan las cámaras a cuadros y el sensor que posee este kit de evaluación, por ejemplo, una cámara de 12 Megapíxeles puede poseer tamaños de píxel 10 veces menores que esta cámara (bajo los  $2\ \mu m$ ) y cuestan en términos monetarios mucho menos, esto revela la gran problemática en disponibilidad de este tipo de cámaras y su temprano estado de exploración y producción. Este último análisis es realizando una comparación con respecto a la capacidad espacial de las cámaras, pero lo cierto es que en términos temporales el sensor de eventos supera por mucho a las cámaras de cuadros, las cuales algunas alcanzan unos pares de miles de cuadros, es decir, en el rango de los  $ms$ , por lo que las cámaras de eventos superan ese valor 1000 veces.

### 1.4.2 Parámetros internos

La captura de eventos con la cámara es una tarea compleja, debido al comportamiento ondulatorio de la luz pequeñas intensidades pueden generar eventos aleatorios en zonas de la escena donde no hay cambios aparentes, esto se repercute en que las grabaciones realizadas por lo general siempre tendrán mucho ruido. El ajuste principal en estos tipos de cámara, dejando de lado el ajuste óptico que se realiza con la elección de uno u otro lente, es el afinamiento que se le puede dar a ciertos niveles de voltajes internos de la cámara para controlar la entrada en el sensor. Los valores de dichos voltajes pueden ser manipulados por software y son comúnmente llamados “*bias*”. En la Figura 1-9 se puede observar el diagrama de bloques donde los “*bias*” corresponden a conversores digital-análogo, luego se describe la principal función de cada “*bias*” y en algunos casos se entregan relaciones dadas por el fabricante.

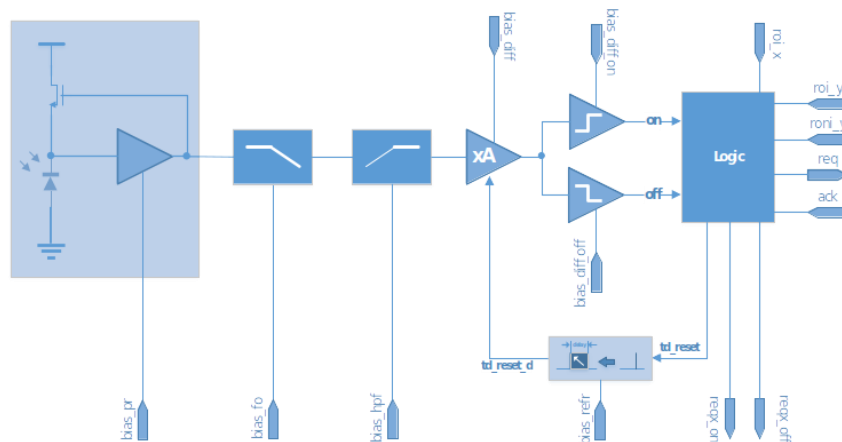


Figura 1-9 Diagrama de bloques de un píxel de la cámara de eventos.

- **Bias\_pr:** Controla la entrada del fotorreceptor, es decir, modifica el punto de operación del píxel. Tiene un gran impacto en la corriente consumida y el fabricante recomienda que solo bajo condiciones específicas se cambie el nivel de operación con el que viene configurado.
- **Bias\_fo:** Controla la frecuencia de corte del filtro pasa-bajos. Mientras más alto sea menor será la frecuencia de corte, esto genera que se reduzca la cantidad de ruido integrado a la medición y por tanto el ancho de banda, esto no significa necesariamente que un valor



mayor de **bias\_fo** es mejor en todos los casos, puesto que muchas veces además de quitar ruido reducirá la cantidad de eventos que entregan información de la escena. Una de las funciones que tiene es la de hacer desaparecer el “flickering” de aparatos electrónicos como LEDs, lo que es muy útil en casos donde la escena es iluminada con luz artificial. En la Figura 1-10 se muestra cómo se comporta la frecuencia de corte del filtro a medida que varía el valor de este bias mientras los demás se mantienen en su valor por defecto.

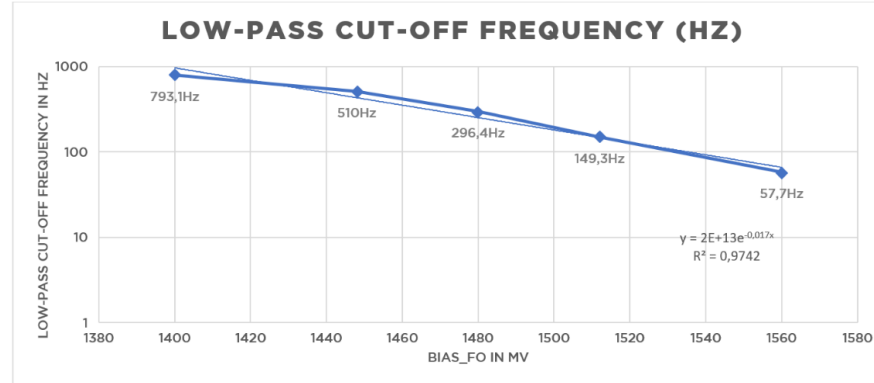


Figura 1-10 Comportamiento de la frecuencia de corte del filtro pasa-bajos al variar los valores de **bias\_fo**, para valores por defecto en los demás bias.

- **Bias\_hpf:** Controla la frecuencia de corte del filtro pasa-altos. La función que posee es filtrar eventos de carácter lento de la escena. Mientras más bajo es más alto es la frecuencia de corte del filtro. En la Figura 1-11 se muestra cómo cambia la frecuencia de corte del filtro pasa-altos mientras se varía el valor de este bias con los demás en su valor por defecto.

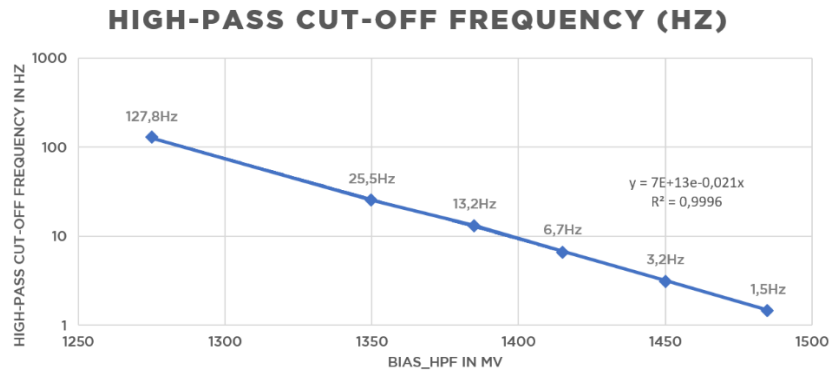


Figura 1-11 Comportamiento de la frecuencia de corte del filtro pasa-altos al variar los valores de **bias\_hpf**, para valores por defecto en los demás bias.

- **Bias\_diff:** Controla el nivel de referencia para la determinación de los umbrales en la generación de eventos ON y OFF. El fabricante recomienda dejar este valor como 300[mV], ya que está referenciado a tierra, aunque se puede incrementar este valor si se tiene como objetivo llegar a bajos niveles de latencia (10  $\mu s$ ).
- **Bias\_diff\_on/ Bias\_diff\_off:** Controlan el nivel del comparador para generar eventos ON y OFF respectivamente. El valor efectivo de los umbrales está dado por las diferencias:

$$umbral_{pos} = Bias\_diff\_on - Bias\_diff$$

$$umbral_{neg} = Bias\_diff - Bias\_diff\_off$$

Mientras más baja la diferencia, más pequeño será el umbral que necesita el píxel para generar un nuevo evento y por tanto más sensible al ruido y a los cambios en iluminación de la escena, comportamiento visto en la Figura 1-6. Por regla general se pide que la diferencia mínima para ambos sea de unos 50[mV], ya que bajo este valor el sensor se satura al crear eventos en la mayoría de los píxeles.

- **Bias\_refr:** Controla el periodo refractario en el cual el píxel permanece en “reset” luego que un evento se ha generado. Incrementar este valor reduce la cantidad de eventos generados. En la Figura 1-12 se observa cómo se comporta este valor, valores en el orden del microsegundo debería ser considerado como el mínimo a ocupar.

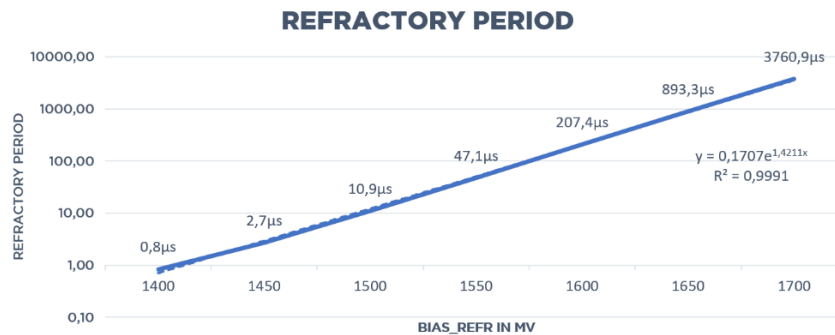


Figura 1-12 Comportamiento del periodo refractario del píxel al variar los valores de bias\_refr, para valores por defecto de los demás bias.

La afinación de cada uno de estos “bias” no es una tarea trivial y depende de la interrelación entre cada uno de estos, pero también de la escena que se quiera capturar, las condiciones ambientales y el objetivo de la captura. Actualmente no existen en la literatura estudios que traten de resolver el problema de ajustar de manera óptima estos valores.

### 1.4.3 Formato de archivos .RAW

Esta cámara genera eventos en el formato conocido como Event Type 2.0 (EVT 2.0), en este formato de información los eventos creados son almacenados en 32 bits de memoria. Soporta variados tipos de eventos, los cuales son diferenciados por los 4 bits más significativos, principalmente 3 son los tipos de eventos indispensables, estos se detallan a continuación.

#### Event Time High

Este tipo de evento no corresponde a un evento como tal, pero se guarda de igual forma que uno de estos. La codificación del tiempo en el formato EVT 2.0 posee un tamaño especial de 34 bits con una resolución en  $\mu s$ , es decir,  $2^{34} \mu s \cong 4 \text{ Hrs}$  y  $46 \text{ mins}$ . Con el motivo de minimizar la cantidad de información guardada en cada evento es que los 34 bits de una marca de tiempo están divididos en dos secciones, la primera sección de los 28 bits más significativos se utiliza para crear un tipo de evento llamado Event Time High, el identificador de 4 bits de este tipo de eventos es el ‘1000’. La otra sección correspondiente a los 6 bits menos significativos es incluida en cada uno

de los eventos ON y OFF. El formato de los eventos Event Time High se puede observar en la Tabla 1-3.

Tabla 1-3 Formato de codificación Event Time High

Rango de bits	Cantidad de bits	Nombre	Descripción
31 – 28	4	Tipo	ID de evento (Event time high) '1000'
27 - 0	28	Marca de tiempo	28 bits más significativos de la marca

En la práctica, se ha observado que los eventos Event Time High son generados periódicamente, es decir, cada vez que se cumplen  $64[\mu s]$  correspondientes a los 6 bits menos significativos colocados en cada evento ON/OFF, no solo uno sino 4 nuevos Event Time High son creados y por tanto guardados en el archivo RAW. En resumen, cada  $64[\mu s]$  4 eventos de este tipo son transmitidos lo que es una carga constante para el sistema de transmisión y la memoria del archivo, esto puede otorgar robustez al sistema de codificación, pero es sumamente ineficiente en términos de memoria almacenada.

### Eventos ON y OFF

Estos dos tipos de eventos señalan cuando ha ocurrido un cambio de contraste positivo (ON) o negativo (OFF) en algún píxel, es decir, cuando ha ocurrido un evento. La diferencia entre uno y otro son los 4 bits más significativos que se utilizan para determinar el tipo de evento, para ON esta palabra binaria es '0001' y para los eventos OFF es '0000'. Luego los siguientes 6 bits corresponden a los 6 bits menos significativos de la marca de tiempo (como se mencionó anteriormente). Por último, los siguientes 11 bits y 11 bits son para la posición en x e y del píxel donde se generó el evento, esta cantidad indica que este tipo de codificación soporta hasta un sensor con una resolución de  $2048 \times 2048$  pixeles. En la Tabla 1-4 se puede observar el orden de los bits en estos tipos de evento.

Tabla 1-4 Formato de codificación para eventos ON/OFF

Rango de bits	Cantidad de bits	Nombre	Descripción
31 – 28	4	Tipo	Tipo de evento ('0001' ON o '0000' OFF)
27 – 22	6	Marca de	6 bits menos significativos de la marca
21 – 11	11	X	Coordenada X del píxel
10 - 0	11	Y	Coordenada Y del píxel

En la Figura 1-13 se presenta un esquema que ilustra el orden que llevan los 3 principales tipos de eventos en el formato EVT 2.0.

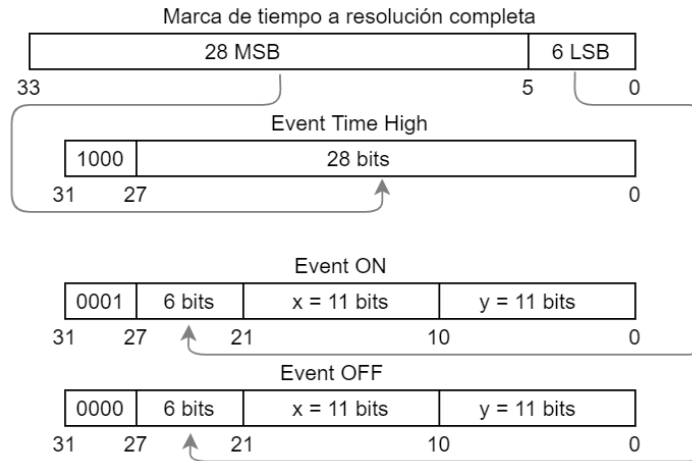


Figura 1-13 Formato y orden de los eventos ON/OFF y Event Time High

Existen otros tipos de eventos con su propio orden de 32 bits, eventos incluyen gatillos externos o eventos específicos detallados para cada vendedor, la forma de identificación se hace de la misma forma, reconocer el evento por sus 4 bits más significativos. Este proyecto no involucra el uso de ninguno de ellos y por tanto se omite detallar su funcionamiento.

Cabe mencionar que cada archivo RAW posee una cabecera de 6 líneas donde se encuentran en orden la fecha y hora de la captura del flujo de eventos, la versión del firmware, el nombre del fabricante, nombre del plugin utilizado, el serial de la cámara y por último su número identificador. Un ejemplo se puede ver en el Listado 1-1.

```

1  % Date 2021-04-01 14:19:26
2  % firmware_version 3.5.0
3  % integrator_name Prophesee
4  % plugin_name _plugin_gen31_fx3
5  % serial_number 00001328
6  % system_ID 28

```

Listado 1-1 Ejemplo de cabecera de un archivo RAW en formato EVT 2.0.

## 2 Estado del arte

La mayoría del trabajo realizado para conseguir aplicaciones y soluciones con cámaras de eventos envuelve el desarrollo de sistemas capaces de procesar eficientemente el flujo de eventos y crear algoritmos que permitan la detección y visualización correcta de los objetos de interés, el área de vigilancia espacial también es el caso, por tanto, a continuación se describe el estado del arte investigado, orientado principalmente al preprocesamiento de los archivos RAW y a la detección de objetos en órbita con una cámara de eventos.

El proceso que a continuación se describe sigue el trabajo hecho por Cohen et al., tanto en [11] como en [12]. En el primero de estos trabajos se demuestra que con la cámara de eventos es posible la grabación de objetos en LEO y GEO e incluso se hacen pruebas positivas del uso de la cámara para estos motivos en el día. En el segundo de estos trabajos se profundiza en la creación de un algoritmo de rastreo de objetos para la mejor visualización de estos. De este último trabajo es del que se trata principalmente la siguiente sección debido a ser una versión más avanzada del primero.

### 2.1.1 Pre-procesamiento del flujo de eventos

Como se mencionó en secciones anteriores los eventos obtenidos poseen 4 magnitudes tal y como fue descrito en (1-1), de donde se sabe que  $i$  es el índice del evento y que la polaridad  $p$  puede tomar valores de 1 o -1. Las demás magnitudes corresponden a la posición física del píxel en el sensor  $(x, y)$  y a la marca de tiempo  $t_i$  la cual tiene una resolución de  $1[\mu s]$ .

Con el fin de no perder la riqueza temporal de los eventos es que se utiliza un método de representación conocido como "exponential decaying time surface" (Superficie de tiempo exponencialmente decreciente). Esta representación asigna un valor a cada píxel correspondiente a su última activación, además se ocupa una exponencial decreciente con el fin de enfatizar los eventos recientes sobre los pasados.

$$S_i(x, y) = e^{(T_i(x, y) - t_i) / \tau} \quad (2-1)$$

En (2-1)  $T_i(x)$  representa la matriz que contiene las marcas de tiempo de los eventos más recientes ocurridos en cada píxel  $(x, y)$  al  $i$ -ésimo evento,  $S_i(x, y)$  es la superficie de tiempo que posee una constante de tiempo  $\tau$  (0.4 segundos en [12]). Para replicar de mejor forma una cantidad mayor

de ambientes de observación, es que en la bibliografía estudiada se ha decidido por procesar cada polaridad por separado, es decir, la superficie de tiempo recibe eventos de solo una polaridad, esto no se suele hacer en este tipo de algoritmos ya que desacopla la información que brinda cada evento y esto puede generar peores desempeños, pero se toma el riesgo con el objetivo de generar un algoritmo de detección y rastreo más robusto sustentado por la característica “sparse” de la entrada.

