

**Centro de Investigación Científica y de Educación Superior de
Ensenada, Baja California**



**Doctorado en Ciencias
en Electrónica y Telecomunicaciones con orientación en
Telecomunicaciones**

**Diseño de un algoritmo de identificación de estrellas:
Aplicación a la estimación de la actitud de satélites pequeños**

Tesis

para cubrir parcialmente los requisitos necesarios para obtener el grado de
Doctor en Ciencias

Presenta:

Enrique Antonio Hernández Elías

Ensenada, Baja California, México

2017

Tesis defendida por

Enrique Antonio Hernández Elías

y aprobada por el siguiente Comité

Dr. Miguel Ángel Alonso Arévalo

Director de Tesis

Dr. Roberto Conte Galván

Dr. David Hilario Covarrubias Rosales

Dr. Edgar Leonel Chávez González

Dr. David Hiriart García

Dr. Julio César Rolón Garrido



Dr. Miguel Ángel Alonso Arévalo

Coordinador del Programa de Posgrado en Electrónica y Telecomunicaciones

Dra. Rufina Hernández Martínez

Directora de Estudios de Posgrado

Enrique Antonio Hernández Elías ©2017

Queda prohibida la reproducción parcial o total de esta obra sin el permiso formal y explícito del autor y director de la tesis.

Resumen de la tesis que presenta **Enrique Antonio Hernández Elías** como requisito parcial para la obtención del grado de Doctor en Ciencias en Electrónica y Telecomunicaciones.

Diseño de un algoritmo de identificación de estrellas: Aplicación a la estimación de la actitud de satélites pequeños

Resumen aprobado por:

Dr. Miguel Ángel Alonso Arévalo
Director de tesis

En la actualidad existen grandes oportunidades para lanzar nanosatélites¹ al espacio, esto es debido a la miniaturización de los componentes electrónicos por la explosión del uso de la tecnología en el mercado de los teléfonos inteligentes. Los nanosatélites están atrasados en la determinación de la actitud² precisa, ya que actualmente la determinación de la actitud de los nanosatélites está limitado a los sensores de sol, magnetómetros y velocidad de giro.

Los sensores que determinan la actitud de forma precisa como los sensores de estrellas tienen costos demasiado elevados para ser usados en nanosatélites.

Una parte importante de los sensores de estrellas son los algoritmos de reconocimiento de estrellas. El objetivo del proceso de identificación de estrellas es reconocer las estrellas utilizando los cuerpos celestes en su vecindad.

En ésta tesis se presenta un trabajo para la identificación de estrellas usando la invariante a la semejanza. El algoritmo propuesto define a un polígono para cada estrella de un catálogo de estrellas reducido, utilizando a los cuerpos celestes dentro del campo de visión como vértices, a este polígono se le calcula su invariante con la técnica derivada del método desarrollado por Chávez *et al.* (2013), que mapea al polígono a un *único* número complejo. A este mapeo no le afecta la transformación de la semejanza; esto es, el polígono bajo la transformación no es afectada por la rotación, el escalamiento o el desplazamiento.

Se realizó una validación experimental exhaustiva del algoritmo propuesto utilizando datos sintéticos generados a partir del catálogo de estrellas con ruido posicional uniformemente distribuido a cada una de las estrellas.

El método de identificación de estrellas que se presenta es muy robusto, teniendo un reconocimiento de hasta el 99.68 % cuando los niveles de ruido llegan hasta $\pm 424 \mu\text{radianes}$ en la posición de las estrellas. En las pruebas realizadas el algoritmo propuesto prueba que si un polígono concuerda con uno almacenado en la base de datos, éste siempre corresponde a la estrella bajo análisis. No se encontró ningún polígono equivocado.

También se probó que el algoritmo es más rápido comparado a otros algoritmos de identificación de estrellas y puede ser usado dentro de un nanosatélite.

¹Los nanosatélites son satélites que pesan de 1 a 10 Kg

²La palabra actitud se refiere a la orientación de cualquier objeto con respecto a un cuadro inercial de referencia u otra entidad, en esta tesis nos referiremos a la orientación del satélite respecto al planeta Tierra.

En su forma actual el algoritmo propuesto no puede reconocer polígonos para el caso cuando algunas de las estrellas de los polígonos creados desaparecen o cuando aparece una estrella falsa en la imagen y que no forma parte del catálogo de estrellas. En este caso el algoritmo solo indica que no encontró ningún polígono y no existe ningún caso de reconocimiento falso.

Palabras Clave: **Sensor de estrellas, seguidor de estrellas, identificación de estrellas y determinación de la actitud**

Abstract of the thesis presented by **Enrique Antonio Hernández Elías** as a partial requirement to obtain the Doctor in Philosophy degree in Electronics and Telecommunications.

Designing a star identification algorithm: Applied to small satellite attitude determination

Abstract approved by:

Dr. Miguel Ángel Alonso Arévalo
Thesis Director

At present, there are great opportunities to launch nanosatellites³ into space, this is due to the miniaturization of the electronic components pushed by the explosion of technology in the market of smartphones. The attitude⁴ determination capability of a nano satellite is limited by a lack of traditional high performance attitude sensors, since nowadays the determination of the attitude of the nanosatellites is limited to the sensors of Sun, magnetometers and speed of rotation. The sensors that determine the attitude with accuracy as the sensors of stars have costs too high to be used in the nanosatellites.

An important part of the star sensors are the star recognition algorithms. In the star identification process the goal is to recognize a star by using the celestial bodies in its vicinity as context. In this thesis we present an approach to star identification using similarity invariants. More specifically, the proposed algorithm defines a polygon for each star, using the neighboring celestial bodies in the field of view as vertices, its invariant is calculated to this polygon with the technique derived from the method developed by Chávez *et al.* (2013). The mapping is insensitive to similarity transformation; that is, the image of the polygon under the transformation is not affected by rotation, scaling or translations. Each polygon is associated with an *essentially unique* complex number. We perform an exhaustive experimental validation of the proposed algorithm using synthetic data generated from the star catalog with uniformly-distributed positional noise introduced to each star. The star identification method that we present is proven to be robust, achieving a recognition rate of 99.68 % when noise levels of up to $\pm 424 \mu\text{radians}$ are introduced to the location of the stars. In our tests the proposed algorithm proves that if a polygon match is found, it always corresponds to the star under analysis; no mismatches are found. Our algorithm execution speed was the fastest compared with others star identification algorithm. The conclusion is that our star identification algorithm can be used in nanosatellites. In its present form, however, our method cannot identify polygons in cases where there exist missing or false stars in the analyzed images, in those situations it only indicates that no match was found and there is no any case of false recognition.

Keywords: star sensor, star tracker, star identification, and attitude determination.

³Nanosatellites are satellites weighing 1 to 10 kg

⁴The word attitude refers to the orientation of any object with respect to an inertial frame of reference or another entity, in this thesis we will refer to the orientation of the satellite with respect to planet Earth.

Dedicatoria

Esta tesis se la dedico con mucho amor a mi hija Ana Vanessa, a Claudia, a mis nietos Leiza y Ricardito, a mi esposa Mely y a mis padres.

Agradecimientos

Agradezco a toda mi familia por todo su apoyo moral y amor que me brindaron, también quiero agradecer especialmente al Dr. Miguel Ángel Alonso por su guía, sus consejos y apoyo para la realización de mis estudios, así también al Dr. Edgar Chávez, al Dr. David Cobarrubias, al Dr. Roberto Conte, al Dr. David Hiriart y al Dr. Julio Rolón por su guía en la realización de mi tesis. También quiero agradecer especialmente al personal de biblioteca por su ayuda por darme las herramientas necesarias en forma de libros y artículos. También quiero agradecer a cada una de las personas de la parte administrativa del Departamento de Electrónica y Telecomunicaciones que también me brindaron su apoyo.

Agradezco a cada una de las personas del Centro de Investigación Científica y de Educación Superior de Ensenada (CICESE) que de una forma o de otra nos apoyan a que podamos realizar nuestros estudios y concluirlos sin mucho problema.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por brindarme la beca #279982 con ayuda económica para realizar mis estudios de Doctorado así como al mismo CICESE por su ayuda económica para la conclusión de mis estudios.

Para finalizar quiero agradecer a cada persona que no ha sido mencionada anteriormente por su apoyo.

Índice general

Resumen en español	ii
Resumen en inglés	iv
Dedicatoria	v
Agradecimientos	vi
Índice de figuras	ix
Índice de tablas	xi
Capítulo 1. Introducción	1
1.1 Actitud del satélite	1
1.1.1 Medición de la actitud	1
1.2 Planteamiento del problema	3
1.3 Objetivos del estudio de investigación	5
1.3.1 Objetivo general	5
1.3.2 Objetivos particulares	5
1.3.3 Objetivos específicos	5
1.4 Organización de la tesis	6
Capítulo 2. Coordenadas celestes y magnitud aparente	8
2.1 Introducción	8
2.2 Sistema de coordenadas	9
2.2.1 Propiedades del sistema de coordenadas de la esfera celeste	9
2.2.1.1 Trigonometría Esférica	11
2.2.1.2 Distancia angular entre dos estrellas	12
2.2.2 Sistema de coordenadas centradas a una nave espacial	13
2.2.2.1 Coordenadas fijas a una nave espacial	13
2.2.2.2 Coordenadas inerciales	14
2.2.2.3 Coordenadas definidas por órbita	16
2.2.3 Sistema de coordenadas ICRS	17
2.3 Paralaje	19
2.4 Magnitudes visual y aparente	20
2.4.1 Magnitud estelar	20
Capítulo 3. Componentes de un sensor de estrellas real	23
3.1 Lentes de cámaras	25
3.1.1 Calculando el ángulo de visión de una cámara	26
3.1.2 Campo de visión	27
3.2 Sensores de imágenes	29
3.2.1 Sensor CCD	29
3.2.2 sensor CMOS	30
3.2.3 Diferencias de los sensores CMOS versus sensores CCD	32
3.3 Procesamiento de imagen	33

3.3.1	Reducción de ruido de la imagen	33
3.3.2	Segmentación de la imagen	35
3.4	Agrupamiento de pixeles que representan estrellas	37
3.4.1	Agrupamiento de un escaneo	38
3.4.2	Agrupamiento de dos escaneos por mínimos y máximos	38
3.4.3	Agrupamiento circular	41
3.5	Cálculo del centroide	41
3.5.1	Método del primer momento o centro de masa	41
3.5.2	Método de ajuste Gaussiana simétrico	42
3.5.3	Método de ajuste Gaussiana circular	42
3.5.4	Método de la Convolución	43
3.5.5	Método del primer momento ponderado	43
3.6	Corrección Gamma	44
3.7	Introducción al catálogo de estrellas	44
3.7.1	Catálogo HYG	45
3.8	Algoritmos de Identificación de Estrellas	46
3.9	Transformación de las coordenadas xy en pixeles de la imagen a coordenadas esféricas	47
3.10	Determinación de la actitud del satélite	48
Capítulo 4.	Método de Identificación de Estrellas	50
4.1	Identificación de estrellas con polígonos	52
4.1.1	Invariante de semejanza para polígonos	53
4.2	Creación de la base de datos de invariante	57
4.2.1	Proyectando estrellas de la esfera celeste al plano xy	58
4.2.2	Creación de Polígonos con las proyecciones de las estrellas en el plano xy	59
4.2.3	Generación de base de datos de invariantes	62
4.3	Imagen sintética de estrellas	66
4.3.1	Creando imágenes sintéticas del catálogo de estrellas	66
4.3.2	Creando polígonos usando estrellas como vértices en el plano de la imagen sintética	68
4.4	Evaluación del algoritmo de identificación de estrellas	69
4.4.1	Evaluando el desempeño del algoritmo de identificación de estrellas	70
4.4.2	Magnitud aparente ≤ 6.0	71
4.4.2.1	Magnitud aparente ≤ 5.5	79
4.4.3	Validación del polígono reconocido	80
4.4.4	Comparación con otro método	82
4.4.5	Casos de estudio	84
4.4.5.1	Caso de estudio usando la estrella CS ID 20155 sin ruido	84
4.4.5.2	Caso de estudio usando la estrella con ID 4842 sin ruido	86
4.4.5.3	Caso de estudio usando la estrella con ID 521 con introducción de ruido de +/- 1.0 pixeles	88
Capítulo 5.	Conclusiones y trabajo a futuro	95
5.1	Conclusiones	95
5.2	Trabajo a Futuro	97
Literatura citada		98