Luego de conseguir la superficie de tiempo se escoge un ROI (región de interés) de tamaño  $D \times D$  alrededor del evento  $e_i$  que esta siendo procesado.

$$ROI_i = S_i(x_i + u_x, y_i + u_y) \quad \text{donde } u_x = [-R:R], u_y = [-R:R] \quad (2-2)$$

$$\sqrt{x^2 + y^2} \leq R, \quad \forall x \in u_x, \quad \forall y \in u_y \quad (2-3)$$

De las ecuaciones (2-2) y (2-3) se nota que el ROI corresponde a un disco centrado en el evento  $e_i$  que contiene los valores de la superficie de tiempo  $S_i$  de todos los pixeles a una distancia de  $R$  en un tiempo  $t_i$ . Luego, el ROI es procesado en una prueba de activación determinada por:

$$L < \sum_{x=x_i-R}^{x_i+R} \sum_{y=y_i-R}^{y_i+R} T_i(x, y) > \Phi \quad (2-4)$$

donde  $\Phi$  es el intervalo de tiempo de activación para los eventos y  $L$  es el numero de pixeles activados requeridos. Entonces, si el numero de pixeles activados recientemente en la superficie de tiempo en el disco  $R$  alrededor de  $e_i$  es mayor a  $L$ , el ROI es aceptado. Para la prueba cuan reciente es un evento se mide de acuerdo si el píxel ha recibido eventos en los últimos  $\Phi$  segundos. Esta etapa funciona efectivamente como un filtro de ruido, lo cual es esencial para las tareas relacionadas a la observación espacial debido a la baja SNR.

### 2.1.2 Detección de objetos

En la siguiente etapa de procesamiento, un ROI aceptado de la etapa anterior es convertido en una activación angular  $A$ , esto se realiza al multiplicar dicho ROI por cada una de las plantillas creadas de medias barras rotadas como se muestra en la Figura 2-1d). Las plantillas consisten en barras que miden  $R + 1$  y 3 pixeles de ancho. Fuera de la barra cada píxel tiene una magnitud de -0.2 para penalizar cualquier activación de ruido. Entonces, el ROI valido es vectorizado con el propósito de ser multiplicado por  $N$  plantillas, en la práctica el ROI de tamaño  $D \times D$  pasa a un vector de tamaño  $D^2 \times 1$  y es multiplicada por una tabla de plantillas (“Look Up Table” o  $LUT$ ) de tamaño  $N \times D^2$ .

$$A_i = LUT \cdot vec(ROI_i) \quad (2-5)$$

El paso anterior es costoso en términos computacionales ya que, en contraste a etapas posteriores, esta operación se hace fundamentalmente sobre eventos de la cámara y por tanto depende del número de eventos procesados.

El resultado de la activación angular  $A$  es comparado con un umbral de detección  $\Psi$ , donde si ningún elemento sobrepasa el valor  $\Psi$ , la prueba falla. Por otro lado, si uno o más elementos son mayores al valor de  $\Psi$ , entonces la variable  $m$ , la cual corresponde al índice del mayor de los elementos activados es pasado a la siguiente etapa de procesamiento junto con un vector  $\Gamma$  que contiene los índices de los elementos de  $A$  que son mayores a  $\Psi$ .

El valor del umbral  $\Psi$ , puede ser escogido como un valor estático o como fue escogido en la bibliografía como un valor dinámico  $W$  igual a la diferencia entre el valor máximo y mínimo de  $A_i$ . La elección del segundo sobre el primero provoca mayor robustez frente al ruido a cambio de cargar más esta etapa de procesamiento.

$$\Psi_i = W(\max(A_i) - \min(A_i)) \quad (2-6)$$

Esta última etapa es útil removiendo eventos que no corresponden a la trayectoria de algún objeto (la cual constituye una línea en superficies de tiempo sucesivas), pero no le es posible distinguir eventos que se encuentren cerca o en la línea y aquellos en la punta de la línea. Para conseguir extraer estos últimos eventos se realiza una prueba de distribución unimodal, en la cual se mide la distancia angular entre el valor máximo en  $A$  y el promedio angular de todos los elementos sobre el umbral  $\Psi_i$ . La prueba unimodal toma como entrada los índices en  $\Gamma$  para calcular el promedio circular  $q_i$ , de esta forma se comprueba la distancia  $\zeta_i$  entre  $q_i$  y  $m_i$ , si dicha distancia es menor a un parámetro  $\delta$  entonces la prueba se comprueba como una detección. Si el evento  $e_i$  pasa la prueba unimodal se añade como información adicional la orientación  $\theta_i = q_i$  y resulta en un evento detectado de salida  $f_i$ .

$$f_i = (x_i, y_i, \theta_i, t_i, p_i) \quad (2-7)$$

La Figura 2-1 muestra el diagrama de bloques (a) de las distintas etapas anteriormente descritas, además entrega ejemplos del procesamiento para un evento  $e_i$  con polaridad positiva y otro con polaridad negativa (b- f). Además, en la Figura 2-2 se muestra un ejemplo de la detección realizada para la grabación del objeto SL-8 R/B correspondiente a un cuerpo de cohete. En este se pueden observar las diferencias en los flujos detectado entre eventos ON y OFF, por otro lado, observando el panel c) con las detecciones angulares se puede notar como el objeto de baja velocidad genera una distribución en todas las orientaciones  $\theta \in [0, 2\pi]$ , en cuanto objetos de alta velocidad generan detecciones discretas como la marcada con el rectángulo negro en a)-b) y luego investigada con un zoom. Por último, en el acercamiento se puede ver que la detección deja una sombra la cual proviene de falsas detecciones, estas ocurren principalmente con los eventos OFF y en la observación de objetos de alta velocidad, además tienen la misma probabilidad de aparecer con una orientación en contra o en la misma dirección que del objeto. Estas falsas detecciones que se gatillan debido a eventos tardíos en la cola de los objetos observados pueden ser efectivamente reducidos con un algoritmo de rastreo basado en eventos.



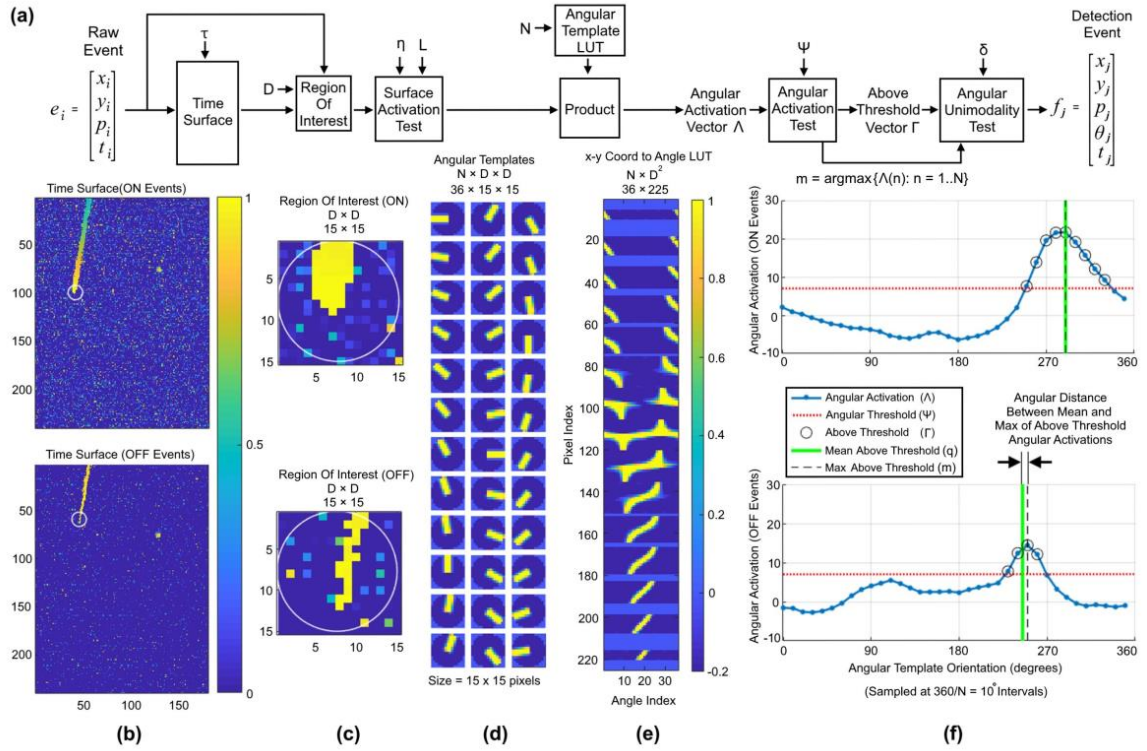


Figura 2-1 **a)** Diagrama de bloques del pre-procesamiento y la detección de objetos basados en un flujo de eventos. **b)** Superficies de tiempo exponencialmente decrecientes para eventos ON y eventos OFF. **c)** Región de interés (ROI) para un evento determinado. **d)** Plantillas de las rotaciones posibles en la detección de objetos. **e)** Transformación de las plantillas en una “Look up table” (LUT) de tamaño  $N \times D^2$ . **f)** Proceso grafico de la prueba de distribución unimodal.

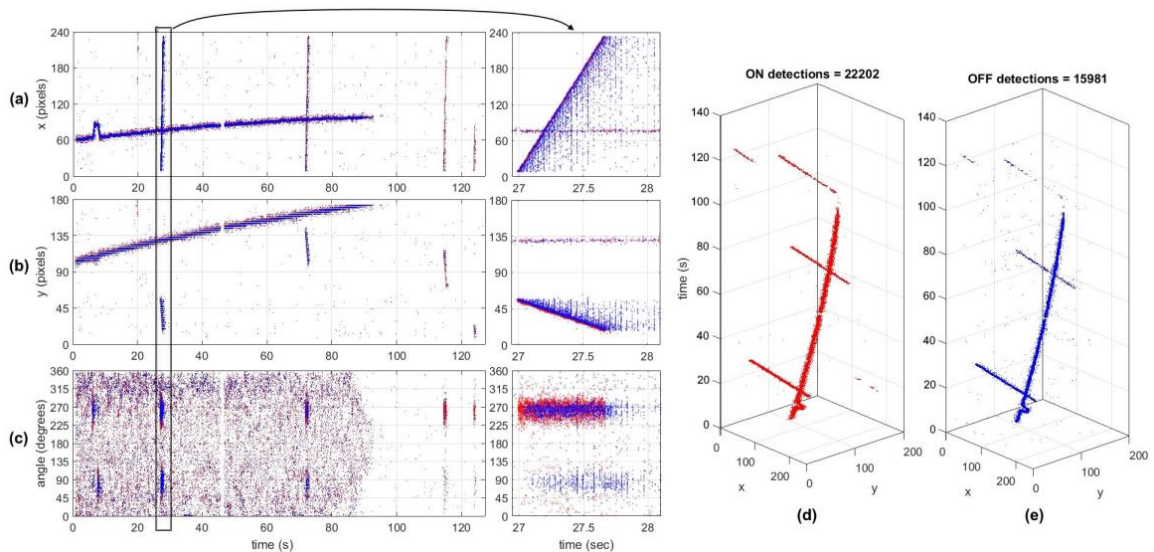


Figura 2-2 Detección de objetos con eventos en la captura de SL-8 R/B. **a), b)** y **c)** muestran las posiciones en x e y, además de la orientación angular de los eventos detectados para eventos ON (en rojo) y eventos OFF (en azul). **d)** y **e)** muestran las representaciones 3D para los eventos ON y OFF respectivamente.



### 2.1.3 Rastreo basado en eventos

El método de rastreo utilizado en la bibliografía genera, actualiza y remueve continuamente hipótesis siguiendo a los eventos aumentados de la salida en la última etapa del detector. Cada objeto rastreado es modelado como una neurona que contiene un potencial de membrana, el cual disminuye en el tiempo y es elevado solo con detecciones provenientes de eventos  $f_i$ . El potencial de membrana representa, por tanto, el nivel de observaciones recientes del objeto. Si el objeto alcanza el potencial de activación  $M_A$  el objeto es activado, de igual forma si el potencial baja a cero el objeto es eliminado.

$$M_k^{(o)} = \begin{cases} M_{k-1}^{(o)} - \gamma(t_j - t_{k-1}) + 1, & \text{si } f_j \text{ es asignado a } H_k^{(o)} \\ 0, & \text{si } M_{k-1}^{(o)} = 0 \\ M_{k-1}^{(o)} - \gamma(t_j - t_{k-1}), & \text{en cualquier otro caso} \end{cases} \quad (2-8)$$

donde  $H_k^{(o)}$  es la k-ésima observación del o-ésimo objeto,  $M_k^{(o)}$  es el potencial de membrana de  $H_k^{(o)}$  a  $t_k$  y  $\gamma$  es el factor de decaimiento del potencial de membrana. Si  $M_k^{(o)}$  alcanza el umbral de activación  $M_A$  el objeto  $H_k^{(o)}$  es considerado un verdadero objeto de rastreo, una variable  $A_k^{(o)}$  sigue el nivel de  $M_k^{(o)}$  hasta que supera el umbral, en cuyo caso  $A_k^{(o)}$  se vuelve 1 y permanece en dicho valor indicando la activación del objeto incluso si el potencial de membrana disminuye.

$$A_k^{(o)} = \begin{cases} M_k^{(o)} / M_A, & \text{si } M_k^{(o)} < M_A \text{ y } A_k^{(o)} < 1 \\ 1, & \text{en cualquier otro caso} \end{cases} \quad (2-9)$$

Para la predicción se define a partir de  $f_j$  (2-7),  $z_j$  como un vector que contiene solo la posición y la orientación angular de la forma:

$$z_j = (x_j, y_j, \theta_j) \quad (2-10)$$

Así se define la posición y velocidad de cualquier objeto activo “o” en la k-ésima observación:

$$H_k^{(o)} = [\hat{z}_k, \beta_k, p_k, t_k] \quad (2-11)$$

donde  $\hat{z}_k = [\hat{x}_k, \hat{y}_k, \hat{\theta}_k]$  y  $\beta_k = [d\hat{x}_k/dt_k, d\hat{y}_k/dt_k, d\hat{\theta}_k/dt_k]$  estimados mediante la ecuación (2-12), con  $\beta_{k-1} = [dx/dt_{k-1}, dy/dt_{k-1}, d\theta/dt_{k-1}]$ .

$$[\hat{x}_k, \hat{y}_k, \hat{\theta}_k] = [\hat{x}_{k-1}, \hat{y}_{k-1}, \hat{\theta}_{k-1}] + \beta_{k-1}(t_j - t_{k-1}) \quad (2-12)$$

Cuando se estima la distancia para una nueva detección para cada objeto  $H_k^{(o)}$ , el peso para la distancia angular relativa a la distancia en x e y, es proporcional a la velocidad anterior de cada objeto y al valor de activación  $A_k^{(o)}$ , esta relación se describe en la ecuación (2-13).

$$\omega_k^{(o)} = V \left( \sqrt{(dx/dt_k)^2 + (dy/dt_k)^2} \right) \cdot A_k^{(o)} \quad (2-13)$$

Donde  $V$  es un factor de escala que en la fuente estudiada se usa como  $V = 0.1$ . De esta forma la distancia entre el nuevo evento detectado  $f_j$  y la predicción para cada objeto activo  $H_k^{(o)}$  en  $t_j$  es definido como:

$$d_k^{(o)} = \sqrt{(x_j - \hat{x}_k^{(o)})^2 + (y_j - \hat{y}_k^{(o)})^2 + \omega_k^{(o)}(\theta_j \ominus \hat{\theta}_k^{(o)})^2} < d_{max} \quad (2-14)$$

donde  $d_{max}$  es el máximo intervalo aceptado para un evento detectado.

En resumen, para cada evento detectado  $f_j$ , la distancia anterior  $d_k^{(o)}$  es medida para cada objeto  $H_k^{(o)}$  con un potencial de membrana activo  $M_k^{(o)} > 0$ . Esta distancia es comparada con el umbral  $d_{max}$  y asignada al objeto más cercano con distancia menor a  $d_{max}$ . Si ningún objeto está a una distancia menor a  $d_{max}$  de la detección actual entonces un nuevo objeto  $H_k^{(o+1)}$  es creado. La salida del algoritmo de rastreo puede ser representado como un flujo de eventos como en (2-15), donde  $o_l$  es el índice del  $l$ -ésimo evento generado por este método.

$$g_l = [x_l, y_l, \theta_l, o_l, t_l, p_l] \quad (2-15)$$

Como se puede ver en la Figura 2-3, el método descrito remueve virtualmente todas las falsas detecciones que se mantenían luego de la etapa de detección, además de agrupar los eventos de cada uno de los objetos capturados.