Índice de figuras

1	Diferentes sensores para determinar la actitud.	2
2	Esfera celeste	8
3	Coordenadas Terrestres	9
4	Triángulo esférico formado por los puntos ABC	11
5	Ángulo esférico creado por las estrellas.	12
6	Coordenadas fijas a una nave espacial	13
7	Coordenadas celestes	14
8	Precesión de los equinoccios	15
9	Constelaciones zodiacales y eclíptica	15
10	Cuadro inercial	17
11	Sistema de coordenadas esféricas celestes centradas al Sol	17
12	Baricentro de un sistema de rotación de dos cuerpos en el espacio	18
13	Paralaje estelar	19
14	Sistema de determinación de la actitud utilizando una cámara con CCD para la captura de imagen	23
15	Diagrama a bloques de un sistema de detección de la actitud por medio de la cap- tura de imágenes con un sensor de visión y con algoritmos de identificación de estrellas típico	24
16	Diagrama esquemático del sistema de determinación de la actitud utilizando una cámara con CCD para la captura de imagen	24
17	Comparando el FOV rectangular <i>versus</i> uno circular. a) FOV rectangular no rotado, b) FOV rectangular rotado 45° , c) FOV rectangular rotado 60° , d) FOV circular.	28
18	cadena de baldes de agua	29
19	Transferencia de las cargas del CCD	30
20	Obtención de la señal de un sensor CMOS.	31
21	Imagen de una estrella con una ampliación de escala de 3000x. Misma imagen con división por renglones y columnas.	39
22	Imagen de dos estrellas con una ampliación de escala de 3000x seccionada por renglones y columnas con creación de arreglos.	39
23	Imagen de dos estrellas con una ampliación de escala de 3000x seccionada por ren- glones, columnas con creación de arreglos y además con numeración en el campo de número de estrella.	40
24	Geometría de un sensor de estrella.	47
25	Vectores observados y sus vectores de referencias	48
26	Imagen original de un lugar en el espacio y la imagen borrosa resultado del filtro promedio.	55
27	Visualización de centroides de las estrellas en la imagen de niveles de gris, e ima- gen blanco y rojo para mejorar la visualización de las estrellas dentro de la imagen.	56
28	20 diferentes escalamientos, rotaciones y desplazamientos al polígono original.	56
29	Estrellas de la esfera celeste proyectadas en el plano xy	59

30	Orden de los vértices en el plano xy .	60
31	Polígonos ordenados por distancia Euclidiana y ángulo a) polígonos generados los cuales sus formas son muy parecidas b) cuatro diferentes formas de polígonos generados utilizando a la misma estrella central CS 87044.	61
32	Figuras a), b) y c) son tres diferentes polígonos con la misma estrella CS 26529 generados en la misma área del cielo y ordenadas solamente con la distancia Euclidiana.	62
33	a) Comportamiento de la invariante debido al ruido introducido a los centroides de las estrellas, b) El conjunto de polígonos resultantes.	63
34	a) Comportamiento de la invariante debido al ruido introducido a los centroides de las estrellas, b) multipolígonos resultantes.	63
35	Gráfica de como las ecuaciones de transformación de coordenadas esféricas a píxeles son obtenidos.	67
36	Orden de vértices en el plano de la imagen sintética.	69
37	Imagen normal. Efecto de la radiación en el CCD.	82
38	a)Estrellas proyectadas en el plano xy amplificadas de la figura de la derecha, b)Estrellas proyectadas en el plano xy y los centroides $\hat{x}\hat{y}$ de la imagen sintética para la estrella ID 20155 como estrella central.	85
39	a)Estrellas proyectadas en el plano xy amplificadas de la figura de la derecha, b)Estrellas proyectadas en el plano xy y los centroides $\hat{x}\hat{y}$ de la imagen sintética para la estrella ID 4842 como estrella central.	87
40	Polígono creado con 3 vértices.	89
41	Polígono creado con 5 vértices.	90
42	Polígono formado con 6 vértices.	92
43	Polígono creado con 4 vértices.	93

Índice de tablas

1	Diferentes sistemas que determinan la actitud	3
2	Sistema de coordenadas inerciales comunes.	16
3	Precisión versus longitud focal y campo de visión para un sensor con pixel de 4.65 um	26
4	Desempeño de 2 sensores de estrellas comerciales marca Goodrich	27
5	Sensores CCD versus sensores CMOS.	32
6	Resultados del cálculo de la Invariante.	57
7	Datos de cuatro diferentes polígonos con la misma estrella CS 87044 generada en la misma área del espacio.	61
8	Datos de tres diferentes polígonos con la misma estrella CS 26529 generados en la misma área del cielo.	62
9	Número total de polígonos en la base de datos de invariantes como una función del número de vértices y el nivel de ruido utilizando estrellas con magnitud aparente ≤ 6.0	65
10	Número total de polígonos en la base de datos de invariantes como una función del número de vértices y el nivel de ruido utilizando estrellas con magnitud aparente ≤ 5.5	65
11	Máximo número de polígonos P_{max} en la base de datos de invariantes completa que un multipolígono puede tener como una función del nivel de ruido y el número de vértices utilizando estrellas con magnitud aparente ≤ 6.0	66
12	Máximo número de polígonos P_{max} en la base de datos de invariantes completa que un multipolígono puede tener como una función del nivel de ruido y el número de vértices utilizando estrellas con magnitud aparente ≤ 5.5	66
13	Comportamiento del algoritmo de identificación de estrellas después de ejecutarlo 10,000 veces por cada una de las 5,024 estrella del catálogo. La tabla presenta el porcentaje de colisiones para las primeras cuatro iteraciones. Se resaltan a los resultados correspondientes al número mayor de reconocimiento de polígonos.	72
14	Comportamiento del algoritmo de identificación de estrellas después de ejecutarlo 10,000 veces por cada una de las 5,024 estrella del catálogo. La tabla presenta el porcentaje de reconocimiento de polígonos para las primeras cinco iteraciones con ruido de 2.0 pixeles, 3.0 pixeles y 4.0 pixeles. Se resaltan a los resultados correspondientes al número mayor de reconocimiento de polígonos.	73
15	Comportamiento del algoritmo de identificación de estrellas después de ejecutarlo 10,000 veces por cada una de las 5,024 estrella del catálogo. La tabla presenta el porcentaje de reconocimiento de polígonos para la sexta y séptima iteración con ruido de 3.0 pixeles y 4.0 pixeles. Se resaltan a los resultados correspondientes al número mayor de reconocimiento de polígonos.	74
16	Porcentaje de colisiones y polígonos no reconocidos con ruido en la imagen (10,000 veces para cada una de las 5,024 estrellas del catálogo).	75

17	La tabla presenta el porcentaje de polígonos reconocidos usando polígonos de 5 lados con nivel de ruido introducido de ± 1.0	76
18	La tabla presenta el porcentaje de polígonos reconocidos usando polígonos de 7 lados con nivel de ruido introducido de ± 1.0	76
19	La tabla presenta el porcentaje de polígonos reconocidos usando polígonos de 9 lados con nivel de ruido introducido de ± 1.0	76
20	Comportamiento del algoritmo de identificación de estrellas después de ejecutarlo 10,000 veces por cada una de las 5,024 estrella del catálogo. La tabla presenta el porcentaje de reconocimiento de polígonos para la séptima iteración con ruido de 4.0 pixeles. Para esta prueba solo se cambia el valor de ℓ para polígonos de 9 vértices.	78
21	Comportamiento del algoritmo de identificación de estrellas después de ejecutarlo 10,000 veces por cada una de las 2,841 estrella del catálogo.	79
22	Rendimiento del algoritmo de identificación de estrellas después de ejecutarlo 10,000 veces a cada una de las 5,024 estrellas del catálogo cuando se pierde una o dos estrellas o si una estrella que no se encuentra en el catálogo es introducida en la imagen.	81
23	Comparación entre método propuesto y método de Wei et al.'s Wei <i>et al.</i> (2009).	82
24	Comparación de algoritmos de identificación de estrellas.	83
25	Coordenadas esféricas de CS ID 20155 y de sus estrellas vecinas.	84
26	Parte de las invariantes almacenadas en la base de datos de invariantes.	85
27	Centroides en la imagen sintética para CS ID 20155.	85
28	Las coordenadas esféricas para la estrella CS ID 4842.	86
29	Otra parte de la base de datos de invariantes.	87
30	Centroides en la imagen sintética para la estrella CS ID 4842.	87
31	Coordenadas esféricas de las estrellas dentro del FOV para la estrella con ID 521 como estrella central.	88
32	Proyección de coordenadas esféricas a coordenadas xy en pixeles.	89
33	Polígonos en colisiones para la invariante $-0.168496 + j0.250242$	90
34	Proyección de las coordenadas esféricas a coordenadas xy en pixeles.	91
35	Polígonos en colisión para la invariante $0.102727 - j1.014748$	91
36	Proyección de las coordenadas esféricas a coordenadas xy en pixeles.	92
37	Polígono coincidente para la invariante $0.389193 - j0.286978$	93
38	Proyección de las coordenadas esféricas al plano xy en pixeles.	94
39	Área de la invariante de la base de datos de polígonos con 4 vértices asociado a la estrella ID 521.	94

Capítulo 1. Introducción

En este capítulo se da una breve explicación para entender cada uno de los sensores utilizados para medir la actitud¹ del satélite, se realiza el planteamiento del problema y los objetivos de la investigación.

1.1. Actitud del satélite

La determinación de la actitud es el proceso de calcular la orientación de la nave espacial relativa a una referencia inercial o algún objeto de interés, tal como la Tierra. Para obtener la actitud de una nave en el espacio se necesitan varios tipos de sensores internos con procesamiento sofisticado de datos (Wertz, 2012).

La precisión de los cálculos para determinar la actitud está determinada usualmente por la combinación de los algoritmos y del hardware de la nave espacial.

Se necesita control de la actitud para evitar daño de los rayos solares a los componentes sensibles, esto nos ayuda a controlar la disipación de calor, tomar fotos de algún lugar de interés en tierra o al espacio, apuntar antenas direccionales, así como orientar a los paneles solares para la generación de potencia de forma óptima. También podemos controlar los cohetes utilizados para maniobrar las órbitas, etc (Wertz, 2012).

Los requerimientos de precisión del control de la actitud para las cargas de las naves espaciales son muy variables, para aplicaciones típicas de ingeniería son del orden de 1° y para las aplicaciones con telescopios y/u orientación de las antenas se necesitan mediciones abajo de 0.1° (Wertz, 2012). Las aplicaciones pueden demandar determinación de la actitud de menos de un segundo de arco ($1/3600^\circ$), tal como ha sido el caso del telescopio espacial Hubble que tiene una precisión de seguimiento rms (acrónimo del inglés root mean square) típica del orden de 2 a 5 milisegundos de arco o menos en toda una órbita (Fruchter *et al.*, 2009).

1.1.1. Medición de la actitud

La mayoría de las naves espaciales tienen sensores redundantes que pueden ser usados en varias combinaciones en el caso de fallas electrónicas o de sensores (Wertz, 2012).

¹La palabra actitud se refiere a la orientación de cualquier objeto con respecto a un cuadro inercial de referencia u otra entidad, en este documento nos referiremos a la orientación del satélite respecto al planeta Tierra.