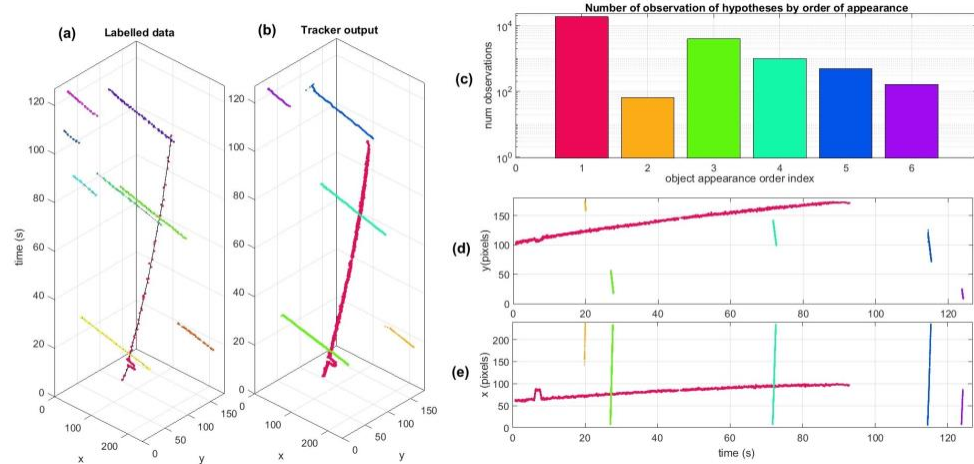


Figura 2-3 **a)** y **b)** muestran las representaciones 3D de la captura del SL-8 R/B etiquetada y la salida del algoritmo presentado respectivamente. **c)** tiene las cantidades de eventos de cada uno de los objetos. **d)** y **e)** muestran la posición de los eventos en  $x$  e  $y$  en el tiempo.

## 3 Referenciación de objetos en órbita

En este capítulo se explora la teoría tras los sistemas de coordenadas orbitales con los cuales será posible ubicar satélites para su captura y entender la notación con la que se entregan los conjuntos de datos en los catálogos de objetos en órbita.

### 3.1 Sistema de coordenadas Earth-Centered Inertial (ECI)

La importancia principal de establecer un sistema de coordenadas radica en poder describir de forma simple los elementos junto con sus vectores de estado (posición, velocidad, entre otros que se tratarán más adelante) para que de esta forma la vigilancia y el rastreo sean más efectivos. El sistema de coordenadas Earth-Centered Inertial (ICE) cumple con lo anteriormente mencionado principalmente debido a que corresponde a un sistema inercial o no acelerado, es decir, sus ejes no rotan siguiendo la rotación de la Tierra y por tanto este sistema pareciese fijo con respecto a las estrellas, por tanto, los objetos en órbita son descritos por medio de su velocidad propia y no una relativa a la rotación terrestre.

Como su nombre lo indica, el sistema de coordenadas ECI es un sistema que tiene como origen el centro de masas de la Tierra se define típicamente como un plano cartesiano en el cual el eje  $z$  corresponde al eje de rotación terrestre apuntando hacia el norte y el eje  $x$  apunta en la dirección del vernal equinox, el cual corresponde a un punto imaginario que se crea por la intersección del plano ecuatorial con la eclíptica (la trayectoria aparente del sol sobre la Tierra). El eje  $y$  completa simplemente el sistema ortogonal de mano derecha, el plano  $x$ - $y$  y por tanto, aparece circunscrito al plano ecuatorial. En la Figura 3-1 se puede observar un diagrama de este sistema.

Al conseguir la posición de un observador en la superficie terrestre y la posición de un satélite u otro objeto en órbita con este sistema, se facilita la tarea de conseguir el ángulo de captura para un evento en particular y se podrá comparar la posición de los satélites con bases de datos internacionales ya que se utiliza este mismo sistema recurrentemente.

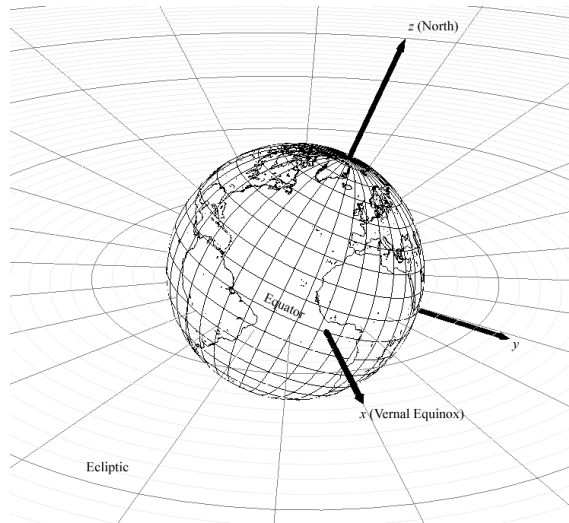


Figura 3-1 Diagrama de los ejes del sistema de coordenadas ECI

En las coordenadas ecuatoriales, un tipo de coordenada ECI, las coordenadas geográficas se transforman en coordenadas celestes, es decir la latitud pasa a llamarse Declinación (DEC) y la longitud se llama ascensión recta (AR). Se miden de igual forma que sus contrapartes geográficas, la primera corresponde al ángulo comprendido en grados desde el ecuador (positivos hacia el norte, negativos hacia el sur), en tanto la segunda corresponde al ángulo entre el meridiano de Greenwich y la posición del objeto, esta última se suele representar en unidades de tiempo, es decir,  $15^\circ$  corresponden a una hora.

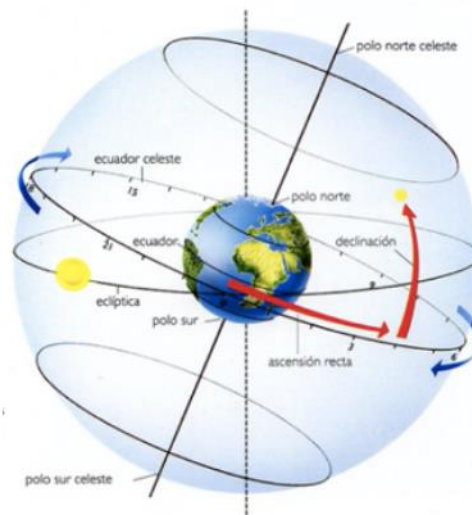


Figura 3-2 Sistema de coordenadas ecuatoriales

### 3.1.1 Posición de un observador en ECI

Considerando que se poseen medios hoy en día para conocer la posición propia en el globo, es decir, latitud y longitud es que estas serán las variables de entrada para los siguientes cálculos. En

primera instancia es necesario recordar que la forma de la Tierra no es esférica, sino que resulta más un esferoide que se achata a medida que se avanza desde el ecuador a los polos. Una exageración de esta característica se puede ver en la Figura 3-3 , en esta se puede identificar que en la posición del observador aparece otro sistema de referencias, este corresponde a las coordenadas topocéntricas y se tratarán más adelante.

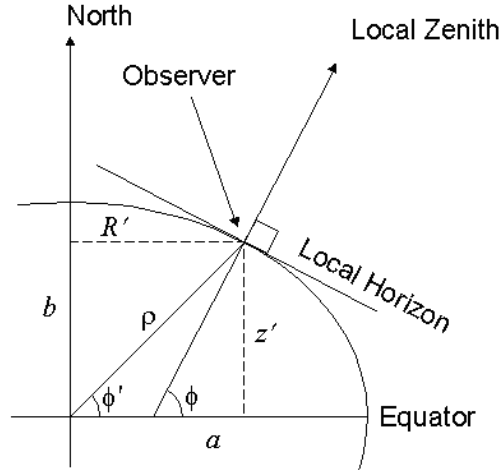


Figura 3-3 Diagrama del corte transversal de la Tierra achatada

Debido a la forma no esférica de la Tierra en cualquier posición que no esté en el ecuador o en el eje z (polo norte o sur) el zenit local, esto es la dirección perpendicular a la superficie en ese punto, no es posible extenderla hasta el centro de la Tierra o, en otras palabras, al extender una línea desde el centro de la Tierra hasta dicho punto, la línea no será perpendicular a la superficie. De aquí es que lo primero será encontrar la relación entre  $\phi$  la latitud geodésica, el ángulo entre el zenit local de un punto y el plano ecuatorial (magnitud común para la latitud, la que se encuentra en los mapas) y  $\phi'$  la latitud geocéntrica, el ángulo entre el observador y el plano ecuatorial. Para esto se utiliza como referencia el esferoide definido en World Geodetic System 1972 donde el radio ecuatorial  $a = 6378.135[Km]$  , se relaciona con el radio polar  $b$  con la siguiente ecuación:

$$b = a \cdot (1 - f) \quad (3-1)$$

donde  $f = 1/298.26$  lo que da una diferencia entre  $a$  y  $b$  de unos  $22[Km]$ . Para conseguir la relación anteriormente mencionada se utilizan la ecuación base de la elipse y trigonometría, el proceso detallado se puede encontrar en [13]. La relación en cuestión está dada por:

$$\tan(\phi') = (1 - f)^2 \tan(\phi) \quad (3-2)$$

Luego se puede notar que las coordenadas ECI están dadas por:

$$x = a \cdot C \cdot \cos(\phi) \cdot \cos(\theta) \quad (3-3)$$

$$y = a \cdot C \cdot \cos(\phi) \cdot \sin(\theta) \quad (3-4)$$

$$z = a \cdot S \cdot \sin(\phi) \quad (3-5)$$

$$R = a \cdot C \cdot \cos(\phi) \quad (3-6)$$

Donde  $C$  y  $S$  están determinados por:

$$C = \frac{1}{\sqrt{1 + f \cdot (f - 2) \cdot \sin^2(\phi)}} \quad (3-7)$$

$$S = (1 - f)^2 \cdot C \quad (3-8)$$

Para el cálculo de  $\theta$  se necesita analizar esta vez el plano x-y o plano ecuatorial, como se puede ver en la Figura 3-4 el ángulo  $\theta$  se define como una función dependiente del tiempo, la forma de calcularlo es con la suma entre la distancia angular entre el observador y el meridiano de Greenwich y la distancia entre este meridiano y el eje x o vernal equinox, esta relación se define en (3-9). De esta última ecuación,  $\theta(t)$  es lo que los astrónomos se refieren a tiempo sidéreo local,  $\lambda_E$  es la longitud Este y  $\theta_g(t)$  es el tiempo sidéreo de Greenwich (GST). El tiempo GST se puede encontrar la mayoría de las veces en relación con las  $0^h$  UTC, entonces la relación (3-9) toma la forma de la ecuación (3-10), donde  $\Delta t$  es el tiempo UTC de interés en fracción de días y  $\omega_e = 7.29211510 \times 10^{-5} \left[ \frac{\text{rads}}{\text{s}} \right]$  la velocidad de rotación de la Tierra.

$$\theta(t) = \theta_g(t) + \lambda_E \quad (3-9)$$

$$\theta(\Delta t) = \theta_g(0^h) + \omega_e \cdot \Delta t \quad (3-10)$$

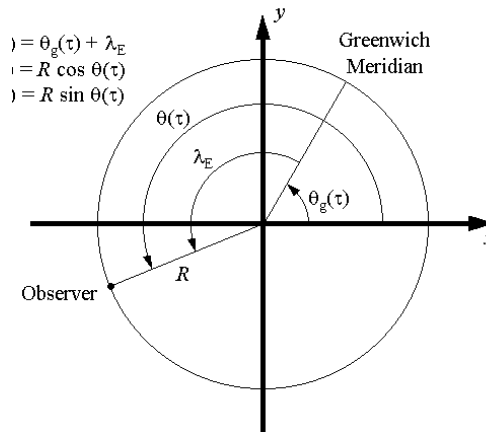


Figura 3-4 Diagrama que muestra la conversión de GMST a tiempo sidéreo local

Para conseguir  $\theta_g(0^h)$  se puede recurrir a tablas de referencia u ocupar las siguientes relaciones:

$$\theta_g(0^h) = 24110^s.54841 + 8640184^s.212866 T_u + 0^s.093104 T_u^2 - 6.2 \times 10^{-6} \cdot T_u^3 \quad (3-11)$$

$$T_u = d_u / 36525 \quad (3-12)$$

En la última relación  $d_u$  esta dado por el número de días transcurridos en tiempo universal desde la fecha en el calendario Juliano 2451545.0 (1 de enero del 2000 a las 12 horas UT1).

### 3.1.2 Objetos en órbita en ECI

Debido a la dificultad referenciar objetos en ECI directamente es que por lo general se utiliza un sistema de coordenadas mucho más simple, el sistema de coordenadas horizontales o altazimutal. Este sistema de coordenadas es un sistema local debido a que utiliza como referencias valores actuales para el observador, estas coordenadas corresponden a la altitud, el ángulo que se comprende desde el horizonte al objeto observado y la azimutal, el ángulo que se comprende desde la dirección norte hasta la posición del objeto. En la Figura 3-5 se puede ver un diagrama de este tipo de coordenadas.

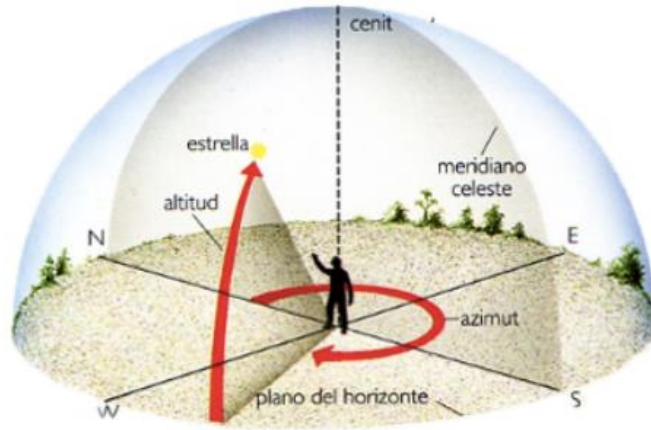


Figura 3-5 Diagrama de sistema de coordenadas altazimutal

Conociendo en el sistema de coordenadas ECI la posición de un observador  $[x_o, y_o, z_o]$  y la posición de un satélite  $[x_s, y_s, z_s]$  es posible conseguir el vector que determina el ángulo de observación como se puede ver en (3-13). El problema que existe con dicho vector es que sigue referenciado desde el centro de la Tierra y lo que se necesita es un sistema que nos permita apuntar una cámara o telescopio desde nuestra posición en la superficie, es por esto por lo que se utiliza un segundo sistema de coordenadas llamado coordenadas topocéntricas. Estas coordenadas tienen su eje  $z$  apuntando al zenit local, su eje  $x$  apuntando al sur y su eje  $y$  al este como se puede ver en la Figura 3-6.

$$[r_x, r_y, r_z] = [x_s - x_o, y_s - y_o, z_s - z_o] \quad (3-13)$$

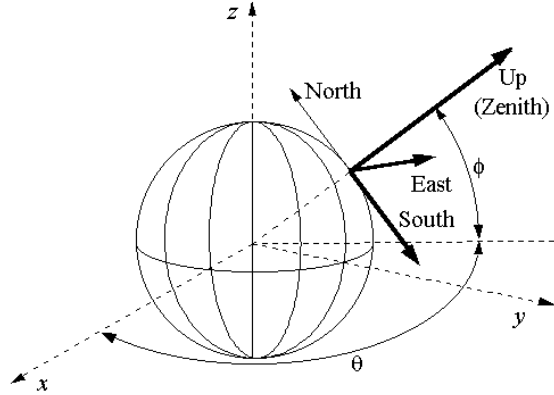


Figura 3-6 Diagrama del cambio de coordenadas ECI a topocéntricas

Para hacer el cambio de coordenadas ECI a topocéntricas solo basta con rotar el eje  $x$  en un ángulo igual a  $\theta$  (tiempo sideral local) y luego rotar el eje  $z$  en un ángulo equivalente a  $\phi$  (latitud geodésica). Las coordenadas en el sistema topocéntrico resultan como:

$$r_s = \sin(\phi) \cos(\theta) r_x + \sin(\phi) \sin(\theta) r_y - \cos(\phi) r_z \quad (3-14)$$

$$r_E = -\sin(\theta) r_x + \cos(\theta) r_y \quad (3-15)$$

$$r_z = \cos(\phi) \cos(\theta) r_x + \cos(\phi) \sin(\theta) r_y - \sin(\phi) r_z \quad (3-16)$$

Por último, es posible conseguir la distancia al satélite u objeto en órbita junto con la elevación y la azimutal:

$$r = \sqrt{r_s^2 + r_E^2 + r_z^2} \quad (3-17)$$

$$EL = \sin^{-1}(r_z/r) \quad (3-18)$$

$$AZ = \tan^{-1}(-r_E/r_s) \quad (3-19)$$

### 3.2 Estandar para mensajes de objetos en órbita

La mantención del más grande catálogo de objetos en órbita está distribuida entre varias organizaciones estadounidenses, entre estas una de las más importantes corresponde al Escuadrón 18 de Control Espacial (18 SPCS) el cual entrega soporte 24/7, administra el catálogo y comparte la información de este en [14]. La mayoría de las detecciones y rastreo se realizan por medio de la llamada Surveillance Space Network (SSN), una red de colaboración de distintos telescopios, sitios de radar e incluso satélites que realizan las tareas mencionadas con el objetivo de mantener actualizados el catálogo. En la Figura 3-7 se puede ver un diagrama de la distribución de la SSN.



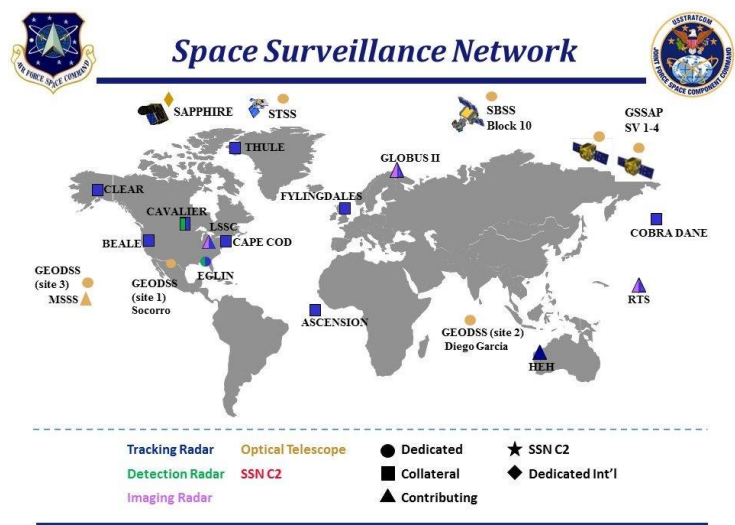


Figura 3-7 Instalaciones de la SSN alrededor del mundo

Los servicios entregados en no solo comprenden la entrega del mensaje común TLE, sino que desde ya hace unos años el 18 SPCS ha estado proveyendo servicios a terceras partes de carácter comercial, entre estos servicios se puede mencionar la posición de todos los objetos rastreados, mensajes de posibles colisiones, soporte para disponer de un satélite cuando se agote su vida útil, cambios de orbita, entre otros.