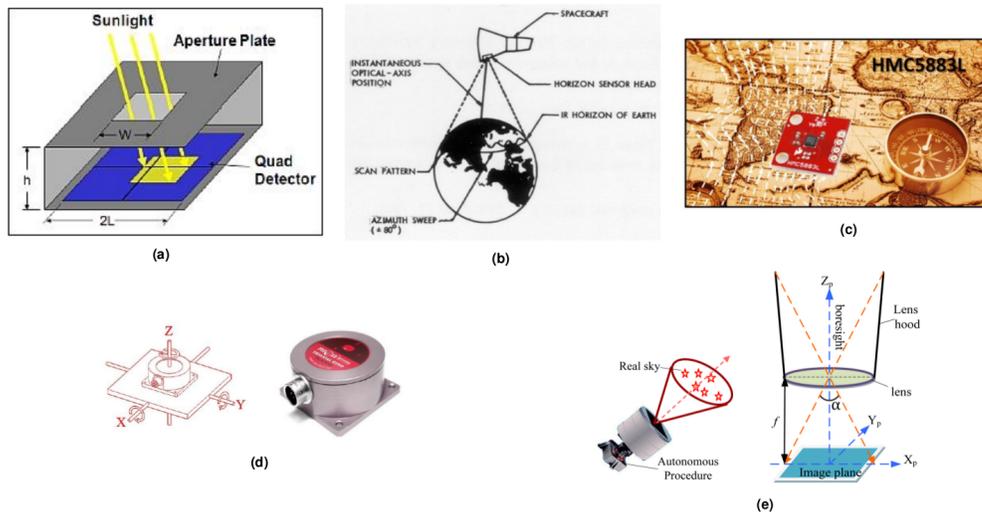


Figura 1: Diferentes sensores para determinar la actitud. a) Sensor de Sol (Janson *et al.*, 2012), b) sensor de horizonte (NASA, 1965), c) magnetómetro (Nelli, 2014), d) giróscopo (Alibaba, 2017) e) sensor de estrellas (Zhang *et al.*, 2012a)

Existen varios tipos de sensores para obtener la actitud que ayudan a controlar la actitud de la nave espacial:

- (a) Sensor de Sol digital, el cual utiliza una pequeña ranura y un patrón de rectángulos fotosensibles para medir el ángulo entre el eje de rotación y el Sol. El ángulo del Sol es una función de como cae la luz del Sol en el rectángulo fotosensible dentro del instrumento (Wertz, 2012), normalmente se utilizan dos sensores de Sol ortogonales.
- (b) Sensor de horizonte es un telescopio viendo hacia el horizonte de la Tierra y observa los límites del globo Terrestre con respecto al espacio (Wertz, 2012). También se le conoce como sensor de Tierra, normalmente se utiliza una cámara infrarroja para la detección de los bordes de la Tierra.
- (c) Magnetómetros, son sensores que miden la fuerza y dirección del campo magnético de la Tierra, solo son útiles cuando se usan abajo de los 6,000 Km.
- (d) Giróscopos, son dispositivos que miden la rotación en el espacio tridimensional sin dependencia de la medición por sensores externos. Un giróscopo se basa en la rotación de una masa (Rozelle, 2009).
- (e) Sensores de estrellas o seguidores de estrellas. Durante el proceso de la determinación de la actitud se utilizan sensores de estrellas, las estrellas en los marcos inerciales son considerados las referencias. Las imágenes son capturadas con una cámara CCD (Charge Coupled

Device), estas imágenes son linealizadas y se implementan una serie de pasos, los cuales son el procesamiento de la imagen, algoritmos de identificación de las estrellas junto al sistema de búsqueda dentro de la base de datos y el algoritmo para calcular la estimación de la actitud (Xu *et al.*, 2013).

A continuación se presenta una tabla en donde se pueden observar los diferentes sistemas que determinan la actitud.

Tabla 1: Diferentes sistemas que determinan la actitud (Liebe, 1992).

Tipo de Sensor	Precisión Teórica
Estrellas	1 arco segundo
Sol	1 arco minuto
Tierra (Horizonte)	6 arco minuto
RF beacon	1 arco minuto
Magnetómetro	30 arco minuto

Cuando una nave espacial hace una maniobra para ir de una orientación a otra, el giróscopo nos proporciona una medición del cambio de orientación y una buena estimación inicial de la nueva actitud. Esta estimación utiliza el dato del sensor de estrellas, para medir la orientación real de la nave y actualiza al giróscopo para eliminar los errores acumulados desde la medición anterior (Wertz, 2012).

Para la primera generación de sensores de estrellas solamente se capturaban pocas estrellas brillantes en el campo de visión por lo que se podían obtener las coordenadas de estas estrellas. Las coordenadas no estaban relacionadas al espacio inercial, y la información de la actitud tenía que ser calculada por un sistema externo. Esto fue debido a que la velocidad de cálculo del microcontrolador no era capaz de procesar la información. El rápido crecimiento de los sistemas de microprocesadores hizo que pudiéramos hacer los cálculos dentro de las naves. La segunda generación de los sensores de estrellas incluyen una cámara electrónica con la que captura todas las estrellas en una vecindad e incluye una microcomputadora. El sensor de estrellas reconoce los patrones de las estrellas dentro de su campo de visión, compara los resultados con un catálogo de estrellas almacenado en su memoria interna y determina la actitud (Rohác *et al.*, 2011).

1.2. Planteamiento del problema

En la actualidad existen grandes oportunidades para lanzar nanosatélites al espacio, esto es debido a la miniaturización de los componentes electrónicos por la explosión del uso de la tecno-

logía y de las baterías en el mercado de los teléfonos inteligentes. Para que los nanosatélites se puedan usar como plataformas tecnológicas y científicas, éstos deben ser capaces de obtener y controlar su actitud de forma precisa.

La determinación de la actitud de la mayoría de los nanosatélites está limitado a los sensores de Sol, magnetómetros, y a las mediciones inerciales de aceleración y velocidad de giro. Un juego de sensores de Sol pueden obtener la actitud de un nanosatélite relativamente precisa, pero solo operan en la luz del sol. Para los satélites que operan en la órbita baja de la tierra el 30 % de su órbita aproximadamente ocurre en la obscuridad. Actualmente existen muy pocas compañías que venden sensores de estrellas para nanosatélites con precios arriba de los \$50,000.00 dls por cada sensor, normalmente se ocupan dos o más sensores que no estén paralelos entre sí para poder identificar las estrellas de forma robusta a pesar de que uno de los sensores falle en la identificación de las estrellas por algún tipo de problemas con el algoritmo de identificación y las estrellas que captura o se quede ciego por estar viendo al Sol o la Luna.

Es importante mencionar que no existe un sensor que sea único para determinar la actitud del satélite y que nos ayude a direccionarlo de forma precisa. Como ejemplo tenemos que un magnetómetro no puede medir en que longitud de la Tierra se encuentra y solo debe de ser usado por abajo de los 6,000 Km de elevación. El sensor de horizonte mide los ángulos en que se encuentra el nadir de la Tierra, este sensor siempre es usado junto al sensor de Sol. El sensor de Sol mide dos ángulos de la posición del Sol, esto sucede siempre y cuando el satélite no se encuentre en eclipse con la Tierra, el giróscopo es un medidor local de la rotación del satélite y el sensor de estrellas determina la actitud en el espacio con mayor precisión comparada a los otros sensores pero uno de sus inconvenientes es que no determina la actitud del satélite con respecto a un punto de la Tierra. Por lo que los satélites llevan una combinación de sensores dependiendo de su aplicación.

Los sensores con algoritmos de identificación de estrellas son importantes en los sistemas satelitales para obtener su posición en el espacio con precisión, se necesita esta posición precisa junto con el sistema de control de posicionamiento para que los nanosatélites puedan realizar experimentos científicos más complejos que los que existen en este momento, por ejemplo: sistemas buscadores de exoplanetas, telescopios espaciales o captura de imágenes de locaciones en Tierra de forma precisa en cualquier actitud del nanosatélite aún durante el periodo de eclipse en que los nanosatélites atraviesan en órbita baja.

Científicamente son muy limitados los números de algoritmos de identificación de estrellas publicados que se pueden utilizar como parte de un sensor real para un nanosatélite, en algunos casos estos algoritmos son tan complejos para que puedan correr en el hardware de un nanosatélite, en otros casos los algoritmos desarrollados no pueden identificar un porcentaje alto de las estrellas en condiciones reales. Por lo que existen oportunidades de contribuir al estado del arte en los algoritmos de identificación de estrellas.

1.3. Objetivos del estudio de investigación

Se presentan a continuación los distintos objetivos de este trabajo de tesis.

1.3.1. Objetivo general

El objetivo general del presente trabajo es proponer un nuevo algoritmo para la identificación de estrellas combinando técnicas del estado del arte de algoritmos de reconocimiento de patrones. Estos algoritmos serán usados en el diseño de un sensor de identificación de estrellas real que puede ser llevado por un nanosatélite con la finalidad de permitir a éste orientarse con precisión en el espacio. Para poder llevar a cabo esta investigación será necesario entender cada uno de los algoritmos diseñados en la actualidad, desarrollar un nuevo algoritmo y simular cada uno de éstos.

1.3.2. Objetivos particulares

- Proponer una técnica novedosa que permita identificar las estrellas captadas por el sensor mediante algoritmos de reconocimientos de patrones junto con la búsqueda de catálogos de estrellas almacenadas en una base de datos.
- Simular y comparar con otros algoritmos de identificación de estrellas.

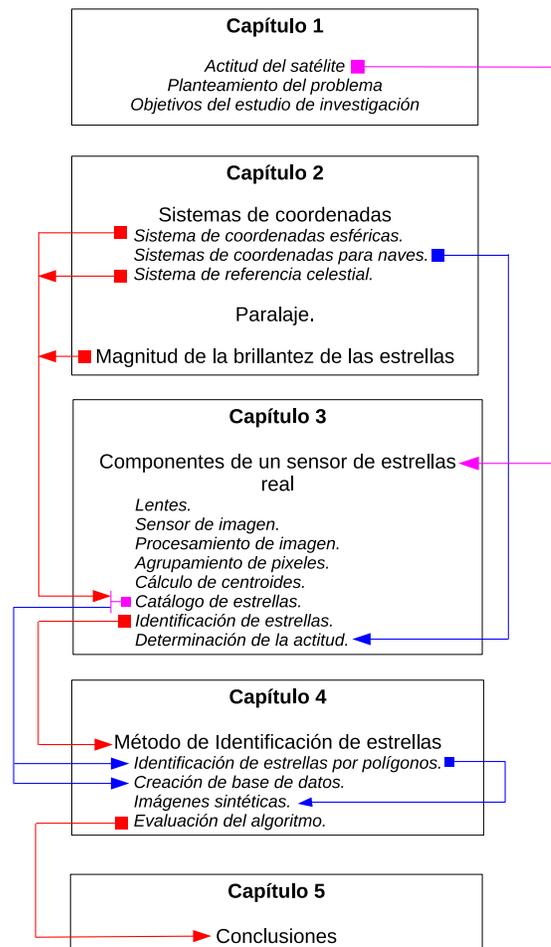
1.3.3. Objetivos específicos

Los objetivos específicos de este trabajo de tesis son:

- Seleccionar el tipo de sensor con lente adecuado para el uso en un nanosatélite tomando en cuenta el campo de visión, el tamaño de la imagen, velocidad de transferencia de datos, y tipo de interfaz.

- Captura de la imagen, linealización y desarrollo del algoritmo de preprocesamiento. El objetivo de este punto es obtener las posiciones de las estrellas sobre la imagen capturada con precisión.
- Investigación y Desarrollo de algoritmos para identificación de patrones que no haya sido usado en los algoritmos de identificación de estrellas y que sea rápido.
- Diseño de la base de datos del catálogo de estrellas de referencia. Éste consiste en la clasificación de la información del catalogo de estrellas en una base de datos, así como diseñar un buscador de datos veloz, es muy importante para poder obtener los vectores del cuadro de la cámara y del cuadro inercial de forma rápida.

1.4. Organización de la tesis



Esta tesis ha sido organizada tal como podemos observar en la disposición de la tesis mostrada anteriormente y a continuación se presenta una breve descripción de cada uno de los capítulos

del escrito la tesis:

En este capítulo 1. se da una breve explicación de los sensores que nos ayudan a determinar la actitud de un satélite.

En el capítulo 2. se presenta la teoría para entender las características de las estrellas que son importantes para su identificación, tales como los sistemas de coordenadas que se utilizan en la posición de las estrellas, trigonometría esférica así como los sistemas de coordenadas para la determinación de la actitud. También se presenta la teoría de lo que es el paralaje y de la magnitud visual.

En el capítulo 3. explica cada uno de los componentes de un sensor de estrellas real, como son las lentes, los tipos de sensores de imágenes, los algoritmos de procesamiento de imagen, el agrupamiento de píxeles, el cálculo de centroides, catálogo de estrellas, algoritmos de identificación de estrellas y la determinación de la actitud.

En el capítulo 4. se presenta la investigación y el desarrollo de un nuevo método de identificación de estrellas mapeando polígonos con vértices formados con estrellas a un número complejo, el cual se busca dentro de una base de datos generado para la identificación del polígono, con lo que se logra la identificación de las estrellas que lo forman.

Por último se presentan en el capítulo 5. las conclusiones y el trabajo a futuro.

Para realizar el estudio de los sensores de estrellas es necesario tener una base teórica la cual se presentará en el siguiente capítulo.

Capítulo 2. Coordenadas celestes y magnitud aparente

En este capítulo se presenta la teoría necesaria para entender el diseño completo de un sensor de estrellas, tales como los diferentes sistemas de coordenadas necesarias para interpretar a los catálogos de estrellas, y que también son útiles para la determinación de la actitud, además se introduce la trigonometría esférica necesaria para poder desarrollar y comprender algoritmos de identificación de estrellas. También se comenta sobre el paralaje de las estrellas y al final se explica la magnitud de la brillantez de las estrellas que es un parámetro importante dentro de los sensores de estrellas.

2.1. Introducción

Si observamos a las estrellas en una noche clara, se tiene la impresión que las estrellas se encuentran situadas en una esfera en el cual el observador es el centro de ésta, se puede estimar el ángulo formado entre dos estrellas con instrumentos para medir ángulos con gran precisión.

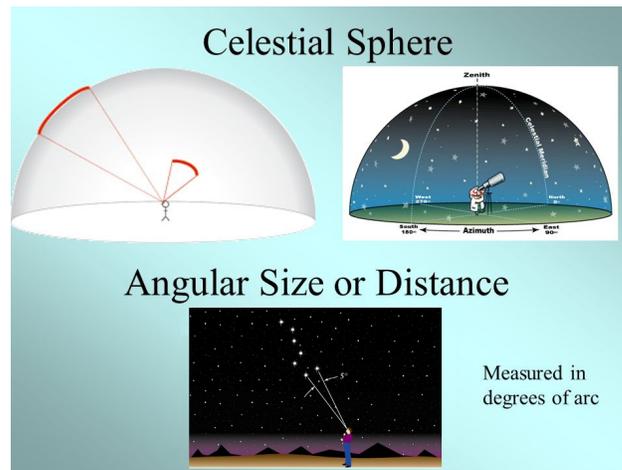


Figura 2: Esfera celeste (Auden, 2011) (Osborne, 2010) (Pogosian, 2017).

La astronomía de posición calcula las direcciones de las estrellas que se observan en el espacio y se definen con respecto a la posición sobre la superficie de la “esfera celeste”, en la cual existen líneas rectas imaginarias que van del observador a las estrellas intersectando la superficie de la esfera celeste, tal como podemos observar en la Figura 2. El radio de la esfera r es arbitrario (Smart y Green, 1977).

2.2. Sistema de coordenadas

Las mediciones en la esfera celeste utilizan sistemas de coordenadas esféricas, las cuales están definidas en términos de la esfera celeste unitaria. Las coordenadas esféricas pueden ser transformadas a coordenadas tridimensionales agregando una tercera variable, que sería el radio r , la distancia del centro del sistema de coordenadas al punto en cuestión (Wertz, 2012).

2.2.1. Propiedades del sistema de coordenadas de la esfera celeste

Los sistemas de coordenadas esféricas tienen dos polos diametralmente opuestos entre sí en la esfera celeste y un ecuador que es la mitad entre los polos. Cualquier plano que pasa por el centro de una esfera segmenta la superficie en un círculo el cual se le llama círculo máximo. Cualquier otro plano intersectando a la esfera pero que no pasa por el centro también segmentará la superficie en un círculo, el cual en este caso es llamado círculo menor. Los círculos máximos a través de los polos y la perpendicular al ecuador son llamados meridianos y los círculos menores a una distancia fija arriba o abajo del ecuador son llamadas paralelas (Wertz, 2012).

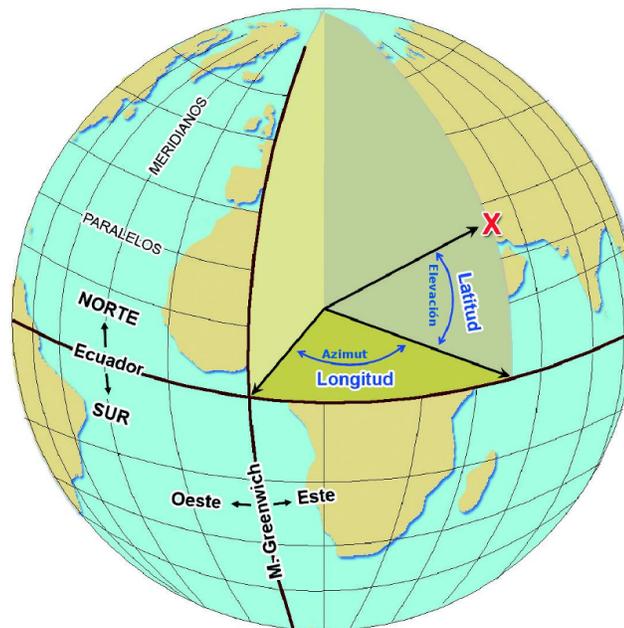


Figura 3: Coordenadas Terrestres (Race, 2014).

La posición de cualquier punto en la esfera está dado en términos de dos componentes equivalentes a la latitud y la longitud sobre la superficie de la Tierra, tal como podemos ver en la Figura 3. La latitud o elevación es la longitud del arco de arriba o de abajo al ecuador. El azimut o longitud

es la distancia angular alrededor del ecuador entre el meridiano pasando a través de un punto en particular y un meridiano de referencia arbitrario (Wertz, 2012).

Por ejemplo, el meridiano de referencia para la longitud sobre la superficie de la Tierra está pasando a través del centro del Observatorio Real de Greenwich en Londres. Se puede definir las posiciones de los puntos P_1 y P_2 en términos de azimut ϕ , y de la elevación λ , tal como podemos ver en la siguiente expresión (Wertz, 2012):

$$P_1 = (\phi_1, \lambda_1) = (75^\circ, 35^\circ) \quad P_2 = (\phi_2, \lambda_2) = (345^\circ, -10^\circ) \quad (1)$$

En la mayoría de los sistemas de coordenadas esféricas la coordenada del azimut es medida de 0° a 360° y el componente de elevación está medido de $+90^\circ$ a -90° . La intersección de los meridianos de referencias y el ecuador en cualquier sistema se le llama punto de referencia. Un grado de elevación es un grado de longitud de arco y la separación angular entre dos puntos en el mismo meridiano es solo la diferencia entre la elevación de los dos puntos. Esto es P_3 en $(75^\circ, 60^\circ)$ está a 25° de P_1 .

Sin embargo 1° de separación en azimut será menos de 1° de arco, excepto a lo largo del ecuador. El punto P_4 está a $(50^\circ, 35^\circ)$ es menos de 25° en longitud de arco de P_1 (Wertz, 2012).

La distorsión en el azimut es muy grande cerca del polo de cualquier sistema de coordenadas. En cualquiera de los polos, el azimut está indefinido. Un procedimiento alternativo para especificar la posición de un punto de la esfera celeste involucra a los tres componentes de un vector de longitud unitaria del centro de la esfera al punto sobre la superficie de la esfera. Ordinariamente, los ejes x , y y z del sistema de coordenadas rectangulares son definidas tal que el eje z está hacia el polo de $+90^\circ$ del sistema de coordenadas esféricas, el eje x va hacia el punto de referencia, y el eje y es perpendicular a x y a z tal que la medición en el sistema de coordenadas va de derecha a izquierda, (por ejemplo, para vectores unitarios a lo largo de x, y y z , $\hat{z} = \hat{x} \times \hat{y}$) (Wertz, 2012).

Cualquier sistema de coordenadas esféricas (o su equivalente rectangular) se especifica completamente indicando el polo positivo y cualquier meridiano de referencia. En la superficie de una esfera, se escogen arbitrariamente los polos y el meridiano principal, y cualquier punto en la esfera puede ser usado tal como el polo de un sistema de coordenadas esféricas. Para la Tierra, un sistema definido por el eje de rotación sería lo más conveniente. Sin embargo, para el cielo que se ve desde una nave espacial, existen una variedad de sistemas de coordenadas alternativos que

se pueden usar y se escoge al sistema de coordenadas dependiendo de su aplicación (Wertz, 2012).

Un ángulo esférico es formado con la intersección de dos círculos máximos. Si tenemos tres puntos sobre la superficie de una esfera, y además la esfera puede ser dividida tal que los tres puntos se encuentran en el mismo hemisferio, y si los puntos están unidos por arcos de círculos máximos (tal como los ángulos esféricos a , b y c de la Figura 4), la figura matemática que se obtiene es un triángulo esférico (Smart y Green, 1977).

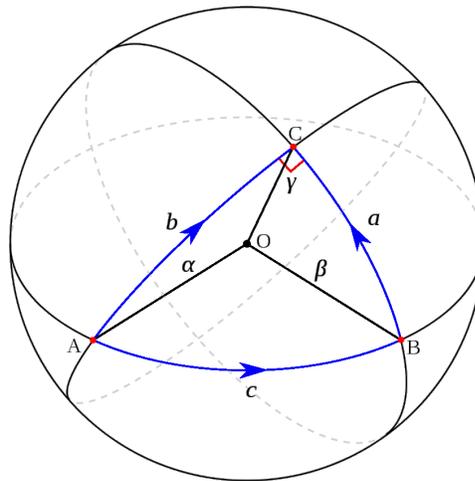


Figura 4: Triángulo esférico formado por los puntos ABC (Smart y Green, 1977).

2.2.1.1. Trigonometría Esférica

Si tenemos que los puntos ABC de la Figura 4 forman un triángulo esférico, por lo que los lados BC , CA , AB serán a , b y c respectivamente. El lado a es medido por el ángulo esférico BOC en donde O es el centro de la esfera y el arco de la esfera máxima BC , similarmente b y c están medidos respectivamente por los ángulos esféricos AOC y AOB . El ángulo interno BAC será denotado simplemente por A , el ángulo interno ABC será B y el ángulo interno BCA será C . La suma de los ángulos internos del triángulo esférico debe de ser mayor de 180° y menor de 540° . Los lados a, b, c y los ángulos internos de un triángulo esférico son menores a 180° . Una fórmula fundamental de la trigonometría esférica es la fórmula coseno, la cual está escrita a continuación (Smart y Green, 1977):

$$\cos(a) = \cos(b) \cos(c) + \sin(b) \sin(c) \cos(A). \quad (2)$$

Ésta ecuación es útil cuando se conoce el valor de dos de sus lados (ángulos esféricos) y el ángulo interno que le corresponde al lado desconocido. También es útil cuando se conocen el valor de los 3 lados pero no los ángulos internos del triángulo esférico.

Otra de las ecuaciones fundamentales de la trigonometría esférica es la fórmula del seno, la cual podemos ver a continuación (Smart y Green, 1977):

$$\frac{\sin(A)}{\sin(a)} = \frac{\sin(B)}{\sin(b)} = \frac{\sin(C)}{\sin(c)} \quad (3)$$

Esta ecuación se usa cuando se conoce el valor de dos de sus lados (ángulos esféricos) y el ángulo interno que le corresponde al lado conocido o de dos ángulos internos y el lado que le corresponde a uno de los ángulos internos conocido.

2.2.1.2. Distancia angular entre dos estrellas

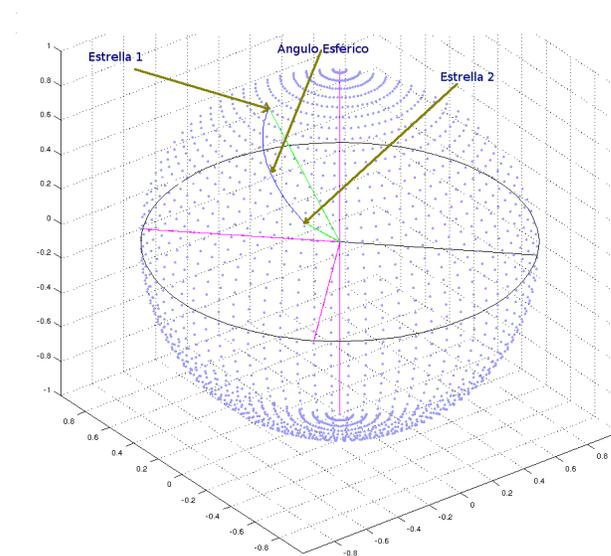


Figura 5: Ángulo esférico creado por las estrellas.

Tenemos que δ_1, α_1 y δ_2, α_2 son las coordenadas esféricas de las estrellas 1 y 2, y que $\Delta\delta, \Delta\alpha$ son las diferencias absolutas entre ellas; por lo que $\Delta\tau$, es la distancia angular o ángulo esférico

entre las dos estrellas tal como podemos ver en la Figura 5, la distancia angular entre las dos estrellas se puede calcular con la siguiente ecuación (Ball, 1915):

$$\Delta\tau = \arccos(\sin \delta_1 \cdot \sin \delta_2 + \cos \delta_1 \cdot \cos \delta_2 \cdot \cos(\Delta\alpha)). \quad (4)$$

2.2.2. Sistema de coordenadas centradas a una nave espacial

Hay tres tipos básicos de coordenadas centradas a la nave espacial:

- 1) Las coordenadas fijas a una nave espacial.
- 2) Las coordenadas inerciales.
- 3) Las coordenadas definidas por órbita.

2.2.2.1. Coordenadas fijas a una nave espacial

El sistema de coordenadas fijas a la nave espacial es el sistema en el cual se hacen las mediciones de la actitud.