3.2.1 Formato Two Line Element

El estándar más utilizado para la distribución de mensajes para objetos en órbita corresponde al Two Line Element (TLE), este tipo de mensaje codifica elementos orbitales de objetos en órbita en dos líneas de 69 caracteres cada una. Dichos elementos, utilizando alguno de los modelos predictivo que se describirán en la siguiente sección, permiten estimar el estado del objeto para cualquier punto pasado o futuro con cierta precisión. A continuación, se explican brevemente cuales son los elementos presentes codificados en cada una de las dos líneas de los mensajes TLE, además, en la Tabla 3-1 y en la Tabla 3-2 se pueden ver el resumen de cada campo para la línea 1 y la línea 2 respectivamente, por último, en el Listado 3-1 se presenta un ejemplo de cómo son los mensajes TLE [15].

1	ISS (ZARYA)
2	1 25544U 98067A 04236.56031392 .00020137 00000-0 16538-3 0 9993
3	2 25544 51.6335 344.7760 0007976 126.2523 325.9359 15.70406856328906
4	123456789012345678901234567890123456789012345678901234567890123456789
5	1 2 3 4 5 6

Listado 3-1 Ejemplo de conjunto de elementos en formato TLE

Tabla 3-1 Descripción de los campos de la primera línea de los mensajes TLE

Campo	Columnas	Descripción
1.1	01	Número de línea del TLE
1.2	03-07	Número de elemento en el catalogo
1.3	08	Clasificación de la información
1.4	10-11	Últimos dos dígitos del año de lanzamiento
1.5	12-14	Número de lanzamiento del año
1.6	15-17	Pieza de lanzamiento
1.7	19-20	Época – Año
1.8	21-32	Época – Día del año y fracción del día
1.9	34-43	Primera derivada del promedio de movimiento
1.10	45-52	Segunda derivada del promedio de movimiento
1.11	54-61	BSTAR termino de arrastre
1.12	63	Tipo de efeméride
1.13	65-68	Número de conjunto de elementos
1.14	69	Checksum

Ambas líneas poseen en su primer campo el número de línea y como segundo elemento (1.2/2.2) el numero referente a su identificador en el SATCAT (satellite catalogue) de NORAD (North American Aerospace Defense Command). Dicho identificador es un numero único que posee cada objeto artificial en órbita.

En la primera de las dos líneas los campos 1.4, 1.5 y 1.6 se refieren a identificaciones internacionales. Tanto el campo 1.3 como el campo 1.12 son utilizados de forma privada puesto que ambos tienen valores fijos cuando los TLE son dispuestos al público general, de todas maneras el primero se refiere a la clasificación de la información por lo que siempre llevará una “U” de unclassified (desclasificado) y el campo 1.12 indica el tipo de modelo utilizado para la propagación, este campo se usa solo de estudio y para público general siempre poseerá un “0” puesto que todos estos objetos son propagados con el modelo SGP4/SDP4 (más de esto en la siguiente sección).

Los campos entre 1.7 – 1.11 entregan información importante del estado del objeto, 1.7 y 1.8 definen el tiempo del conjunto de elementos, a esto se le conoce como época, además los demás campos dependientes del tiempo ocupan este valor como referencia. El campo 1.9 representa la primera derivada dividida por dos y en unidades de revoluciones por día<sup>2</sup> y el campo 1.10 representa la segunda derivada dividida por seis y en unidades de revoluciones por día<sup>3</sup>. Juntos dan una mejor medida del movimiento promedio, pero ninguno de estos valores es utilizado en los modelos SGP4/SDP4 solo en la versión más simple SGP. El campo 1.11 entrega información de una magnitud llamada B\*(BSTAR) que es un coeficiente de arrastre utilizado en el modelo SGP4, en la teoría aerodinámica todo objeto tiene un coeficiente balístico B representado por (3-20), donde  $C_D$  es el coeficiente de arrastre,  $A$  la sección transversal y  $m$  la masa del objeto. B\*, como se ve en (3-21) es un ajuste utilizando el valor de referencia de la densidad atmosférica  $\rho_0$ . El

coeficiente balístico representa la susceptibilidad del objeto a ser arrastrado, mientras más alto más susceptible.

$$B = C_D \cdot A/m \quad (3-20)$$

$$B^* = B \cdot \rho_o/2 \quad (3-21)$$

El campo 1.13 es el número del conjunto de elementos, normalmente cada vez que un nuevo mensaje TLE es generado para dicho elemento, este número aumenta, posibilitando tener conocimiento que tan actual es el mensaje. Por último, el campo 1.14 (al igual que el campo 2.10 de la segunda línea) es el checksum modulo 10 que verifica la integridad del mensaje.

Tabla 3-2 Descripción de los campos de la segunda línea de los mensajes TLE

Campo	Columnas	Descripción
2.1	01	Número de línea del TLE
2.2	03-07	Número de elemento en el catalogo
2.3	09-16	Inclinación (grados)
2.4	18-25	Ascensión derecha del nodo ascendente (grados)
2.5	27-33	Excentricidad (se asume punto decimal)
2.6	35-42	Argumento del perigeo (grados)
2.7	44-51	Anomalía media (grados)
2.8	53-63	Movimiento promedio (Revoluciones por día)
2.9	64-68	Numero de revolución de la época (Revoluciones)
2.10	69	Checksum

Los elementos presentes en esta segunda línea guardan información acerca de la órbita que se describe más que del objeto en cuestión. Para explicar los campos del 2.3 al 2.7 se usa como referencia la Figura 3-8.

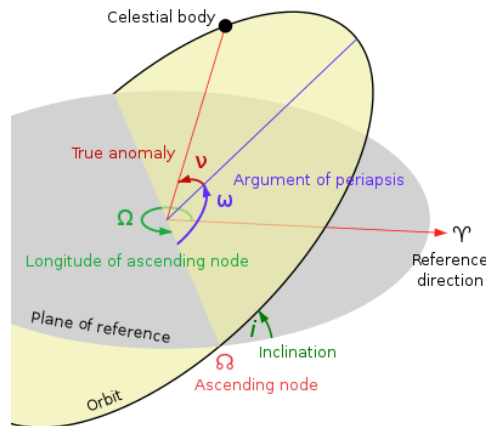


Figura 3-8 Diagrama con los elementos orbitales de un objeto en órbita

Cabe recordar que el plano de referencia se corresponde con el plano ecuatorial terrestre y la dirección de referencia es el eje  $x$  en el sistema de coordenadas ECI, es decir, la dirección al vernal equinox.

El campo 2.3 indica el ángulo que existe entre el plano de referencia y el plano orbital, indicada en la figura por  $i$ . En tanto 2.4, la ascensión derecha del nodo ascendente o también conocido como la longitud al nodo ascendente es el ángulo que se comprende desde la dirección de referencia al nodo donde el objeto comienza la ascensión ( $\Omega$  en la figura), este ángulo es medido en sentido contra reloj y está inscrito en el plano de referencia. 2.5 es la excentricidad, característica propia de las elipses, se asume que son valores decimales por lo que un valor de 1234567 en TLE corresponde a 0.1234567. El campo 2.6 corresponde al ángulo ( $\omega$  en la figura) que va desde el nodo ascendente hasta el perigeo (el punto donde la distancia entre el objeto y Tierra es mínima). El campo 2.7 la anomalía media es la fracción en grados que ha transcurrido desde el perigeo hasta la posición actual del objeto orbitando ( $\nu$  en la figura).

Los últimos dos campos corresponden tanto al movimiento medio en el 2.8 y el número de revolución actual de la época en el 2.9, este último corresponde a la cantidad de revoluciones alrededor de la Tierra del objeto en ese preciso instante (época).

Es necesario mencionar que el uso de este tipo de codificación viene desde la década del 60 y no ha estado exento de problemas. El primer problema fue encontrado a principios de siglo cuando el campo 1.4 amenazaba con tener que cambiar este formato debido a que la identificación era de dos dígitos y el cambio de siglo. Aun que el cambio fue nulo debido a que no existen cuerpos artificiales que orbiten la Tierra antes del 57 y por tanto no habría que hacer ningún cambio por lo menos por medio siglo más, si algunos sistemas y algoritmos tuvieron que ser revisados para su correcto funcionamiento. Actualmente un nuevo problema de mayor complejidad aparece en la creación de estos mensajes, la cantidad de objetos catalogados está superando los 5 dígitos con los que se identifican estos objetos (campos 1.2 y 2.2). Hasta el momento es una incertidumbre si se optara por cambiar el tipo de mensaje por otro que considere estos problemas o si se cambiara algún segmento de la codificación de los TLE, por ejemplo, una solución dada es la llamada Alpha-5 la cual consiste en que el primero de los 5 dígitos sea cambiado por una letra a la cual se le asigna un valor numérico de acuerdo con su posición en el alfabeto, esto para los números superiores a 99,999. Esto aumenta la cantidad para representar en más de 240,000 objetos.

### 3.3 Stellarium

Gracias a que los mensajes TLE llevan décadas como mensajes libres al público y debido a que existen una gran cantidad de entusiastas de la actividad astronómica es que programas como Stellarium proveen con las herramientas necesarias para generar avistamientos no solo de fenómenos astronómicos, sino que también es posible ubicar satélites y basura espacial con esta aplicación.

Stellarium es un software de código abierto que permite la visualización del cielo tal y como se puede ver a simple vista, posee muchas funcionalidades entre las más importantes para este proyecto corresponden la capacidad de poner tu ubicación, la de crear el sensor y cámaras a

elección con el objetivo de saber aproximadamente lo que el sistema óptico puede captar y por último el hecho de que permite visualizar el movimiento de los satélites ya que actualiza desde catálogos internacionales las trayectorias de estos. Permite incluso buscar y añadir las fuentes de catálogos que más se adapten a las necesidades del usuario. Además, muestra los objetos tanto en coordenadas celestiales como coordenadas horizontales lo que permite utilizar dichos datos para referenciar las mediciones que se realicen.

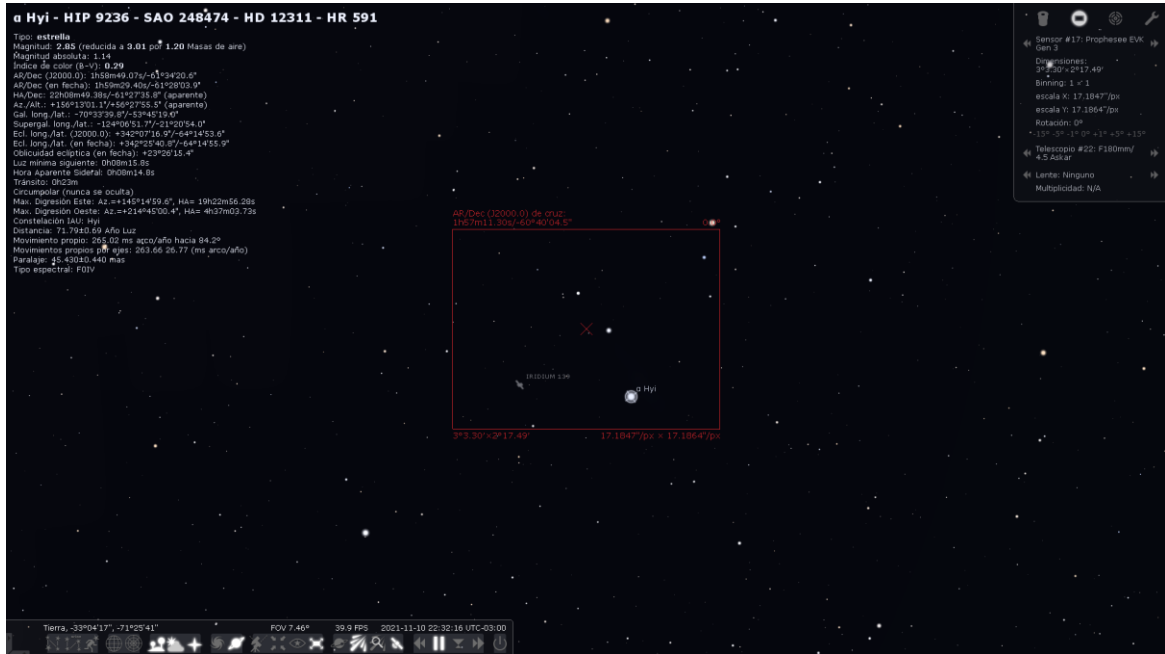


Figura 3-9 Pantalla de Stellarium, arriba a la izquierda la información del objeto seleccionado, arriba a la derecha selección de sensores y lentes, abajo controles de visualización.

En la Figura 3-9 se muestra una imagen de lo que se presenta en la pantalla de Stellarium. En esta imagen arriba a la izquierda se pueden ver las características principales del objeto seleccionado (aquí es donde se encuentran las coordenadas AR/DEC y Alt/Az), arriba a la derecha se encuentra el plugin de "Oculares" el cual permite la creación de sensores y sistemas ópticos, con el cual se puede obtener en pantalla el "field of view" aproximado del setup que se utiliza. Por último, abajo a la izquierda se encuentran los controles de visualización y control de tiempo.

## 4 Aplicación para la detección de satélites

En el presente capítulo se describen las principales características de la aplicación que se utilizará en capítulos posteriores para conseguir la captura de satélites y basura espacial en orbitas LEO. La aplicación en cuestión fue desarrollada enteramente en lenguaje C++. A continuación, se mencionan algunas de las librerías utilizadas para el procesamiento de los eventos y que son proporcionadas por el fabricante, luego se presenta la estructura de la aplicación junto con el funcionamiento de cada una de sus partes.

### 4.1 Librerías

Para simplificar la comunicación de ordenadores con la cámara es que Prophesee provee librerías con funciones y código precompilado en forma de archivos de librerías estáticas (.lib) y librerías dinámicas (.dll), en esta sección se explica el funcionamiento de las principales librerías utilizadas en el código del anexo señalado, para esto último se utiliza la documentación entregada por el fabricante en [16].

Las librerías están formuladas bajo el concepto de HAL (Hardware Abstraction Layer), con el cual se usan los plugins para obtener acceso a las características del hardware de la cámara. Cada característica es mapeada por separado a una API (Application Programming Interface) la cual recibe el nombre de instalación (“facility”).

#### 4.1.1 Metavision::Device

La cámara basada en eventos es abstraída sobre la clase Metavision::Device, esto permite utilizar las propiedades que otorgan las instalaciones. Además, es la clase que permite el reconocimiento de los dispositivos conectados e incluso permite abrir un archivo .RAW y utilizar las mismas características tal y como si fuera un flujo de eventos obtenido desde una cámara. Esta clase es necesaria para poder utilizar las demás instalaciones y sus características.

La función Metavision::DeviceDiscovery::open() es la utilizada para interrogar a cada uno de los plugins acerca de que dispositivos detectan, el primer dispositivo compatible es el abierto para su

utilización. Por otro lado, con `Metavision::DeviceDiscovery::open_raw_file("Archivo .RAW")` se abre un archivo .RAW en lugar de conseguir un flujo de eventos en tiempo real.

Luego cada instalación puede ser definida ocupan el operador junto con el objeto de clase Device de la siguiente manera:

```
1 Metavision::Device device -> get_facility<"Metavision::Facility">
```

Listado 4-1 Función para añadir "*facilities*"

De la forma anterior, la instalación puede ser utilizada para realizar las tareas que le corresponden, a continuación, se explican las más importantes utilizadas en la creación de la aplicación.

#### 4.1.2 Metavision::I\_EventStream

Esta clase se utiliza para conseguir buffers de datos desde una cámara o un archivo según como se ha definido `Metavision::Device`. Con esta clase se puede comenzar y detener la transmisión de eventos con los operadores `->start()` y `->stop()` respectivamente. Además, solo basta con usar `->get_lastest_raw_data('n_bytes')` para conseguir el buffer con los últimos datos, 'n\_bytes' es el tamaño de este buffer y es entregado por esta función. Dicha función trabaja de forma distinta dependiendo si Device corresponde a un archivo .RAW o una cámara, si se trata de este primer caso el buffer que tomará para la lectura del flujo de eventos será constante una vez iniciada la transmisión, en cambio, si se trata de una cámara la función en cuestión entregará todos los eventos que se hayan generado desde el ultimo llamado de la misma función, esto se debe al carácter asincrónico nativo de las cámaras basadas en eventos ya que el flujo de eventos dependerá de la escena en cuestión. Debido a esta última característica es que el procesamiento se debe llevar con cuidado alrededor de esta función porque si tarda demasiado los eventos se acumularan continuamente hasta que se genere un error de procesamiento.

#### 4.1.3 Metavision::I\_DeviceControl

Esta instalación permite controlar el modo de operación de la cámara de eventos ya que puede ser utilizada como un dispositivo autónomo o en un arreglo tanto como esclavo o maestro, pero además permite controlar la generación de eventos desde el lado de la cámara. Este proyecto solo contempla la utilización de una sola cámara por lo que solo se utilizara la segunda de las funciones de esta clase. El modo de funcionamiento es similar con `I_EventStream` usando los operadores `->start()` y `->stop()`.