Las coordenadas esféricas fijas a la nave espacial usan α para el azimut y δ para la elevación. Alternativamente, θ será usado para la coelevación; esto es $\theta = 90^\circ - \delta$. Para una nave espacial que está rotando, el polo positivo del sistema de coordenadas será el vector de rotación positivo. El meridiano de referencia se toma como el que pasa a través de un punto de referencia arbitrario en el plano de rotación el cual es el ecuador del sistema de coordenadas. También podemos trabajar este sistema con coordenadas rectangulares tal como podemos observar en la Figura 6 (Wertz, 2012).

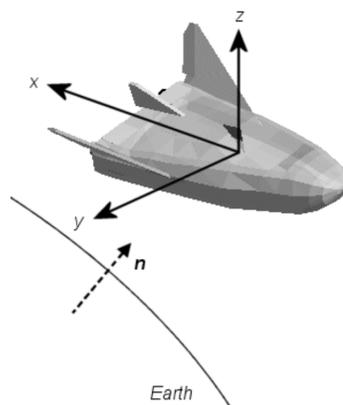


Figura 6: Coordenadas fijas a una nave espacial (MathWorks, 2014).

Para el hardware del sensor de la actitud, la orientación del campo de visión es importante, no la localización del hardware dentro de la nave (Wertz, 2012).

2.2.2.2. Coordenadas inerciales

El sistema de coordenadas inerciales más común es la coordenada celeste relativa al eje de rotación de la Tierra, como se muestra en la Figura 7. Para el caso en que la nave es el centro de la esfera, los ejes del sistema de coordenadas celestes se obtienen uniendo los polos norte y sur celestes, por lo que se encuentra paralelo al eje de rotación de la Tierra. El polo norte celeste está aproximadamente a 1 grado de la estrella polaris, mejor conocida como Estrella Polar. Para definir completamente al sistema de coordenadas, se tiene que definir al meridiano de referencia o punto de referencia. El punto sobre el ecuador celeste que se escoge como la referencia es el punto donde la eclíptica, o el plano de la órbita de la Tierra alrededor del Sol, cruza el ecuador yendo de sur a norte, conocido como el equinoccio vernal, ésta es la dirección paralela a la línea del centro de la Tierra al Sol en el primer día de la primavera (Wertz, 2012).

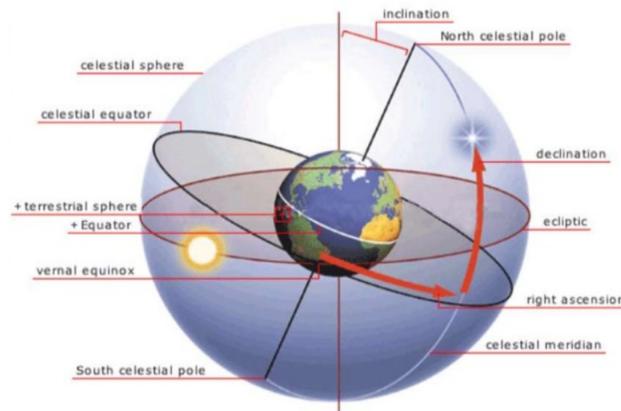


Figura 7: Coordenadas celestes (Patel, 2015).

Desafortunadamente, el sistema de coordenadas celestes no son verdaderamente inerciales ya que no están fijadas a la posición media de las estrellas en la vecindad del Sol. En astronomía, la precesión de los equinoccios (ver Figura 8) es el cambio lento y gradual en la orientación del eje de rotación de la Tierra, que hace que la posición del eje de la Tierra en la esfera celeste se desplace alrededor del polo de la eclíptica, trazando un cono y recorriendo una circunferencia completa cada 25,776 años, esto hace que se deslice el equinoccio vernal a lo largo de la eclíptica relativa a las estrellas fijas a una razón de aproximadamente 50 segundos de arco por año.

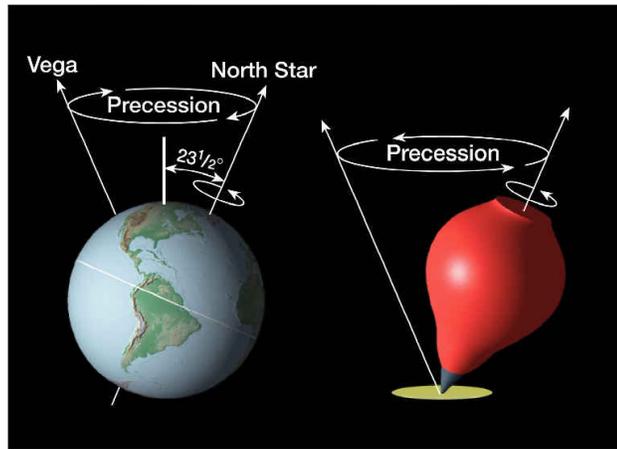


Figura 8: Precesión de los equinoccios (Zurich, 2017).

Cuando se le dio los nombres a la constelación zodiacal hace varios miles de años, el equinoccio vernal estaba en la constelación de Aries, el carnero. El símbolo zodiacal para el carnero ♈ es usado astronómicamente para el equinoccio vernal, también es llamado el primer punto de Aries, desde ese tiempo el equinoccio se ha movido a la constelación de Piscis y actualmente está entrando a Acuario. La otra intersección de la eclíptica y el ecuador es llamado el equinoccio de otoño y está representado por el símbolo zodiacal de Libra ♎ tal como podemos ver en la Figura 9 (Wertz, 2012).

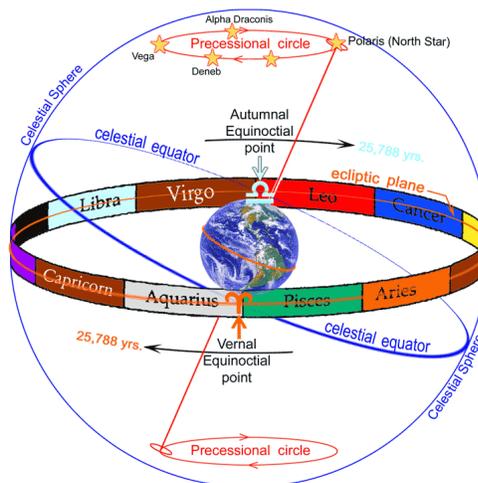


Figura 9: Constelaciones zodiacales y eclíptica (Woebcke, 2016).

La importancia de la precesión de los equinoccios es que ésta genera un cambio lento en las coordenadas celestes de objetos fijos como las estrellas, los cuales deben de tomarse en cuenta para la determinación precisa de la orientación. Esto es, las coordenadas celestes requieren de una fecha de referencia para definir precisamente la posición del equinoccio vernal. El sistema

más usado comúnmente son coordenadas del año 1950, coordenadas del año 2000 y la fecha actual (TOD, acrónimo de la palabra en inglés *True of Date*). Las coordenadas anteriores están definidas al momento del tiempo de la órbita y son comúnmente usadas en el funcionamiento de la nave espacial, donde se hacen pequeñas correcciones para mantener las coordenadas (Wertz, 2012).

Los componentes de elevación o latitud de una coordenada celeste son universalmente conocidos como la declinación δ . Similarmente, el componente del azimut es conocido como la ascensión a la derecha α . La ascensión a la derecha es medida en grados en la nave espacial, en la mayoría de las tablas astronómicas es medida en horas, minutos, segundos, en donde 1 hora = 15° , 1 minuto = $1/60$ hora = $1/4^\circ$ y 1 segundo = $1/60$ minuto = 0.004166° . Cada una de estas mediciones corresponden a la porción de rotación de la Tierra en ese periodo de tiempo. Hay que tener cuidado que minutos y segundos de la ascensión a la derecha no son equivalentes a minutos y segundos de arco en ningún momento. Las coordenadas celestes son las más usadas, también existen otros sistemas de coordenadas inerciales usadas para propósitos especiales (Wertz, 2012) tal como podemos observar en la Tabla 2.

Tabla 2: Sistema de coordenadas inerciales comunes.

Sistema de Coordenadas	Plano ecuatorial	Punto de referencia	Coordenada azimut	Coordenada elevación
Celeste (o Ecuador)	Ecuador Celeste (Paralelo al ecuador)	Equinoccio vernal	Ascensión a la derecha	Declinación
Eclíptico	Eclíptico (Plano de la órbita de la Tierra)	Equinoccio vernal	Longitud celeste	Latitud celeste
Galáctico	Plano adoptado de la galaxia	Dirección adoptado del centro de la galaxia	Longitud galáctico	Latitud galáctica

2.2.2.3. Coordenadas definidas por órbita

Por último, se definirá un sistema de coordenadas que tiene su orientación relativa a la Tierra y con la nave que se mueve en su órbita, estas coordenadas son conocidas como alabeo (roll), cabeceo (pitch), y guiñada (yaw) o RPY. En este sistema, el eje de guiñada está dirigido hacia el nadir (por ejemplo, al centro de la Tierra), el eje de cabeceo está dirigido hacia la normal negativa de la órbita y el eje de alabeo es perpendicular a los otros dos tal que el vector unitario a lo largo de los tres ejes tienen la relación $\hat{R} = \hat{P} \times \hat{Y}$. En una órbita circular, el eje del alabeo estará a lo largo del vector de velocidad. Los ángulos de alabeo, de cabeceo y de guiñada, $(\epsilon_r, \epsilon_p, \epsilon_y)$, para una nave en órbita circular y un observador en la nave orientada en la dirección del movimiento

con la Tierra abajo, una rotación de cabeceo positiva hace que la nariz de la nave se mueva hacia arriba, la guiñada positiva mueve a la nave hacia la derecha, y el alabeo positivo rota a la nave en sentido horario. El sistema RPY es comúnmente usado para naves orientadas a Tierra (Wertz, 2012).

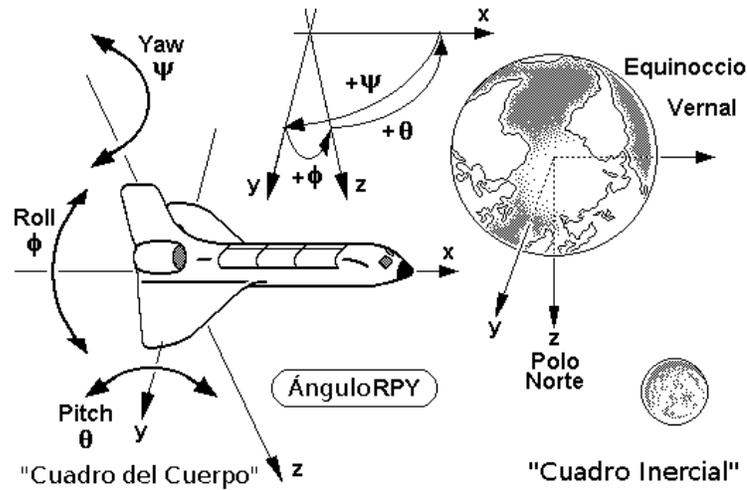


Figura 10: Cuadro inercial (Balancer, 2010).

2.2.3. Sistema de coordenadas ICRS

El sistema de referencia celeste internacional (ICRS acrónimo de *International Celestial Reference System*) es accesible por medio de coordenadas de referencia de las fuentes de radio extragaláctica, tal como el Marco de Referencia Celestial Internacional (ICRF acrónimo de *International Celestial Reference Frame*) (FACG, 2013).

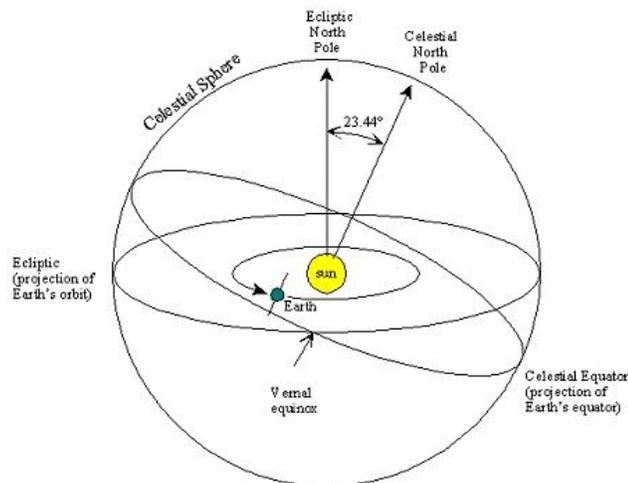


Figura 11: Sistema de coordenadas esféricas celestes centradas al Sol (ESRL, 2017).