#### 4.1.4 Metavision::I\_EventDecoder

`I_EventDecoder` se utiliza para decodificar cierto tipo de evento. Como anteriormente fue mencionado en la sección **¡Error! No se encuentra el origen de la referencia.**, existen diferentes tipos de evento como los EM (exposure measure – eventos obsoletos en las últimas versiones de Metavision), los Extrenal Triggers (Triggers utilizados con los pines que entrega la cámara) o los CD (Contrast Detection), estos últimos son los detallados en dicha sección y los cuales representan las características de estas cámaras, `I_EventDecoder` puede decodificar cada uno de

los tipos mencionados, aunque para ello se necesita invocar utilizando la clase de dicho tipo de evento como:

```
1 Metavision::Device device ->get_facility
  <Metavision::I_EventDecoder<Metavision::EventCD>>()
```

Listado 4-2 Función para añadir el decodificador de eventos

Por otro lado, con las funciones `->set_add_decoded_event_callback()` y `->set_add_decoded_vevent_callback()` es posible registrar funciones lambda a cada evento decodificado o a cada lote de eventos decodificados respectivamente.

#### 4.1.5 Metavision::I\_LL\_Biases

Esta instalación es de gran utilidad ya que permite el ajuste de los biases explicados en la sección 1.4.2. Con las funciones `->set("nombre bias", "valor")` y `->get("nombre bias")` es posible ajustar "nombre bias" con "valor" o conseguir el valor de "nombre bias" respectivamente.

#### 4.1.6 Metavision::I\_Decoder

Esta instalación se encarga de recibir los datos codificados desde la cámara y despachar partes de dicho buffer a las instancias que correspondan de `I_EventDecoder`. Con el operador `->decode(*ev, *evend)` se ingresan tanto el puntero del principio del lote de datos como el puntero del final para enviar los eventos a la función decodificadora.

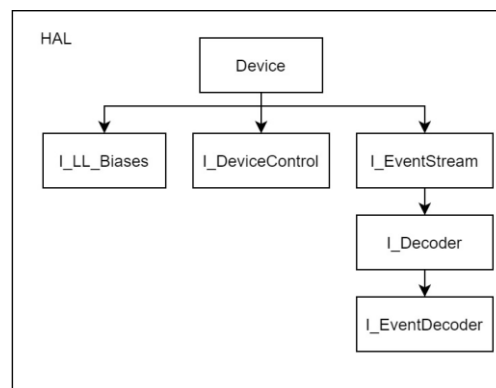


Figura 4-1 Diagrama del Hardware Abstraction Layer

## 4.2 Estructura del código

La estructura del archivo posee 3 partes esenciales que se han dividido en hebras para lograr un funcionamiento en paralelo, la primera de ellas corresponde al hilo de decodificación donde los eventos son extraídos desde la cámara y decodificados para el procesamiento y la visualización, en segundo lugar se encuentra el núcleo de visualización el cual se encarga de la interfaz de usuario y las ventanas para observar en tiempo real la captura de eventos, y por último el hilo de detección es donde se procesan los eventos para encontrar satélites en las imágenes captadas. A continuación, se detallan más a fondo cada una de estas secciones.



### 4.2.1 Nucleo de decodificación

Esta sección utiliza las librerías de Metavision para conseguir desde la cámara el flujo de eventos de manera dinámica, decodificarlos del formato nativo raw de los eventos para conseguir los parámetros de cada activación  $(x,y,ts,p)$  y luego hacer uso de esta información de distintas maneras, siendo la principal el almacenaje de las marcas de tiempo en las llamadas superficies de tiempo.

Las superficies de tiempo permiten la implementación de un filtro de ruido basado en vecindades en regiones de interés, tal y como se explicó en el apartado 2.1.1. Dicho filtro funciona evento a evento tomando una región de interés alrededor del evento actual y analiza si es que existieron activaciones en pixeles de la vecindad dentro de un tiempo límite anterior ( $\Phi$ ), una cantidad más alta que algún umbral fijado ( $L$ ) por el usuario promueve a dicho evento a la siguiente etapa de procesamiento.

Para probar esta etapa de procesamiento se realizó una grabación en tiempo real de una mano moviéndose como se ilustra en la Figura 4-2. En la imagen, se nota como disminuye el ruido entre la imagen filtrada (derecha) y la no filtrada (izquierda), pero se pierden bastantes detalles de la escena. Por otro lado, se toma como suposición y objetivo de diseño que las escenas serán fundamentalmente “*sparse*” por lo que para esta aplicación escenas ricas en movimientos o con muchos detalles afectan negativamente el rendimiento debido a la mayor cantidad de eventos que es necesario procesar, por suerte la captura de satélites no cae en ninguna de estas dos categorías.

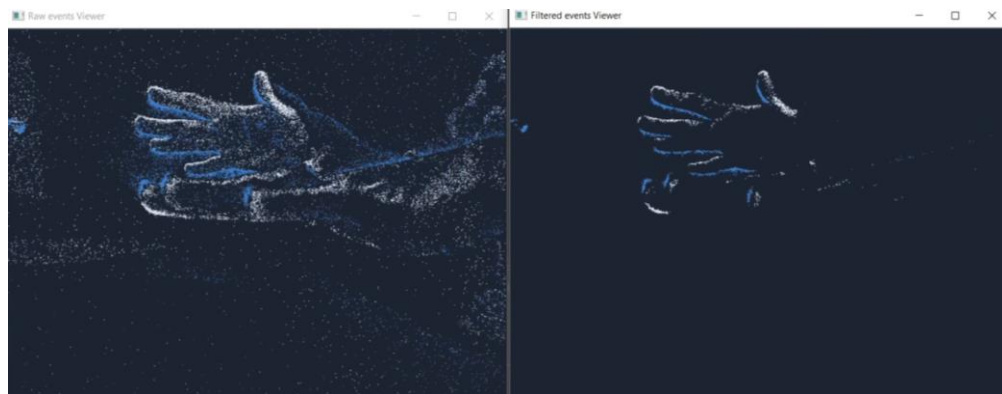


Figura 4-2 Prueba en tiempo real utilizando la cámara de eventos con  $L = 6$  y  $\Phi = 40000$ .

En general, el ajuste de sensibilidad de la aplicación se puede hacer tanto por hardware, cambiando los distintos valores para los bias o por software ajustando los parámetros de los ejemplos anteriores. Ambas tareas no son triviales en complejidad y es necesario realizar cuantiosas pruebas con el fin de conseguir una relación que resulte útil para el correcto funcionamiento de la aplicación, en la mayoría de los casos existirá un intercambio entre la saturación del sensor o el visualizador contra la pérdida de información importante en la escena. El objetivo final de hacer estos ajustes siempre será poseer el mayor contraste posible entre los eventos activos luego del filtrado expuesto anteriormente.

### 4.2.2 Nucleo de visualización

El hilo de visualización controla el flujo principal de la aplicación y permite la interacción entre el usuario y el programa. Se han utilizado las librerías de OpenCV, una conocida biblioteca libre para la realización de proyectos relacionados a la visión por computador, para generar ventanas que permitan la visualización de los eventos codificados y luego filtrados en imágenes, además una ventana permite el ajuste de los 4 bias principales (*bias\_diff\_on*, *bias\_diff\_off*, *bias\_fo*, *bias\_hpf*) por medio de deslizadores. Por otro lado, el programa espera en todo momento la retroalimentación del usuario para realizar tareas como lo son seleccionar un área con el ratón en una de las ventanas para asignar una región de interés o presionar un botón del teclado para comenzar o detener algunas funciones.

Las ventanas de visualización de la aplicación son 2, la primera de estas corresponde a la ventana ‘*Raw events Viewer*’ en la cual se muestran los eventos extraídos desde la cámara sin ningún procesamiento, en segundo lugar esta la ventana ‘*Filtered events Viewer*’ donde se presentan los eventos luego del filtro de ruido explicado en la sección anterior, ambas ventanas son refrescadas en este hilo con una tasa de 25 cuadros por segundo, aunque debido a la naturaleza de los eventos esta tasa no es estable. La Figura 4-2 muestra ambas ventanas con el ejemplo de las manos de la sección anterior.

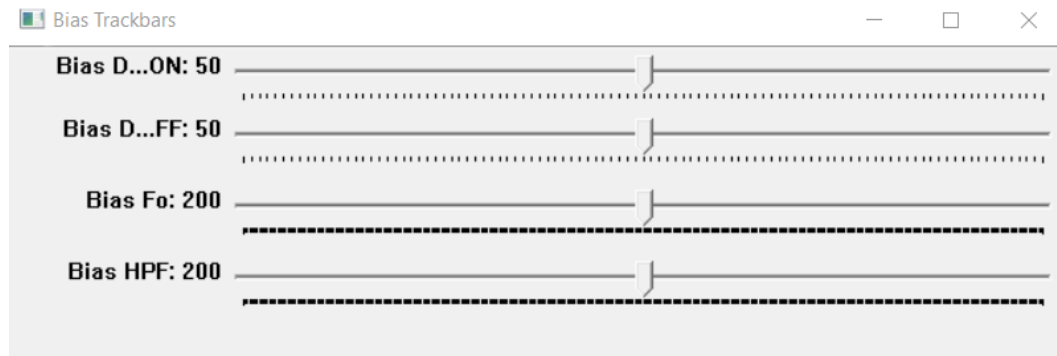


Figura 4-3 Ventana ‘Bias Trackbars’

Una tercera ventana se encarga de mantener las barras deslizantes para el cambio de los bias. Se ha optado mantener solo los 4 bias mencionados debido a que estos generan cambios importantes en el funcionamiento del sensor, y se ha excluido bias que no se recomienda el cambio de su valor de funcionamiento (como *bias\_pr* o *bias\_diff*) o bias cuya función sobrepase las funcionalidades necesarias para la aplicación (como *bias\_refr* cuya función ajusta el tiempo refractario de cada pixel en el orden de unos cuantos microsegundos, tiempo que ya es suficiente para esta aplicación). Además, para a los bias escogidos se les ha limitado los valores que se pueden manejar con las barras deslizantes para asegurar la robustez de la aplicación contra la saturación del sensor, estos nuevos valores son más que suficientes para realizar las pruebas correspondientes. En la Tabla 4-1 se presentan los valores que pueden tomar estos parámetros, también en la Figura 4-3 se muestra la disposición que tienen estos bias en la ventana ‘*Bias Trackbars*’.

Tabla 4-1 Valores límites para bias en las barras deslizantes

Bias	Valor inicial	Rango
Bias_diff_on	50	1-100
Bias_diff_off	50	1-100
Bias_fo	200	1-400
Bias_hpf	200	1-400

Por último, el operador puede relacionarse con la aplicación dentro de las ventanas de visualización seleccionando una región de interés o presionando un botón particular del teclado. Para seleccionar un ROI basta con hacer click en una de las ventanas de visualización y arrastrar hasta conseguir el resultado esperado, luego en cualquier momento es posible presionar 'r' para reestablecer la imagen sin un ROI seleccionado. Para conseguir los valores referenciados de las detecciones es necesario ingresar los valores de altitud y azimutal con los cuales se está apuntando con la cámara, para hacer esto se debe presionar la tecla 'a' para que en la ventana de comandos sea posible colocar dichos valores uno a uno en formato grado entero, fracción de arcmin, fracción arcsec tal y como aparecen en Stellarium. Por otro lado, para comenzar la grabación y detección de objetos es necesario apretar la tecla 'g' en el teclado, esto permite que la grabación se vea exenta de cualquier proceso ajeno a la detección como el enfocar el sistema óptico, el ajuste de parámetros y apuntar alguna región de la esfera celeste, para terminar la grabación se debe presionar nuevamente la tecla 'g', se pueden realizar estas acciones las veces que sean necesarias. El ultimo comando que puede ser ingresado es presionando la tecla 'q' la cual finaliza y cierra inmediatamente el programa. Un resumen de los comandos que se pueden realizar en la aplicación se presenta en la Tabla 4-2.

Tabla 4-2 Resumen de comandos para la interfaz de usuario

Comando	Descripción
Arrastrar ratón sobre ventana	Establece un ROI en el cual solo los pixeles de dicha zona generan eventos.
Presionar letra 'a'	Permite ingresar por la línea de comandos los valores de altitud y azimutal.
Presionar letra 'r'	Reestablecer la visualización con todos los pixeles activos.
Presionar letra 'g'	Comienza la grabación del archivo/Detiene la grabación del archivo.
Presionar letra 'q'	Terminar el procesamiento y cerrar la aplicación.

### 4.2.3 Nucleo de detección

Esta última sección del código encuentra en el flujo de eventos los movimientos de objetos que se cruzan por la escena, esto se realiza por medio de pequeñas integraciones cada medio segundo

aproximadamente a las cuales se le aplica la transformada de Hough, un algoritmo detector de líneas.

Cuando la trayectoria de un objeto que se cruce en la escena genere una línea esta será detectada por medio de la transformada de Hough, comenzará así el proceso de detección el cual comienza a sumar las imágenes extraídas siguientes, esto permite conseguir completar la trayectoria del objeto por el espacio capturado. Cuando el detector ya no encuentra líneas en las imágenes extraídas una cuenta regresiva comienza, si en los siguientes 2 ciclos no se detecta nuevamente una línea el proceso de integración se detiene y pasa a la última etapa de detección, esta cuenta regresiva ayuda a detectar objetos tenues que desaparezcan aparentemente o que su propia rotación genere 'glinting', es decir, en ciertos momentos no refleje luz solar.

Como último eslabón de procesamiento a la imagen integrada se le aplica nuevamente una transformada de Hough, esta vez con umbrales que detecten solo líneas de mayor tamaño teniendo en cuenta que en la imagen integrada debería aparecer una trayectoria más completa este paso ayuda a evitar detecciones falsas provocadas por el ruido.

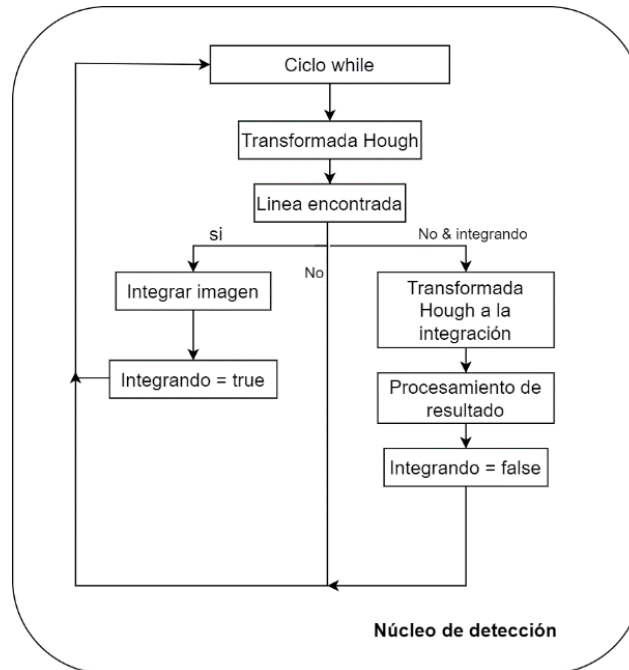


Figura 4-4 Diagrama del funcionamiento del núcleo de detección

Si una detección es confirmada las líneas generadas con el algoritmo de la transformada de Hough son utilizadas para crear una única línea la cual represente de forma lineal la trayectoria del objeto. Una única línea simplifica la extracción de las marcas de tiempo cuando hay que hacerlas coincidir con puntos de dicho movimiento. Luego esta información se utiliza para imprimir en la ventana de comando los valores más significativos como lo son la velocidad en ["/s], la distancia con la que se calculó dicho valor en ["] y las coordenadas celestiales de los puntos extraídos junto con el tiempo UTC en el que ocurrieron (Figura 4-5). Una imagen del diagrama de flujo de este hilo puede observarse en la Figura 4-5.

```

Detection at 2021-11-25-23_35_54
Tiempo deteccion 1: 3.157726
Distancia calculable : 5348.003670" | Velocidad : 1693.624992"/s
Punto1 AR/DEC : 23.000000h 1.000000m 29.370541s / -28.000000° 42.000000' 13.288182" | Tiempo : 2.000000h 35.000000m 50.137793s
Punto2 AR/DEC : 22.000000h 59.000000m 35.233944s / -29.000000° 27.000000' 18.648354" | Tiempo : 2.000000h 35.000000m 53.295519s

```

Figura 4-5 Ejemplo de mensaje de detección de objeto.

Además, este hilo posee una ventana visualizadora ‘*Last Detection Viewer*’ donde se imprime la imagen integrada de la última detección realizada, esto se realiza con el objetivo de corroborar las detecciones hechas en el acto (Figura 4-6). Por otro lado, se almacenan en disco tanto la imagen de la integración y una imagen con la línea dibujada que representa la confirmación de la detección.

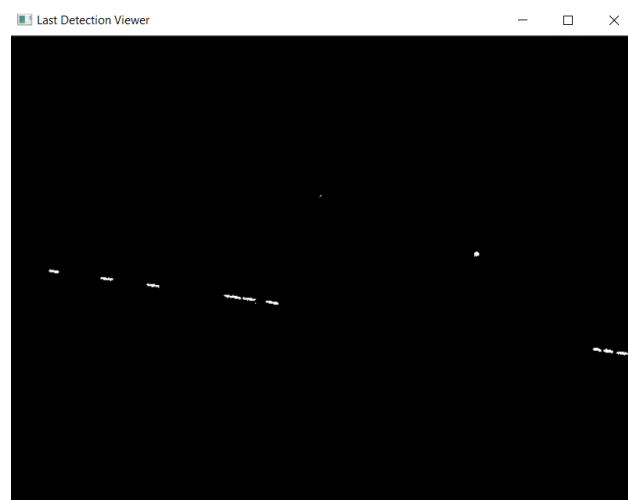


Figura 4-6 Ventana ‘Last Detection Viewer’ con la imagen de la última detección realizada

Cabe mencionar que este hilo se mantiene en ejecución en todo momento, pero solo comienza a realizar procesamientos una vez que del hilo principal (hebra de visualización) se comienza con la grabación de datos.

Por último, la aplicación almacena de forma automática el flujo de eventos .RAW, imágenes con la integración de las detecciones y un archivo de texto con los mensajes de cada detección. Los archivos de texto contienen una cabecera con los valores de las coordenadas geográficas (latitud y longitud) y las coordenadas horizontales (altitud y azimutal), esto con el objetivo de poseer la información necesaria a la hora de realizar el emparejamiento utilizando Stellarium.

### 4.3 Marco teórico

En esta sección se explican dos métodos que se han incluido en el programa, esto se realiza para explicar de mejor forma como se detectan los objetos capturados. El primero corresponde a la transformada de Hough, un algoritmo de detección de líneas del cual se conoce mucha literatura debido al tiempo que tiene en uso. El segundo es la forma en la que se realiza la transformación del espacio de las imágenes capturadas al plano de la esfera celeste en las coordenadas ecuatoriales, para esto es necesario también incluir ciertos cálculos acerca de las características del sistema óptico utilizado.

### 4.3.1 Transformada de Hough

La transformada de Hough es el algoritmo principal utilizado en la detección de esta aplicación. Este método expresa las líneas en una imagen en coordenadas polares como se puede ver en la Figura 4-7 a), la ecuación puede ser expresada como la (4-1).

$$r = x \cdot \cos(\theta) + y \cdot \sin(\theta) \quad (4-1)$$

En general, a cada punto  $(x_0, y_0)$  se puede definir una familia de líneas que la atraviesan, al graficar dichas líneas en el espacio polar obtenemos una senoide como se puede apreciar en Figura 4-7 b). Luego, si se grafican todos los puntos de una imagen obtenemos distintas sinusoides que se cruzan en distintos puntos del plano polar (Figura 4-7 c) ), estas intersecciones representan que dichos puntos pertenecen a la misma recta en el plano cartesiano.

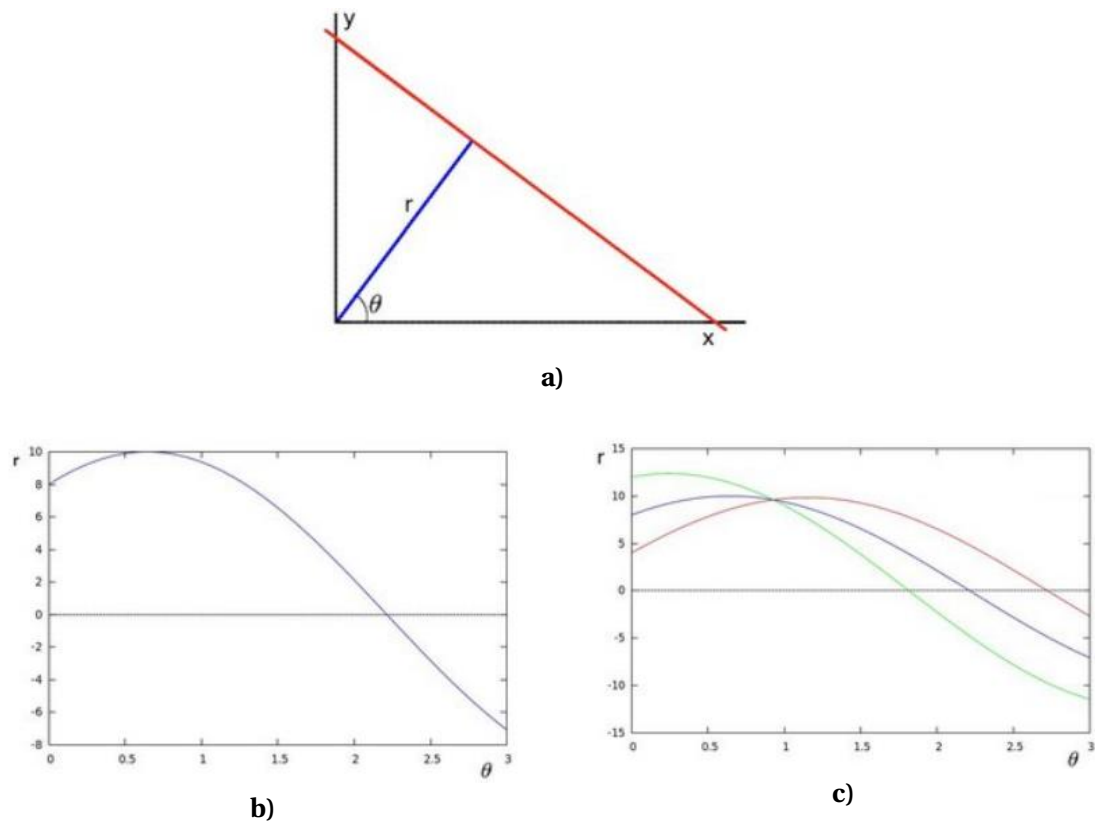


Figura 4-7 a) Representación de una recta en coordenadas polares. b) y c) Ilustración del funcionamiento de la transformada de Hough.

Puntos en el plano polar con mayor cantidad de intersecciones de sinusoides representan una mayor probabilidad a que el punto en cuestión coincida con una recta en la imagen original, luego definiendo algún valor de umbral que se deba superar para confirmar una detección es que finaliza dicho proceso.

Al momento de procesar la imagen, este algoritmo necesita que las líneas posean un nivel de claridad que destaque, por lo general al utilizarlo con imágenes tradicionales a color con tres

canales, estas se suelen llevar a escala de grises para luego ocupar algún algoritmo de obtención de bordes, procesos que complejizan los sistemas. En este caso debido a la naturaleza de los eventos es fácil conseguir una imagen de alto contraste que permita el uso directo del algoritmo, además la escena no posee formas complejas que generen carga al procesamiento, y también debido al bajo campo de visión que posee esta cámara podemos decir con certeza que cualquier cuerpo orbitando que se capture en la escena poseerá una trayectoria mayoritariamente recta, lo que vuelve ideal el uso de esta transformada para esta aplicación [17].

#### 4.3.2 Conversión de coordenadas horizontales a coordenadas ecuatoriales

Antes de poder realizar la conversión en cuestión, es necesario conseguir las coordenadas ecuatoriales en todo el espacio capturado. Para esta tarea apuntando algún objeto del cual se conozca con certeza la altitud y azimutal en la que se encuentra en un momento dado, se podrá extrapolar las coordenadas horizontales de cualquier punto de la escena utilizando lo que se conoce como “*instantaneous field of view*” o IFOV. Este valor representa la resolución en el espacio observado a través de un sistema óptico, es decir, cuanto es el cambio angular entre un píxel y otro. La forma de calcularlo está dada por (4-2). Conociendo este valor solo basta multiplicarlo con la diferencia entre el píxel conocido y el que se quiere conocer, en el eje x para la azimutal y en el eje y para la altitud.

$$IFOV = 206.265 \cdot \left( \frac{\text{tamaño de pixel } [\mu m]}{f \text{ [mm]}} \right) \quad (4-2)$$

Es necesario notar que este método de conseguir las coordenadas horizontales comienza a ser menos preciso al subir en valores de altitud ya que la curvatura natural de este tipo de coordenadas al zenith comienza a distorsionar la escena. Además, es útil siempre y cuando se mantenga la escena de forma recta con resta, es decir, en paralelo al horizonte. Ambos problemas anteriores se podrían resolver utilizando algún tipo de montura, es más, no sería necesario el calculo para conseguir las coordenadas ecuatoriales puesto que estos dispositivos lo entregan de manera integrada. Para este proyecto no se dispone de este dispositivo la mayoría del tiempo, por lo que es necesario este tipo de ajuste, esto se trata de mejor manera en la sección donde se explica acerca de la implementación.

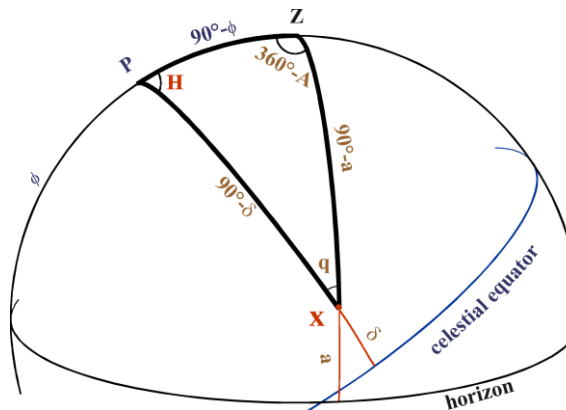


Figura 4-8 Diagrama “triángulo astronomico XPZ”

Luego, conociendo las coordenadas horizontales basta con realizar los cálculos necesarios para convertir a coordenadas ecuatoriales. Para realizar esto último, se utiliza el llamado “triángulo astronómico XPZ”, el cual se puede visualizar en la Figura 4-8. En este se utilizan dos posiciones bien conocidas para realizar los cálculos, P corresponde al Polo Norte celestial, Z al zenit local y X la posición del objeto al cual se quiere conseguir sus valores en coordenadas ecuatoriales. Lo siguiente es utilizar las ecuaciones de la (4-3) a la (4-6), en las cuales se tiene que  $\delta$  es el ángulo DEC,  $\alpha$  el ángulo de AR,  $a$  es el ángulo de altitud,  $A$  el ángulo de azimutal,  $\phi$  el ángulo de latitud,  $H$  es lo que se conoce como el ángulo de la hora local y por último LST es la hora sideral local.

$$\sin(\delta) = \sin(a) \cdot \sin(\phi) + \cos(a) \cdot \cos(\phi) \cdot \cos(A) \quad (4-3)$$

$$\sin(H) = -\sin(A) \cdot \cos(a) / \cos(\delta) \quad (4-4)$$

$$LST = GST - Longitud\ Oeste \quad (4-5)$$

$$\alpha = LST - H \quad (4-6)$$

El método de utilización de las ecuaciones anteriores consiste en utilizar (4-3) para conseguir  $\delta$  y luego utilizar (4-4) para conseguir  $H$ . Luego con la ecuación (3-10) es posible conseguir el valor de GST con el cual se puede utilizar (4-5) para conseguir LST y posteriormente (4-6) para obtener  $\alpha$ .



## 5 Implementación y resultados

En este capítulo se describen los instrumentos utilizados en los dos tipos de pruebas realizados para conseguir detecciones. Luego, se presentan los resultados obtenidos en cada una de ellas haciendo énfasis en los objetos detectados, los no detectados, los objetos que se pudieron emparejar con el catálogo y los que no.

### 5.1 Implementación

Para la totalidad de grabaciones la cámara se utiliza conectada a un telescopio guía Askar FMA180, con especificaciones en la Tabla 5-1. Los telescopios guía son telescopios astronómicos de menor envergadura utilizados principalmente para la tarea de apuntar objetos celestes de manera más fácil, ya que mientras más grande el telescopio (mayor tamaño se refiere a una mayor distancia focal) menor será el campo de visión y por tanto apuntar se vuelve una tarea engorrosa.

Tabla 5-1 Características Lente astrógrafo Askar FMA 180 /4.5 APO

Características Askar FMA 180	
Diámetro optico	40 mm
Largo Focal	180 mm
Relación focal	F/4.5

La elección de este telescopio se debe a esto último, al tener menor distancia focal es posible capturar una mayor porción del cielo sobre todo considerando el pequeño tamaño que ya posee el sensor que solo disminuiría aún más al colocarlo en un telescopio de tamaño mayor. Para comprender esta característica es posible calcular el “*field of view*” (FOV) efectivo que posee el sistema óptico con la (5-1).

$$FOV_{vertical,horizontal} = 2 \cdot \tan^{-1}\left(\frac{Cant. \text{ pixeles}_{vertical,horizontal} \cdot \text{tamaño de pixel}}{2 \cdot f}\right) \quad (5-1)$$

De esta forma, se consiguen cuantos grados el sistema consigue captar, el cual corresponde a un FOV de  $3.06^\circ \times 2.29^\circ$ . Por otro lado, para poder realizar el cálculo necesario para la conversión de coordenadas horizontales a ecuatoriales es útil saber el IFOV con la ecuación (4-2). El IFOV se

ocupa sin diferencia para ambas direcciones y se suele representar en arcosegundos debido a su pequeña magnitud. El IFOV para este sistema óptico corresponde a 17.18875”.



Figura 5-1 Lente astrógrafo Askar FMA 180 /4.5 APO

En la mayor cantidad de grabaciones realizadas se utiliza este telescopio montado en un trípode convencional con una montura ajustable. Fijar esta configuración de forma correcta es bastante complejo debido al peso y por tanto siempre se incurrirá en un error al tratar de apuntar algún objeto. La metodología utilizada en la captura de satélites por este medio siguió los siguientes pasos:

- Iniciar el programa para visualizar los eventos generados por la cámara.
- Hacer el ajuste del foco del lente del telescopio para enfocar. Para esto se utiliza un objeto lejano y se trata de hacer que la imagen posea bordes los más finos posibles.
- Hacer el ajuste de los bias si es necesario. Debido a la reiteración al realizar este tipo de prueba se llegó a valores estándares que funcionaron eficientemente la mayoría de las veces, por lo que este ajuste se puede realizar previamente ajustando en el código del programa.
- Apuntar una estrella o planeta brillante el cual se pueda visualizar y el cual se pueda encontrar en Stellarium para conocer la altitud y azimutal actual. Luego, ingresar dichos valores al programa presionando ‘a’.
- Comenzar la grabación presionando la letra ‘g’.

Por otro lado, se realizarán otras pruebas anclando el sistema anterior a un telescopio de mayor tamaño y una cámara con sensor CMOS que realiza cuadros con una resolución mucho mayor. Estas pruebas permitirán contrastar los resultados de las detecciones con las imágenes tomadas con esta cámara, pero además todo el setup es montado en una montura ecuatorial, un aparato que es alineado en paralelo con el eje de rotación de la tierra y de esta forma ayuda a compensar el movimiento de esta, por lo que al apuntar un objeto este permanece enfocado como si estuviese quieto en la imagen en todo momento. A continuación, se presentan las características de los implementos mencionados anteriormente.

Tabla 5-2 Especificaciones Cámara QHY600M

Características Cámara QHY600M	
Tamaño píxel	$3.76 \mu\text{m} \times 3.76 \mu\text{m}$
Cantidad de pixeles	9756 * 6388



a)



b)

Figura 5-2 a) Cámara QHY600M. b) Sharpstar refractor apocromático 61EDPHII

Con los datos entregados en las tablas anteriores es posible conseguir el FOV y el IFOV de la misma forma que se consiguió para la primera configuración. Para este sistema óptico se tiene: FOV de  $7.49^\circ \times 5.00^\circ$  y IFOV de  $2.82''$ . Se debe hacer notar la gran diferencia entre la cámara convencional y la cámara basada en eventos, siendo el sistema de esta primera cámara un sistema que logra captar no solo una mayor porción del cielo (mayor FOV), sino que también puede resolver objetos más pequeños en sus imágenes (menor IFOV).

Tabla 5-3 Especificaciones Sharpstar refractor apocromático 61EDPHII

Características Sharpstar refractor apocromático 61EDPHII	
Diámetro óptico	61 mm
Largo Focal	275 mm
Relación focal	F/4.5

Para esta ultima configuración, que se puede observar montada en la Figura 5-3, se consiguen distintos valores para las detecciones debido a la diferencia que existe en el método de captura. Se ha quitado la transformación a coordenadas ecuatoriales debido a que el funcionamiento con la montura ecuatorial trabaja de manera distinta. Hacer una implementación con este dispositivo en este caso sería lo ideal, pero no se dispuso de este para el desarrollo de este proyecto. Es así como en las detecciones donde debería estar el valor en coordenadas ecuatoriales es reemplazado por la localización en el espacio de los pixeles como se puede ver Figura 5-4.



Figura 5-3 Montaje de la configuración con ambas cámaras, la de eventos y la cámara a cuadros de alta resolución.

Para las pruebas realizadas en ambos casos las grabaciones se localizaron en las horas correspondientes a los crepúsculos (cuando el sol se pone o comienza a salir). Esto se debe a que en estas horas se produce el ángulo idóneo entre el satélite, el observador y el sol para que la luz pueda ser captada. Estas condiciones se dan solo un par de horas luego de la puesta de sol y antes del amanecer, ya que en otras condiciones la luz reflejada por el sol ensucia la captura o la Tierra eclipsa al sol y no permite que los satélites u otros objetos reflejen luz solar, cuanto tiempo se pueda extender esta condición depende de la latitud geográfica en la que se encuentre el observador, la época del año y la altura del satélite.

```
Detection at 2021-11-24-23_29_47
Tiempo deteccion 2: 2.680040
Punto1 X/Y : 351 / 265 | Tiempo : 2.000000h 29.000000m 43.347746s
Punto2 X/Y : 349 / 375 | Tiempo : 2.000000h 29.000000m 46.027786s
```

Figura 5-4 Ejemplo de mensaje de detección de objetos con la configuración de ambas cámaras.

Por otro lado, utilizando Stellarium para saber aproximadamente por donde circularían una cantidad determinada de satélites es que en ambos tipos de pruebas se solía cambiar regularmente el objetivo para así tratar de capturar y detectar la mayor cantidad de objetos, sobre todo considerando el pequeño FOV de la cámara de eventos.