El ICRS cumple con las condiciones especificadas por las recomendaciones de la Unión Astronómica Internacional (IAU acrónimo de *International Astronomical Union*) de 1991. Su origen está localizado en el baricentro del sistema solar mediante el modelado apropiado de las observaciones por interferometría de base ancha (VLBI acrónimo de *Very Long Baseline Radio Interferometry*) en el marco de referencia de la Relatividad General. Su polo está en la dirección definida por los modelos IAU convencional para la precesión (Lieske *et al.*, 1977) y la nutación (Seidelmann, 1982). El origen de la ascensión a la derecha fue implícitamente definido fijando la ascensión a la derecha del quásar 3C 273B (Hazard *et al.*, 1971) al valor convencional de FK5 transferido a J2000.0 (12 h 29 m 6.6997 s) (Kaplan *et al.*, 1982).

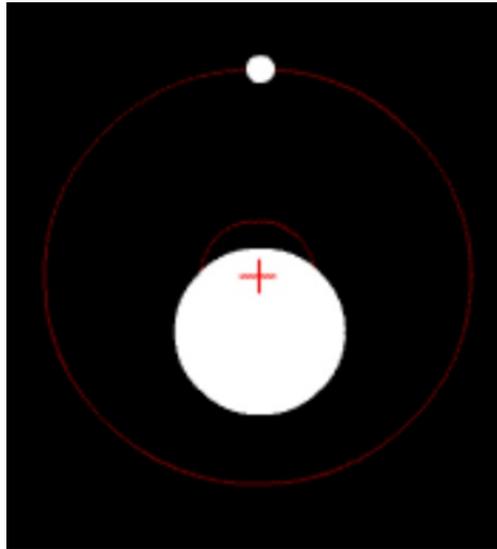


Figura 12: Baricentro de un sistema de rotación de dos cuerpos en el espacio (Cooperk, 2014).

Las posiciones de las estrellas Hipparcos, sus movimientos y las efemérides del sistema solar JPL (acrónimo de “Jet Propulsion Laboratory”) están expresados en el ICRS.

El ICRS puede ser conectado al Sistema de Referencia Terrestre Internacional (ITRS acrónimo de International Terrestrial Reference System) usando los Parámetros de Orientación de la Tierra IERS (EOP acrónimo de Earth Orientation Parameters).

2.3. Paralaje

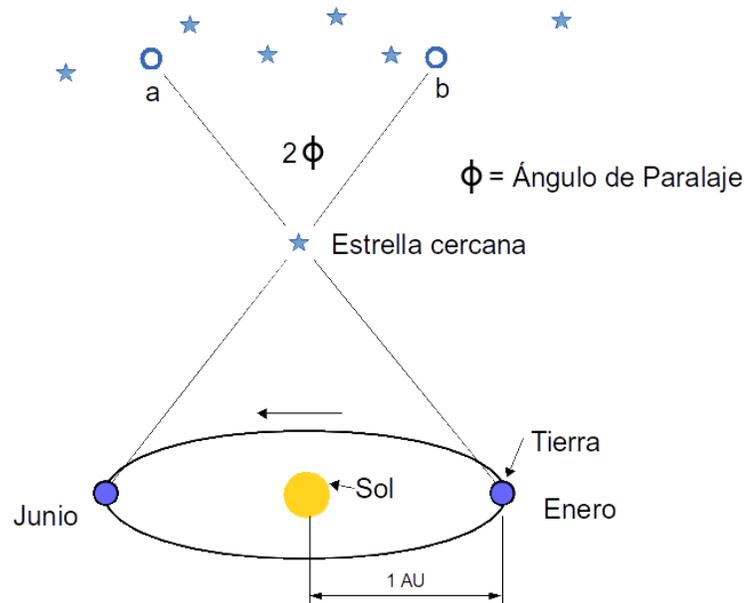


Figura 13: Paralaje estelar.

Paralaje es el desplazamiento de la dirección de un objeto cercano cuando el observador se mueve, tal como se ilustra en la Figura 13, (la cantidad de paralaje para la órbita normal está exagerada en esta figura). No hay paralaje debido al desplazamiento del centro del sistema de coordenadas de un lugar a otro, debido a que los ejes se mueven paralelamente a estos mismos y se definen para mantener las direcciones fijas en el espacio más que apuntar a un objeto real, con la excepción de las coordenadas del alabeo, del cabeceo y de la guiñada. Tal como el eje de polo a polo de las coordenadas celestes centradas a la nave es paralelo al eje de la Tierra (Wertz, 2012).

ϕ es el ángulo del paralaje de una estrella.

$$\phi = \frac{\text{Distancia de la Tierra al Sol}}{\text{Distancia a la estrella desde el Sol}} \quad (5)$$

En principio, existe un pequeño paralaje debido a los sensores de actitud que no están instalados precisamente en el centro geométrico de la nave espacial. En la práctica este desplazamiento se puede despreciar. Para el hardware del sensor de la actitud, la orientación del campo de visión es importante, pero la localización del hardware dentro de la nave no lo es. Dos tipos de paralajes pueden ser importantes en algunas circunstancias. El paralaje solar es el desplazamiento en la dirección del Sol cuando un satélite terrestre se mueve en su órbita. La cantidad de desplazamiento

debido al movimiento del satélite al diámetro completo de su órbita perpendicular a la dirección del Sol es de 0.005° en la órbita baja de la Tierra, 0.032° a la altitud síncrona, y de 0.29° a la distancia de la Luna (Wertz, 2012).

El paralaje estelar es el desplazamiento de la dirección de las estrellas cercanas (otras distintas al Sol) debido al movimiento orbital de la Tierra alrededor del Sol. Nota: el paralaje estelar está definido como el desplazamiento de la dirección de una estrella debido al movimiento del observador (perpendicular a la dirección de la estrella) de 1 AU (unidad astronómica), o el radio de la órbita de la Tierra. De ahí, el máximo desplazamiento en la dirección estelar para un satélite terrestre para 1 año será dos veces el paralaje estelar. El paralaje estelar más grande es de $0.76'' = 2 \times 10^{-4}$ grados para la estrella vecina más cercana al Sol, Alfa Centauro. Por lo que los paralajes estelares son solo de interés para el trabajo de sensores de estrellas muy precisos (Wertz, 2012).

2.4. Magnitudes visual y aparente

Históricamente, la magnitud de una estrella fue calculada basada en la brillantez tal como es percibida por el ojo humano y de ahí llamada **magnitud visual**. Dado que el ojo es más sensitivo en la longitud de onda verde ($\approx 550\text{nm}$), la **magnitud visual** es la intensidad de la estrella aproximada a esta longitud de onda. Desafortunadamente, la luminancia de las estrellas a esta longitud de onda no es monocromática y contiene componentes de varias longitudes de onda desde el ultravioleta profundo al infrarrojo lejano (Katake, 2006).

Con el objetivo de estandarizar la estimación de la magnitud, se utiliza un sistema fotométrico llamado sistema UBV. En el sistema UBV la magnitud es medida en tres diferentes longitudes de onda: U (centrado cerca de 350 nm en el ultravioleta), B (435 nm en el azul) y V (555 nm en el verde visible). La banda V fue escogida ya que se aproxima a la respuesta del ojo (Katake, 2006).

La **magnitud aparente** es una combinación de la magnitudes medidas a la longitud de las ondas UBV y oficialmente está definida como la brillantez de la estrella observada por arriba de la capa superior de la atmósfera de la Tierra. En ausencia de cualquier información relacionada a la longitud de onda medida, la **magnitud aparente** se supone que será equivalente a la **magnitud visual** (Katake, 2006).

2.4.1. Magnitud estelar

La clasificación de las estrellas visibles a ojo más antigua fueron hechas por Hiparco de Nicea en el año 120 A.C. Él dividió las estrellas basadas en su brillantez en 6 clases o magnitudes.

En la primera clase se encuentra la estrella más brillante sin incluir al sol y la sexta clase tiene la estrella visible más tenue. Las seis clases formaron una escala uniforme en donde cada diferencia de la magnitud representa una medida igual en la percepción visual (Katake, 2006).

Esta clasificación fue modificada por Ptolomeo y le asignó valores enteros a las seis magnitudes. Las clasificaciones que hizo Ptolomeo fueron descubiertas a mitad del siglo *XIX*. Sin embargo, ya que la respuesta del ojo humano a la luminancia no es lineal, esto se debe a que el ojo humano no mide los niveles de la luminancia absoluta de las estrellas, sino que percibe las diferencias relativas. Basado en esta observación, Norman Pogson definió una escala logarítmica en 1856 (Katake, 2006):

$$\frac{E_1}{E_2} = 2.512^{(m_1 - m_2)}, \quad (6)$$

en donde E_1 , E_2 representan la luminancia grabada y m_1 , m_2 representan las magnitudes correspondientes. El número 2.512 es la raíz quinta de 100 y corresponde a un cambio de 5 magnitudes ($m_1 = 1$ y $m_2 = 6$) es decir que la estrella de magnitud 6 es 100 veces más tenue que la estrella de magnitud 1. La ecuación de Pogson se presenta a continuación (Katake, 2006):

$$m = -2.5 \log(E) + C. \quad (7)$$

El coeficiente -2.5 es llamado la escala de Pogson y el término C es una constante escogida arbitrariamente llamado el punto cero. Por lo que la magnitud es la unidad estándar para medir la brillantez aparente de un objeto astronómico (Katake, 2006).

Si m_1 y m_2 son las magnitudes de las estrellas con flujo luminoso I_1 e I_2 medidas por fotometría (Bessell, 2005) entonces:

$$m_1 - m_2 = -2.5 \log(I_1/I_2). \quad (8)$$

Alternativamente

$$\frac{I_1}{I_2} = 10^{-0.4(m_1 - m_2)}. \quad (9)$$

Hay que notar que 1 mag corresponde a una relación de intensidad de 2.5 y para 5 mag corresponde a una relación de intensidad de 100. Cuando el valor de la **magnitud aparente** de

una estrella es más bajo, su intensidad es más brillante comparado al de una estrella con el valor de su **magnitud aparente** más grande (Katake, 2006). Las estrellas visibles menos brillantes a simple ojo son de una **magnitud aparente** de 6, mientras la estrella Sirius que es la más brillante tiene una **magnitud aparente** de -1.46.

Para estandarizar la escala de magnitud, los astrónomos escogieron la estrella Vega o *Alpha Lirae* para representar la magnitud cero en todas las longitudes de onda. Por muchos años, la estrella Vega sirvió como base para calibrar la escala de brillantez fotométrica UBV, sin embargo éste no es el caso actualmente, ya que la **magnitud aparente** del punto cero está definida actualmente en términos de una intensidad especificada numéricamente. Esta aproximación es más conveniente para los astrónomos, ya que la estrella Vega no siempre está disponible para hacer la calibración (Bohlin y Gilliland, 2004; Cochran, 1981).

Teniendo la base teórica presentada en este capítulo se necesita entender cada una de las partes que compone a un sensor de estrellas la cuales se presentarán en el siguiente capítulo.

Capítulo 3. Componentes de un sensor de estrellas real

En este capítulo se hace un estudio básico de cada uno de las partes que componen a un sensor de estrellas, esto es importante para poder entender la división de las áreas de investigación dentro de un sensor de estrellas.

Un sensor de estrellas está compuesto de varias partes, una de ellas es la lente la cual se escoge para que se tenga el ángulo del campo de visión necesario para la aplicación. Esta lente puede captar reflejos de los rayos solares indirectos, por lo que es necesario utilizar un deflector para bloquear los reflejos que provocan los rayos solares indirectos como se puede observar en la Figura 14. La cámara tiene un sensor CCD o CMOS el cual tiene una resolución de pixeles en X así como en Y en el cual se proyecta el campo de visión de la lente, para nuestro caso se propuso un sensor de 2000×2000 pixeles, esto se hace para distribuir a las estrellas en más pixeles del sensor y obtener una mejor estimación del centroide de la estrella. Además, con estas resoluciones se puede resolver mejor dos estrellas que se encuentran muy próximas entre sí en comparación a los sensores de baja resolución.