## 5.2 Resultados

Se han realizado múltiples capturas a lo largo de varios días de trabajo, las detecciones que se muestran a continuación se han conseguido por medio del uso de la aplicación en tiempo real y los emparejamientos se han realizado posterior a las capturas utilizando las imágenes de las



También existen objetos capturados, pero no detectados como el de la Figura 5-7 estos objetos son muy difíciles de captar en tiempo real incluso a simple vista, esta imagen fue extraída de un análisis posterior donde se integran visualmente más eventos en cada cuadro, por lo que se puede considerar que en condiciones normales debe ser compleja la tarea de correlacionar dichos puntos. Lo anterior se puede deber a que este tipo de objeto es muy pequeño o está a demasiada altitud como para no poder pasar la primera prueba con la transformada de Hough. Tampoco hay que negar que las condiciones meteorológicas pueden empeorar el rendimiento, considerando en estas no solo las nubes sino el viento de igual manera.

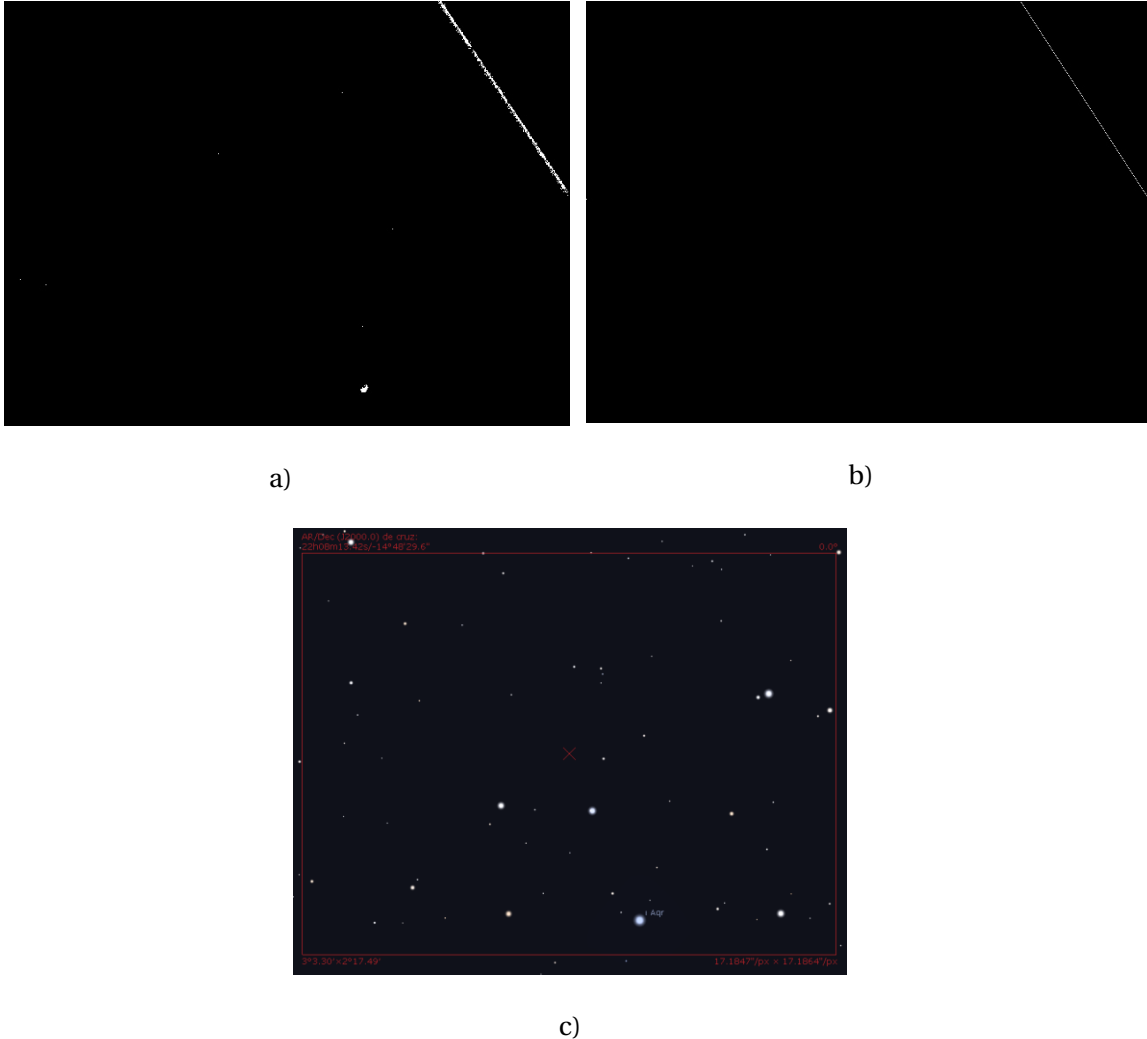


Figura 5-6 Detección de un objeto en órbita el día 15/11/21 a las 21:58:28. a) Imagen de integración, trayectoria descrita por el satélite en la escena. b) Detección de líneas creadas por medio de la transformada Hough. c) Imagen de Stellarium a las 21:58:28.

Al tratar de detectar basura espacial sucede algo muy similar a lo anterior, donde la mayoría de los objetos no son percibidos por la cámara debido seguramente a su mucho menor tamaño el cual el sistema óptico no alcanza a resolver en el sensor. Otros si que son percibidos, pero muy tenuemente, esto último podría hacer pensar que ajustar de mejor manera los bias o los

parámetros internos de la aplicación resolvería este problema, pero realizar esto no es tarea fácil y en la mayoría de los casos hacer al sistema más sensible conflictúa provocando en la salida mayor cantidad de ruido lo que genera que el desempeño del detector sea mucho peor.

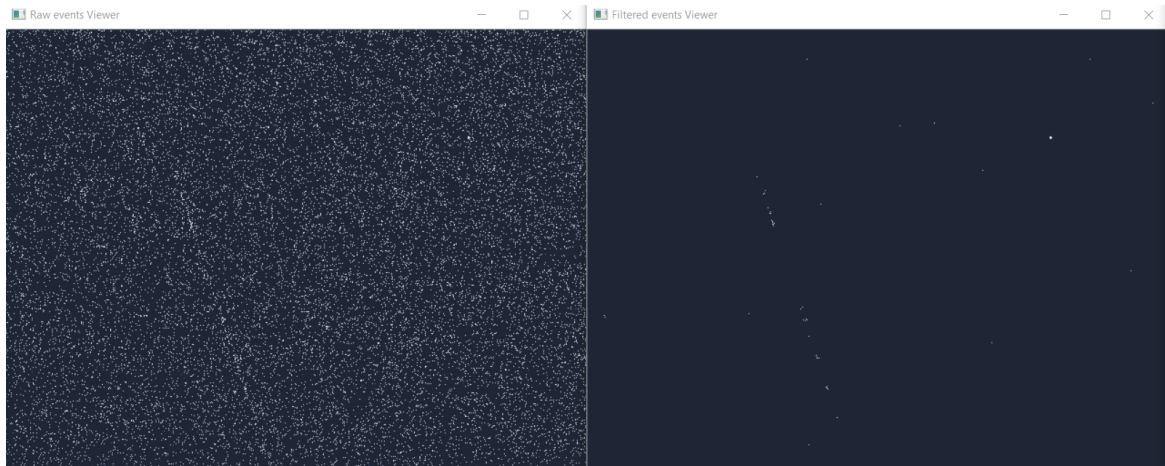


Figura 5-7 Ejemplo de objeto no detectado

En el apéndice A se incluyen algunas de las detecciones realizadas en tiempo real con la aplicación en cuestión. Se incluyen los valores de en coordenadas ecuatoriales de la detección y si es que llegase a poder realizarse un emparejamiento con Stellarium se incluyen también los valores de estos elementos junto con imágenes de la posición en los dos puntos calculados, de una forma similar a la Figura 5-5.

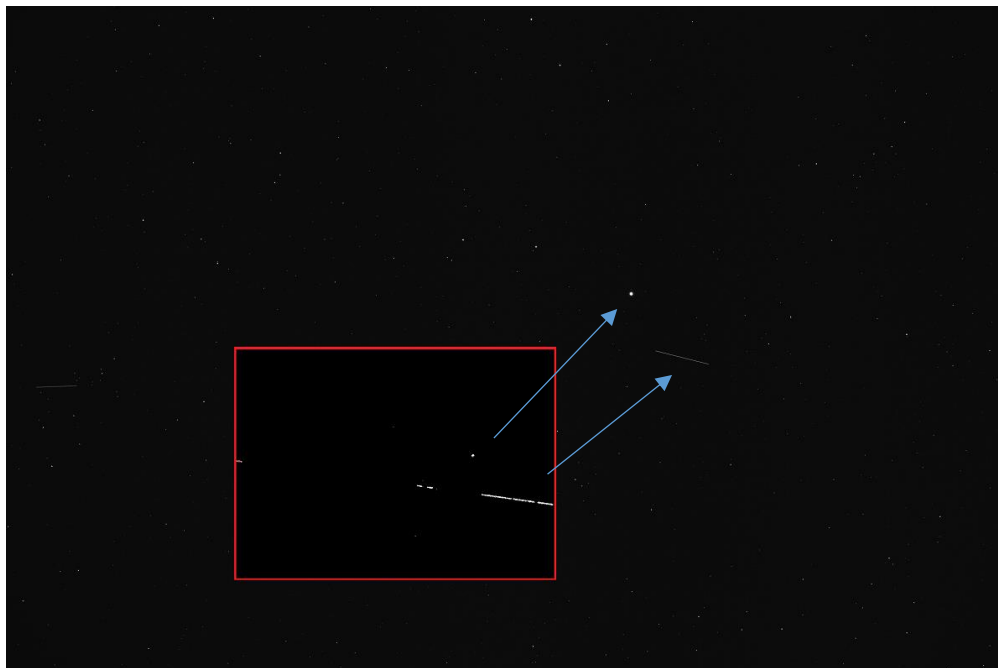


Figura 5-8 Imágenes de la detección del satélite Iridium 139 el 10/11/21 a las 22:32:16. La imagen grande corresponde a la captura con la cámara de alta resolución y la con bordes rojos la captura de la cámara de eventos.

Por último, en la Figura 5-8 se pueden ver algunos resultados preliminares al utilizar el setup con la cámara de ultra resolución, el telescopio y la montura ecuatorial. Se nota claramente la diferencia en escala entre ambos sistemas.

Aunque la cámara de alta resolución puede abarcar un espacio más amplio necesita de varias etapas para generar una imagen que permita una correcta captura de un objeto, etapas que tienen un costo mucho mayor en tiempo, al contrario, la cámara de eventos genera una representación de alto contraste sin ni un preámbulo.

Algunos inconvenientes que surgieron fueron el ajuste de los bias y los parámetros internos del programa. Como la aplicación se utilizó en su mayoría con la configuración simple los dos tipos de parámetros mencionados están optimizados para su uso de aquel modo, esto da cuenta de la dificultad y las exigencias que requiere conocer y ajustar dichos parámetros para cada caso de funcionamiento. Esto desalienta la utilización de este dispositivo en general puesto que este tipo de ajustes no es una tarea trivial y se realizó a través de varios días de trabajo.

En el apéndice A se presentan detecciones realizadas con esta prueba, pero a diferencia del caso anterior no se presentan los datos referentes a las coordenadas ecuatoriales debido a las consideraciones presentadas anteriormente.



## Discusión y conclusiones

En este documento se ha presentado la creación de una aplicación para la detección de objetos en órbitas cercanas a la Tierra, utilizando como medio de captura una cámara basada en eventos, un tipo de sensor novedoso que no posee los mismos problemas que las cámaras convencionales generar una señal de bajo ancho de banda, baja cantidad de redundancia de información y ocupar poca energía.

Se ha demostrado que con las configuraciones presentadas es posible conseguir detecciones de objetos con mucho contraste para satélites en orbitas LEO, almacenar estas detecciones ocupando muy poco espacio en disco y poder luego de forma “off-line” realizar un emparejamiento utilizando un software que se actualiza con los principales catálogos internacionales de trayectorias de satélites y basura espacial.

Esto ultimo se ha realizado a pesar de la gran limitante que posee la configuración estudiada en este documento, el campo de visión efectivo que posee esta cámara. El pequeño tamaño del sensor evita la posibilidad de utilizar telescopios de mayor envergadura que permitan poder visualizar objetos más lejanos o pequeños, por consiguiente, pensar en utilizar lentes “*wide angle*” nos podría brindar un mayor FOV, pero incluso podría verse limitada esta configuración ya que el sensor solo recibiría parte de este campo. El pequeño tamaño de estas cámaras se debe principalmente al poco tiempo de desarrollo que poseen, que también explican sus alto precios y la baja disponibilidad de estos sensores.

Por otro lado, al tratar de detectar basura espacial los resultados han sufrido debido a las limitaciones anteriores. La mayoría de los objetos pertenecientes a estos grupos pasa desapercibido al sensor, solo una pequeña cantidad de basura se reporta haber podido ser avistada en tiempo real, pero debido a su tamaño, no ha superado el filtro de ruido de la etapa de decodificación. Esto puede hacer pensar que bajo ciertas condiciones podría ser posible detectar estos objetos, pero lo cierto es que la aplicación en su estado actual trabaja de forma sensible y aumentar esta sensibilidad causa que el ruido incremente y por tanto la señal y la efectividad de la detección empeoren. Al ajustar tanto los bias como los parámetros de funcionamiento de la aplicación se entra a una compensación entre que nivel de ruido se quiere en la salida contra que tan fino se quiere ser con respecto a los eventos capturados.

Continuando con el tema anterior, el ajuste de los bias no es una tarea trivial como se ha mencionado en varias ocasiones a través de este informe, en la literatura es un problema recurrente en multitud de aplicaciones y no existe trabajo académico acerca de esto. Este puede ser un interesante punto para explorar y con el cual se podrían mejorar muchas de las aplicaciones, incluyendo esta, creadas a partir de estas cámaras. Particularmente, en este proyecto el ajuste se hizo experimentando noche a noche y se puede mencionar que los valores conseguidos funcionan bien en condiciones meteorológicas normales y con el setup simplificado, con el que se tiene mayor experiencia.

Además, se reporta al haber realizado las pruebas con el setup en conjunto a la cámara de alta resolución se presentaron problemas debido a los valores de bias y ajuste de parámetros. Se cree que el principal inconveniente sufrido se debe a que la montura ecuatorial permanece de manera más estática que el trípode corriente que se utiliza en el setup simple. Este trípode al no ser tan estable puede generar un pequeño movimiento que ayuda de cierta manera a la generación de eventos en la cámara de eventos. Este problema al tratar de extrapolar el ajuste conseguido con un experimento en otro da cuenta de lo problemático que pueden llegar a ser los bias y como el diseño de una solución funciona para el problema al cual fue planeado.

Utilizando la primera y la más recurrente configuración implementada se incurre en varios factores de error. El primero se trata de la forma de apuntar con el setup, la montura no posee ninguna forma de asegurar que la cámara se encuentre recta, al apuntar es imposible dejar centrado un objeto de interés y finalmente al obtener los valores de altitud y azimutal desde Stellarium se aproxima el conocer verdaderamente estos datos.

Es por eso que si se piensa en crear una aplicación para detectar satélites de forma más robusta la adición de una montura ecuatorial debería ser un paso obligatorio. Estos dispositivos permiten saltarse las etapas referentes a la referenciación de las coordenadas en las observaciones, además, ayudan a apuntar a los cuerpos celestes que se quieran, pero por lejos lo más importante es que mantienen una escena lo más fija posible y por tanto el movimiento relativo de estrellas no quedaría plasmado en los resultados de las detecciones, haciendo el diseño de la aplicación una tarea más simple.

Con referencia a los resultados en conjunto a la cámara a cuadros de alta resolución, es imposible no notar la gran diferencia en el campo de visión entre una cámara y la otra. Los resultados con esta cámara son variados, en ciertas ocasiones las trayectorias de los satélites son muy notorias y en otras casi no se perciben a simple vista. Esto último dependerá siempre de si el satélite está reflejando suficiente luz que la cámara pueda captar al hacer la integración, por consiguiente, se tendrán resultados de distintas calidades, el problema con esto ultimo es que distintas calidades en un programa significan generar procesamientos especiales para estandarizar la salida, lo que complejiza los procesos. La cámara de eventos procesa la integración con máximo de contraste en todo momento, si es que recibió la cantidad adecuada de eventos desde una fuente. Por otro lado, un mayor tiempo de integración a veces soluciona el problema de la visualización de las detecciones, pero también genera que las imágenes generadas pesen más y llenan la memoria

ram rápidamente, sin mencionar que es posible terminar con una imagen muy clara en la cual no se pueda distinguir nada.

Existen, además factores de los que no se pueden dar cuenta en este documento acerca del procesamiento y las imágenes obtenidas con la cámara de alta resolución. El procesamiento es lento debido a la gran cantidad de datos que se deben procesar, lo que resulta un poco extraño considerando en que se graba una escena fundamentalmente “*sparse*” y quieta. Además, las imágenes pesan bastante en disco, entre 100[mb] y 200[mb] lo que equivale a unos 30 min de video con los archivos .RAW, una diferencia sustancial en lo que de almacenamiento se trata.

La aplicación tratada en este proyecto es solo un acercamiento a la solución para el problema de mantener una activa vigilancia de los objetos que orbitan nuestros cielos. Para poder conseguir una solución más factible es necesario primero que todo resolver el problema del bajo FOV que posee el sistema utilizado en este proyecto. Para tratar de solventar este problema se podrían utilizar arreglos de cámaras de este tipo (ya que sensores de mayor tamaño aún puede ser algo pronto) o incluso hacer un primer tipo de detección con cámaras con alto FOV y luego con una cámara de eventos en una montura de rápido movimiento seguir estos objetos mientras sea posible. Además, como se menciono anteriormente una montura con referencia de coordenadas es un paso indispensable ya que esto permite precisión a la hora de apuntar objetos y de esta forma conseguir valores referenciados exactos que permitan saber con certeza si un objeto esta cambiando su orbita de forma preocupante.

Por último, en el presente proyecto se han ocupado las ventajas de una cámara basada en eventos en cada una de las etapas de la aplicación creada. Estas demuestran un gran potencial que puede seguir siendo explotado en un futuro, cuando el desarrollo y el estudio de estos dispositivos avance aun más.

# Bibliografía



- [1] European Space Agency , «Types of Orbits,» [En línea]. Available: [http://www.esa.int/Enabling\\_Support/Space\\_Transportation/Types\\_of\\_orbits](http://www.esa.int/Enabling_Support/Space_Transportation/Types_of_orbits). [Último acceso: 30 03 2020].
- [2] NASA CubeSat Launch Initiative, «CubeSat 101 Basic Concepts and Processes for First-Time CubeSat,» 2017.
- [3] D. J. Kessler, B. G. Cour-Palais, «Collision frequency of artificial satellites: The creation of a debris belt,» *Journal of Geophysical Research*, vol. 83, n° A6, p. 2637, 1978.
- [4] European Space Agency, «[https://www.esa.int/Safety\\_Security/Space\\_Debris/Space\\_debris\\_by\\_the\\_numbers](https://www.esa.int/Safety_Security/Space_Debris/Space_debris_by_the_numbers),» [En línea]. [Último acceso: 15 04 2021].
- [5] NASA Orbital Debris Program Office, «Frequently Asked Questions,» [En línea]. Available: <https://orbitaldebris.jsc.nasa.gov/faq/>. [Último acceso: 05 05 2021].
- [6] J. Hyde, E. Christiansen, D. Lear, J. Kerr, F. Lyons, J. Herrin, S. Ryan, «Investigation of Shuttle Radiator Micro-Meteoroid Orbital Debris(MMOD) Damage,» de 2009, Palm Springs, California., 50th AIAA/ASME/ASCE/AHS/ASC Structures,Structural Dynamics and Materials Conference.
- [7] D. L. Oltrogge, S. Alfano, «The technical challenges of better Space Situational Awareness and Space,» *ournal of Space Safety Engineering*, vol. 6, n° 2, pp. 72-79, 2019.
- [8] Dvorsky, G., «A Chunk of Debris Nearly Hit SpaceX CrewDragon as It Carried Astronauts to the ISS,» 2021.
- [9] A. J. Tellis, «India's ASAT Test: An Incomplete Success,» Carnegie Endowment for International, 2021.

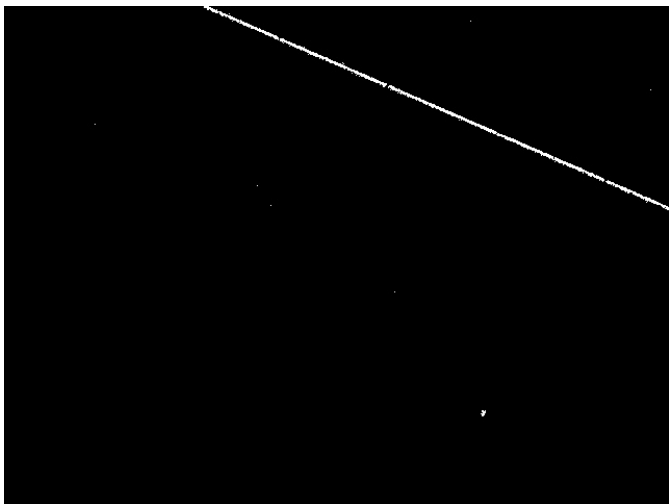
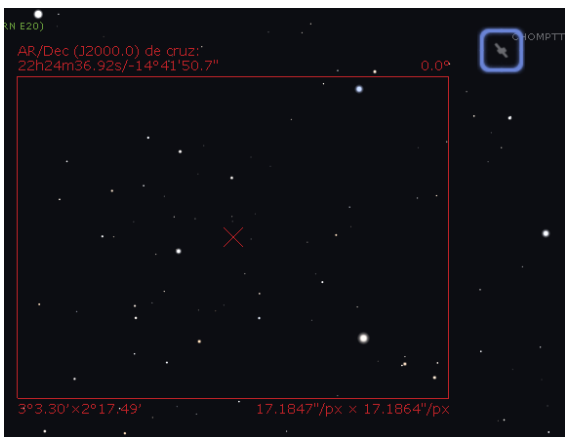

- 
- [10] G. Gallego, T. Delbrück, G. Orchard, C. Bartolozzi, B. Taba, A. Censi, S. Leutenegger, A. J. Davison, J., «Event-based Vision: A Survey,» *IEEE Transactions on*, pp. 1-1, 2020.
- [11] G. Cohen, S. Afshar, A. van Schaik, «Event-based Sensing for Space Situational Awareness,» de *Advanced Maui Optical and Space Surveillance Technologies*, Maui, 2017.
- [12] S. Afshar, A. P. Nicholson, A. van Schaik, G. Cohen, «Event-based Object Detection and Tracking for Space,» *IEEE Sensors Journal*, vol. 20, nº 24, pp. 15117-15132, 2020.
- [13] T. S. Kelso, «Orbital Coordinate Systems III,» Celestrak, 1996. [En línea]. Available: <https://www.celestrak.com/columns/v02n03/>.
- [14] 18 SPCS, «Space-Track.org,» [En línea]. Available: <https://www.space-track.org>.
- [15] T. S. Kelso, «Frequently Asked questions: Two-Line Element Set format,» Celestrak, 1998. [En línea]. Available: <https://www.celestrak.com/columns/v04n03/>.
- [16] Prophesee, «Metavision Intelligence Docs 2.1.0,» [En línea]. Available: <https://docs.prophesee.ai/2.1.0/index.html>. [Último acceso: Septiembre 2021].
- [17] OpenCV, «Hough Line Transform,» [En línea]. Available: [https://docs.opencv.org/3.4/d9/db0/tutorial\\_hough\\_lines.html](https://docs.opencv.org/3.4/d9/db0/tutorial_hough_lines.html). [Último acceso: 4 11 2021].




# A Catálogo de objetos detectados

Objeto detectado: Starlink - 1035		Imagen de la detección	
Tiempo de detección: 4.503576 [s]			
Distancia calculada: 9800.536921"			
Velocidad: 2176.167766 ["/s]			

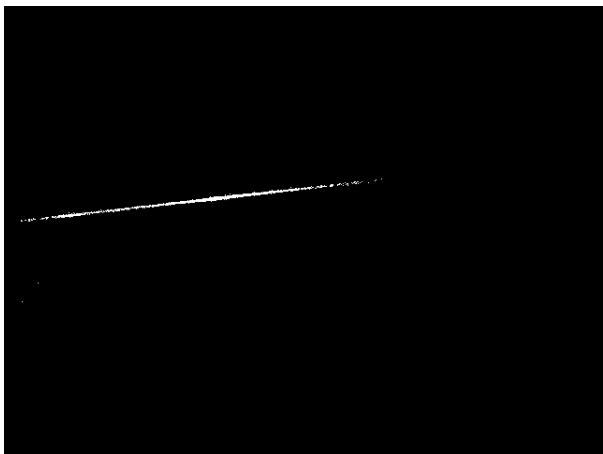
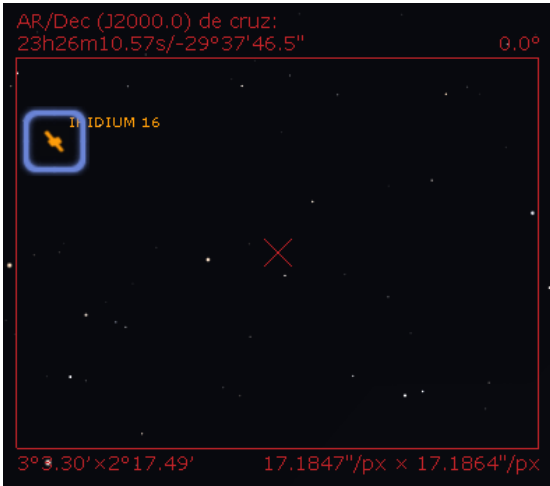
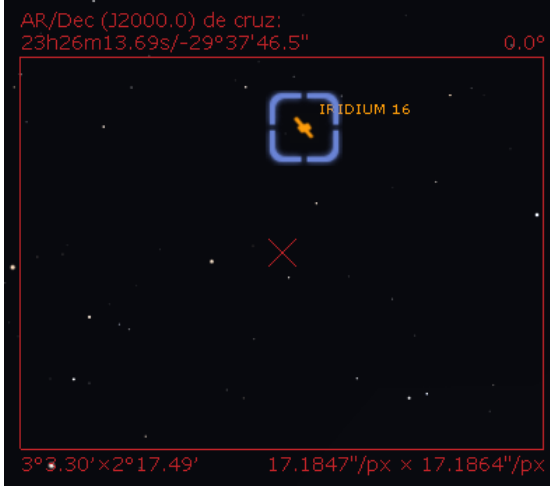
Punto 1:		Punto 2:	
Fecha: 25/11/21 - 21h 52m 25s		Fecha: 25/11/21 - 21h 52m 29s	
AR detectado: 21h 58m 36.81s	AR Stellarium: 21h 58m 11.84s	AR detectado: 22h 02m 3.8s	AR Stellarium: 22h 04m 52.18s
Dec detectado: -13° 58' 53.85"	Dec Stellarium: -15° 14' 53.1"	Dec detectado: -14° 47' 49.53"	Dec Stellarium: -13° 48' 00.2"

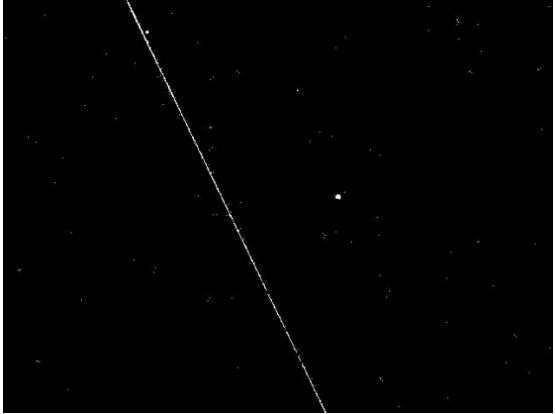
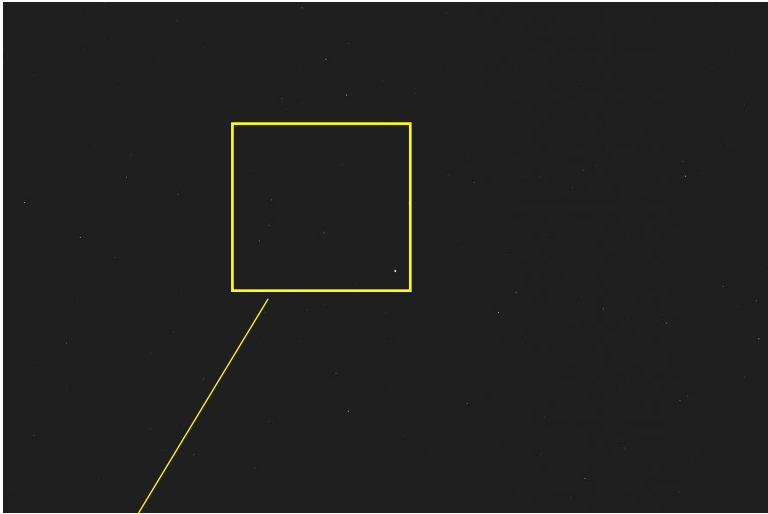


	
---	--


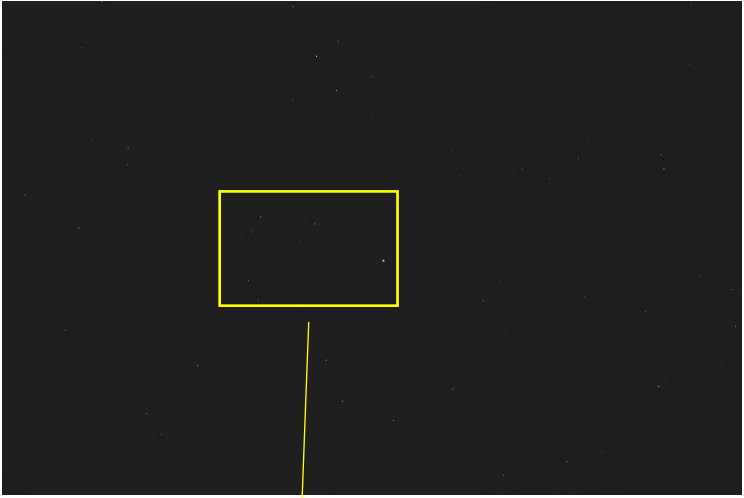
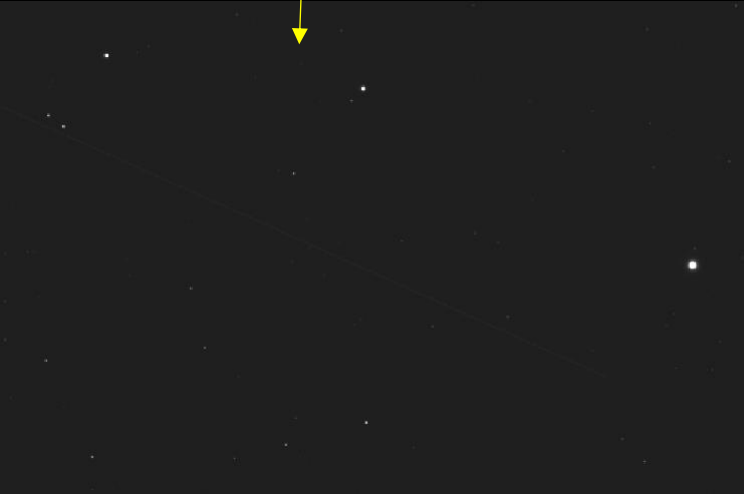
Objeto detectado: Chomptt		Imagen de la detección	
Tiempo de detección: 4.031039 [s]			
Distancia calculada: 7893.6705"			
Velocidad: 1958.222315 ["/s]			
Punto 1:		Punto 2:	
Fecha: 25/11/21 - 22h 15m 11s		Fecha: 25/11/21 - 22h 15m 15s	
AR detectado: 22h 28m 41.32s	AR Stellarium: 22h 47m 25.61s	AR detectado: 22h 26m 49.15s	AR Stellarium: 22h 45m 29.39s
Dec detectado: -14° 48' 56.43"	Dec Stellarium: 1° 10' 49.7"	Dec detectado: -13° 16' 53.72"	Dec Stellarium: -1° 02' 53.4"
			

Objeto detectado: Starlink-1128		<div>Imagen de la detección</div> 	
Tiempo de detección: 1.965026 [s]			
Distancia calculada: 3772.814072"			
Velocidad: 1919.981757 ["/s]			
Punto 1:		Punto 2:	
Fecha: 25/11/21 - 22h 52m 45s		Fecha: 25/11/21 - 22h 52m 47s	
AR detectado: 23h 00m 10.68s	AR Stellarium: 22h 55m 27.14s	AR detectado: 22h 57m 58.37s	AR Stellarium: 22h 57m 10.46s
Dec detectado: -14° 27' 18.52"	Dec Stellarium: -13° 59' 06.1"	Dec detectado: -13° 39' 24.17"	Dec Stellarium: -14° 54' 07.6"
<div>AR/Dec (J2000.0) de cruz: 23h01m53.38s/-14°42'14.2"0.0°</div> 		<div>AR/Dec (J2000.0) de cruz: 23h01m55.09s/-14°42'14.3"0.0°</div> 	



Objeto detectado: Iridium 16		Imagen de la detección	
Tiempo de detección: 3.157726 [s]			
Distancia calculada: 5348.003670"			
Velocidad: 1693.624992 ["/s]			
			
Punto 1:		Punto 2:	
Fecha: 25/11/21 - 23h 35m 50s		Fecha: 25/11/21 - 23h 35m 53s	
AR detectado: 23h 31m 29.37s	AR Stellarium: 23h 27m 13.54s	AR detectado: 23h 32m 35.23s	AR Stellarium: 23h 29m 37.06s
Dec detectado: -28° 42' 13.29"	Dec Stellarium: -31° 08' 32.5"	Dec detectado: -27° 27' 18.64"	Dec Stellarium: -29° 42' 47.9"
			

Objeto detectado: No presente en catálogo	<p>Imagen de la detección</p> 
Fecha Pto 1: 24/11/21 - 22h 24m 15s	
Fecha Pto 2: 24/11/21 - 22h 24m 25s	
	<p>Imagen alta resolución</p> 
	 

Objeto detectado: No presente en catálogo	Imagen de la detección 
Fecha Pto 1: 24/11/21 - 22h 26m 39s	
Fecha Pto 2: 24/11/21 - 22h 26m 45s	
	Imagen alta resolución  

Objeto detectado: Globalstar – M094	Imagen de la detección
Fecha Pto 1: 24/11/21 - 23h 32m 52s	
Fecha Pto 2: 24/11/21 - 22h 32m 59s	
	
Imagen alta resolución	
	
	