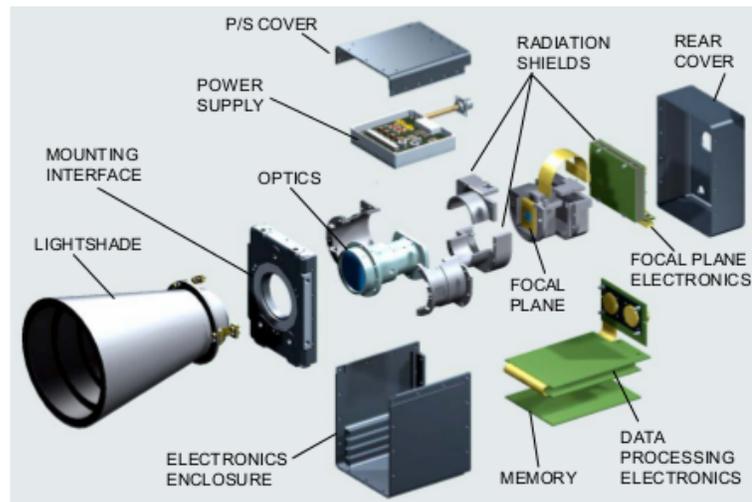


Figura 14: Sistema de determinación de la actitud utilizando una cámara con CCD para la captura de imagen (Xiaojuan y Xinlong, 2011).

Esta cámara se conecta a una microcomputadora el cual va a realizar el procesamiento de la imagen, con el que se obtiene el posicionamiento preciso de todas las estrellas dentro de la imagen, también se tiene un algoritmo de reconocimiento de patrones y se hacen comparaciones con las estrellas del catálogo con la finalidad de identificar las estrellas observadas dentro del campo de visión de la cámara. Por último, con la información que se obtiene anteriormente podemos calcular la actitud del nanosatélite. Este sistema se puede ver en la Figura 15.

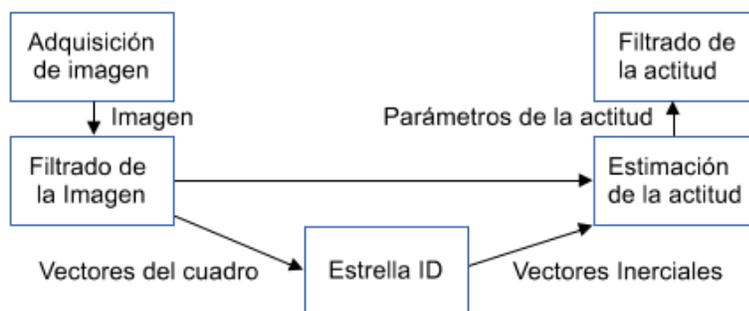


Figura 15: Diagrama a bloques de un Sistema de detección de la actitud por medio de la captura de imágenes con un sensor de visión y con algoritmos de identificación de estrellas típico (Spratling y Mortari, 2009).

Las funciones principales que se procesan en la microcomputadora del sensor de estrellas son: el preprocesamiento de la imagen, la obtención de los centroides, la identificación de estrellas y la determinación de la actitud, ver Figura 16. De todos estos procesos, la identificación de las estrellas es el proceso más complicado y de mayor consumo en tiempo. (Yoon *et al.*, 2013).

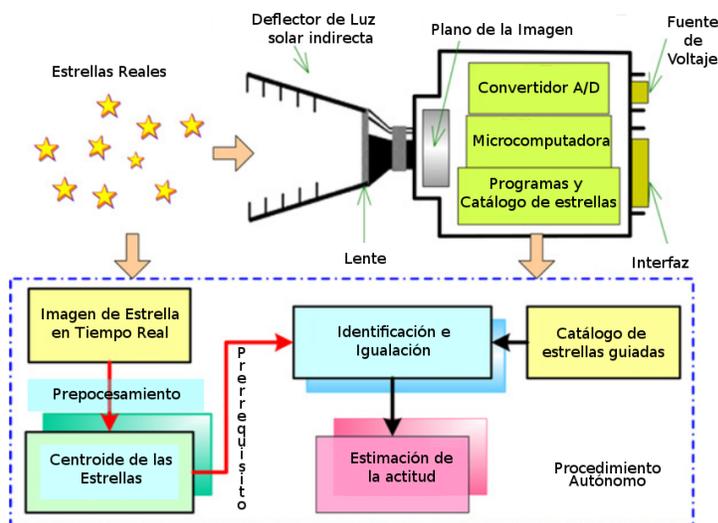


Figura 16: Diagrama esquemático del sistema de determinación de la actitud utilizando una cámara con CCD para la captura de imagen (Xiaojuan y Xinlong, 2011).

Factores tales como el mal alineamiento de la lente, aberración, edad del instrumento (Sun *et al.*, 2013), las no linealidades, respuestas no estándar de cada pixel pueden contribuir al error de medición y causar una desviación en los resultados de los cálculos de los sensores de estrellas ideales (Wang *et al.*, 2012). El mal alineamiento, las no linealidades, las respuestas no estándar de cada pixel y la aberración son independientes del tiempo, o de errores estáticos, los cuales deben de ser calibrados antes del lanzamiento, los efectos de la temperatura y el envejecimiento del instrumento son variantes en el tiempo, o errores dinámicos, los cuales deben ser calibrados

en órbita en tiempo real (Kawano *et al.*, 2008).

Para tener un sensor de estrellas útil y robusto se deben de asegurar los siguientes requerimientos (Samaan y Theil, 2012):

- a) Medición de la actitud para el caso de perdidos en el espacio, es decir que no hay información conocida a priori.
- b) Error de la actitud menor a 0.1° en los tres ejes.
- c) Habilidad para determinar la actitud con velocidades angulares de hasta 2 grados/seg.
- d) Tiempo de procesamiento desde captura de imagen hasta obtención de la determinación de la actitud menor a 100 ms.
- e) Tiempo de exposición menor que 50 ms para velocidades angulares altas de hasta 4 grados/seg.

3.1. Lentes de cámaras

Una lente de cámara es una lente óptica o ensamble de lentes usadas en conjunción con el cuerpo de la cámara y los mecanismos para capturar imágenes de cualquier objeto.

Una simple lente convexa es suficiente pero en la práctica una lente compuesta hecha con un número de elementos de lentes ópticos es requerida para corregir la aberraciones ópticas que surgen. Algunas aberraciones están presentes en cualquier sistema de lentes. El trabajo del diseñador de lentes es balancear estas aberraciones y producir un diseño que es idóneo para el uso de la aplicación y posiblemente para su producción en masa.

- La longitud focal de una lente está definida como la distancia entre el centro óptico de la lente y el sensor de la imagen de la cámara cuando es enfocado al infinito. Para entender esta definición de la longitud focal se necesita definir al centro óptico. El centro óptico de la lente es el punto de la lente en el cual los rayos de luz de dos diferentes fuentes entran a la lente y se cruzan (Guide, 2015).
- La apertura de una lente es el segundo parámetro usado en las especificaciones de ésta y describe que tanta luz deja pasar. La aperturas pueden ser expresadas de diferentes maneras, con F4, f/4, 1:4, los cuales significan lo mismo (Westlake, 2012).

- La apertura *clear* está definida como el diámetro o tamaño del componente óptico que satisface las especificaciones del diseño. Fuera de éste el fabricante no garantiza que la óptica funcione a las especificaciones establecidas (Inc, 2016a).
- El ángulo del campo de visión describe al valor del ángulo de una escena que es tomada por una cámara. Es importante distinguir el ángulo del campo de visión del ángulo de cobertura, el cual describe el intervalo del ángulo que una lente puede tomar.
- Cuando un diseñador óptico trata de comparar el desempeño del sistema óptico, una medida comúnmente usada es la función de transferencia de modulación (*modulation transfer function* MTF) (Inc, 2016b).
- La resolución es la habilidad del sistema de imagen para distinguir detalles de un objeto. Éste frecuentemente es expresado en términos de los pares de líneas por milímetro (en donde un par de líneas es una secuencia de una línea negra y una línea blanca). Esta medida de pares de líneas por milímetros (lp/mm) es también conocido como frecuencia (Inc, 2016b).

3.1.1. Calculando el ángulo de visión de una cámara

Para lentes que proyectan imágenes de objetos distantes, la longitud focal efectiva y la dimensión del formato de la imagen definen al ángulo de visión.

Para las lentes que proyectan a una imagen rectilínea (enfocado al infinito), el ángulo de visión (α) puede ser calculado de la dimensión escogida (d), y de la longitud focal efectiva (f) tal como sigue:

$$\alpha = 2 \arctan \frac{d}{2f}, \quad (10)$$

en donde d representa al tamaño del sensor en la dirección medida.

Se presenta la información del campo de visión de una lente con respecto a la precisión obtenida (McBryde y Lightsey, 2012), tal como se puede observar en la Tabla 3.

Tabla 3: Precisión versus longitud focal y campo de visión para un sensor con pixel de $4.65 \mu\text{m}$ (McBryde y Lightsey, 2012).

ϕ teórico (arco-segundos)	Longitud focal (mm)	Campo de visión (grados)
19.98	4.8	46.62
11.99	8	32.42
7.99	12	22.94
4.17	23	12.45
2.74	35	8.23

Las precisiones de 2 sensores de estrellas comerciales Goodrich (Goodrich, 2004) versus el campo de visión se pueden observar en la Tabla 4.

Tabla 4: Desempeño de 2 sensores de estrellas comerciales marca Goodrich (Goodrich, 2004).

Desempeño	Campo de visión estrecha	Campo de visión ancha
Campo de visión	8° × 8°	20° × 20° circular
Sensitividad de la magnitud	+6.5	+5.1
Potencia necesaria (promedio a +45)°C	10W	10W
Peso (con reflector)	8.5 lb	7.5 lb
Frecuencia de muestreo	6 Hz	2 Hz
Estrellas capturadas simultáneamente	6	6
Precisión total		
cabeceo (pitch) y guiñada (yaw)	2 arcseg	5 arcseg
alabeo (roll)	40 arcseg	40 arcseg

3.1.2. Campo de visión

Los sensores de estrellas adquieren imágenes prácticamente de cualquier área de la esfera celeste no importando la orientación. Para un sensor de estrellas la selección del campo de visión (FOV del inglés *Field of View*) juega un papel fundamental. Por lo que para un área dada de la esfera celeste presente no solo depende de la óptica usada durante la adquisición sino también de la sensibilidad, resolución y orientación del sensor de imagen.

Muchos de los sensores de imágenes tal como los CMOS (*Complementary Metal Oxide Semiconductor*) o CCD (*Charge Coupled Device*) tienen un FOV rectangular por lo que las estrellas visibles en la imagen capturada depende del ángulo de rotación de este sensor. Si el sensor de imagen se rota, las estrellas visibles serán diferente al de una imagen en donde el sensor no está rotado. Este comportamiento se muestra en la Figura 17, las estrellas que el sensor captura para el FOV rectangular no rotado con líneas de color azul de la Figura 17-a son 1, 2, 3, 5, 6, 7, 8, 9 y 15. Si el sensor se rota a 45° (FOV con líneas de color rojo de la Figura 17-b), las estrellas que éste captura son 1, 2, 3 y 4. Si el sensor se rota a 60° (FOV con líneas de color magenta de la Figura 17-c), las estrellas que éste captura son 1, 2, 3, 4, 5, 6, 7 y 14, por lo que para este caso no es buena idea utilizar un FOV rectangular para usarlo dentro de un sensor de estrellas. Al contrario, si el sistema de identificación de estrellas utiliza un FOV circular el cual el diámetro es igual a la dimensión del lado con la longitud menor del sensor de la imagen rectangular (FOV con líneas de color verde de la Figura 17-d), las estrellas visibles en la imagen capturada serán las mismas en cualquier rotación del sensor, para este caso las estrellas que el sensor captura son 1,

2 y 3, el cual es el comportamiento que se necesita para usarlo en un sensor de estrellas real.

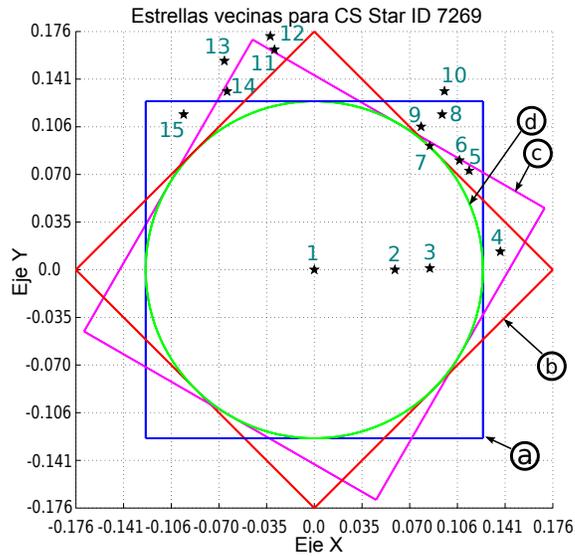


Figura 17: Comparando el FOV rectangular *versus* uno circular. a) FOV rectangular no rotado, b) FOV rectangular rotado 45° , c) FOV rectangular rotado 60° , d) FOV circular.

Para obtener todas las estrellas que se encuentran dentro del campo de visión de un área de la esfera celeste se siguen los siguientes pasos:

1. Convertir las coordenadas esféricas para cada una de las estrellas dentro del catálogo a un vector unitario en las coordenadas cartesianas usando la siguiente expresión (Rad *et al.*, 2014):

$$\vec{r} = \begin{bmatrix} \cos(\delta) \cos(\alpha) \\ \cos(\delta) \sin(\alpha) \\ \sin(\delta) \end{bmatrix}$$

2. Después calcular el ángulo $\psi = \vec{r}_1 \cdot \vec{r}_2$ entre los vectores de las estrellas \vec{r}_1, \vec{r}_2 , siendo \vec{r}_1 el vector de la estrella de referencia en estudio.
3. Si el ángulo ψ es menor o igual que el ángulo del FOV/2, entonces la estrella con el vector \vec{r}_2 se encuentra dentro del campo de visión, después se repiten todos los pasos desde el inicio hasta que todas las estrellas dentro del catálogo han sido procesadas.

3.2. Sensores de imágenes

La tarea de los sensores de imágenes CCD (Charge Coupled Device) y CMOS APS (*Complementary Metal Oxides Semiconductor, Active Pixel Sensor*) es transformar la luz (fotones) en señales eléctricas (electrones). La información es transmitida por ambos sensores de forma y medio diferente, por lo que el diseño de cada uno de los sensores es diferente (Von Fintel, 2015).

3.2.1. Sensor CCD

En los sensores CCD las cargas de los pixeles sensitivos a la luz se obtiene por recorrimiento de éstas y convertidas en señales. Las cargas de los pixeles, las cuales son creadas al exponer el semiconductor a la luz, son transportadas a un convertidor A/D central con la ayuda de muchos registros de recorrimiento horizontal y vertical, parecido a una cadena de baldes de agua (Von Fintel, 2015).

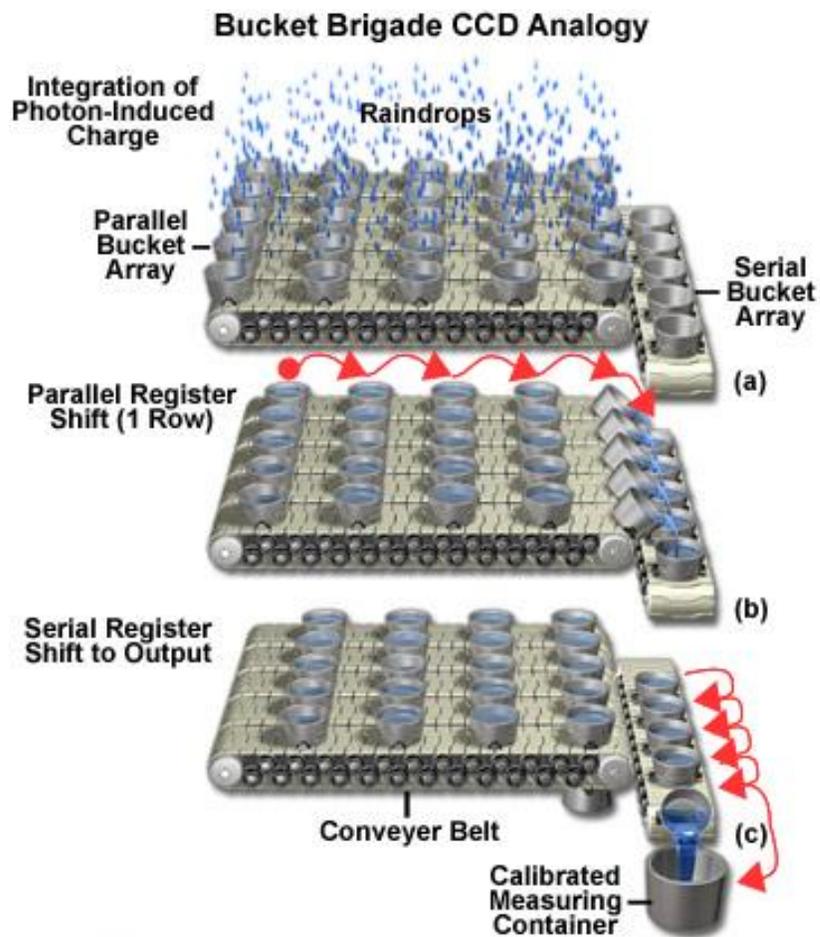


Figura 18: cadena de baldes de agua (Davidson, 2015).

La transferencia de las cargas es forzada con la ayuda de campos eléctricos, los cuales son generados por electrodos en los sensores.

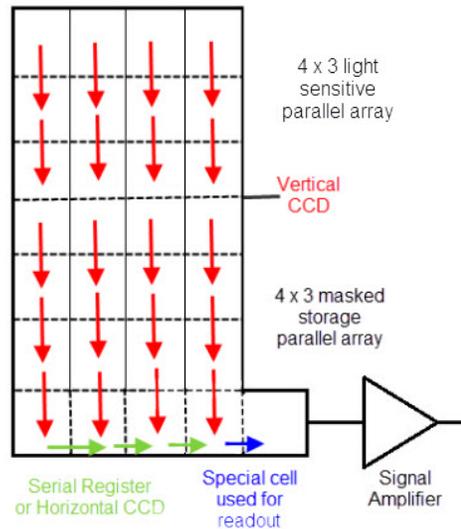


Figura 19: Transferencia de las cargas del CCD (Agarwal, 2015).

- Una característica importante del CCD es su corriente de fuga también conocida como la corriente oscura. Todos los semiconductores sufren de corriente de fuga que resultan de los electrones que tienen suficiente energía térmica para liberarse de la estructura cristalina (Waltham, 2013).
- Un sensor con multiderivación, la superficie del sensor es dividida en áreas con múltiples derivaciones, esto para reducir el tiempo de la transferencia de la carga en el sensor CCD de alta resolución (Von Fintel, 2015).
- La capacidad es una medida de la carga de la señal máxima que un pixel puede almacenar y transferir, y es una función del área del almacenamiento del electrón aislado el cual es determinado por el tamaño del pixel y el número de fases de electrodos usados para definir al pixel (Waltham, 2013).
- Dos de las más importantes métricas de los CCD de grado de ciencia son su capacidad y el ruido de lectura. Juntos determinan al intervalo dinámico del CCD (Waltham, 2013)

3.2.2. sensor CMOS

En los sensores CMOS, se usa un capacitor como almacenamiento de carga el cual es puesto en paralelo a cada pixel individualmente. Este capacitor es cargado por su corriente fotoeléctrica

cuando se expone a cada pixel. El voltaje creado en el capacitor es proporcional a la brillantez y al tiempo de exposición. Diferente al CCD, los electrones capturados en los capacitores por la exposición del sensor a la luz no trabajan con el recorrimiento a un amplificador de salida simple pero son transformados a un voltaje de medición directamente a las fuentes por medio de un circuito electrónico asociado a cada pixel. Este voltaje puede ser obtenido por el procesador de señal análogo (Von Fintel, 2015).

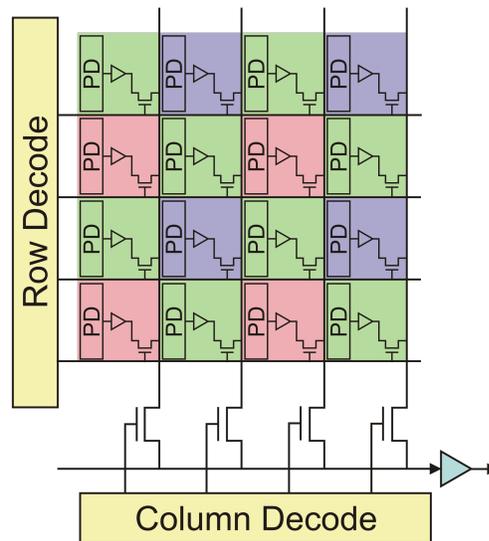


Figura 20: Obtención de la señal de un sensor CMOS.

- Debido al diseño interno del sensor CMOS se pueden obtener altas velocidades y altas resoluciones sin la necesidad de utilizar una arquitectura multiderivación (Von Fintel, 2015).
- Grandes superficies tanto en el sensor como en los píxeles individuales ofrecen más espacio para capturar luz. Mientras más área de la superficies es disponible, es mejor la señal a ruido (SNR). Mientras mayor es la SNR, mayor es la calidad de la imagen (Von Fintel, 2015).
- Una limitación es el movimiento del voltaje lineal que puede ser obtenido fuera de las regiones del umbral del transistor en los procesos CMOS de bajo voltaje. El intervalo dinámico lineal de los sensores CMOS es ≈ 5000 , considerablemente menor que el de un CCD (Waltham, 2013).
- El sensor CCD sigue siendo mejor indiscutiblemente en el intervalo dinámico y la precisión fotométrica. Los efectos de la radiación en los sensores CMOS y los sensores CCD son parecidos y ambos sufren de la radiación ionizante y los daños por desplazamiento cuando son usados en el espacio. Sin embargo, la ventaja clave de la APS CMOS es que no hay una degradación de la eficiencia de transferencia de carga (CTE) (Waltham, 2013).

- Un efecto no visto en los sensores CCD es la susceptibilidad a los eventos únicos de *latchup* (SEL)(Waltham, 2013).
- Con los sensores CMOS el obturador *rolling* puede ser manejado por medio de una combinación de velocidad rápida del obturador y la temporización de la fuente de luz. Existen algunos sensores CMOS que implementan al obturador global aunque su formato y sus características de desempeño del video todavía no son óptimas para muchos requerimientos de ciencias de la vida.

3.2.3. Diferencias de los sensores CMOS versus sensores CCD

Solamente los sensores CMOS de alta resolución tienen obturador global, los de baja resolución utilizan el obturador *rolling* (Von Fintel, 2015).

Tabla 5: Sensores CCD versus sensores CMOS.

	Sensores CCD	Sensores CMOS Nuevos
Obturador	Global	Global o <i>rolling</i>
Costo de cámara /sensor para la misma resolución	Muy alta	Medio (obturador global) A muy bajo (obturador <i>rolling</i>)
Máxima velocidad de lectura	No más de 20 fps	Muy alta, virtualmente ilimitado (e.g. sensor de 4 MP obturador global a 180 fps)
Consumo eléctrico	Alto	Bajo
Selección de Lente	Limitado, debido al formato del sensor la mayoría > 2/3"	Muy grande (1/4" – 1")
Generación de calor y tiende a ser ruidoso si éste no ha sido enfriado	Muy alta	Baja
Calidad de imagen: Intervalo dinámico	Muy bueno	Bajo a muy bueno
Calidad de la imagen Sensitividad	Bueno	Bajo a muy bueno
Calidad de la imagen Bajo ruido	Poco	Poco
Interferencia de la imagen por configuración de derivación	Puede ocurrir, es debido a una calibración laboriosa (Solo valida para sensores CCD con multiderivación.)	No
Efecto <i>blooming</i> en la imagen	Si	No
Efecto <i>smearing</i> en la imagen	Si	No
Evento Latchup	No	Si

3.3. Procesamiento de imagen

Los cálculos importantes del procesamiento de imágenes para un sensor de estrellas son: reducción de ruido de la imagen, segmentación (Wen-Cheng y Xiao-Jun, 2013), agrupamiento y obtención de los centroides (Arbabmir *et al.*, 2014).

Para una imagen proyectada a su plano focal, los puntos de una estrella que ocupa un solo pixel limita la precisión con la cual su posición puede ser estimado en el pixel, se puede desenfocar a la óptica deliberadamente (Katake, 2006), esto para dispersar a las estrellas en más de un pixel.

Los centroides de las estrellas desenfocadas pueden ser estimados mejor que un 1/10 de un pixel, efectivamente mejorando la precisión por un orden de magnitud (Katake, 2006).

3.3.1. Reducción de ruido de la imagen

Remover el ruido de las imágenes con muchas estrellas es un gran reto debido a que es difícil reconocer los pixeles ruidosos. Para tratar de disminuir el problema se puede emplear algún tipo de filtro que son usados comúnmente tal como el filtro promedio aritmético, el filtro Gaussiano, el filtro de Wiener, los filtros de difusión anisotrópicas (DA), etc (Arbabmir *et al.*, 2014).

La influencia del ruido en el proceso de estimación puede ser dividida en dos tipos, el ruido aleatorio y el ruido sistemático. El ruido aleatorio incluye al ruido corto, ruido de corriente oscura, ruido de lectura del sensor de imagen, y ruido de radiación, los cuales están relacionados al hardware del sensor. El ruido de los sensores de imagen tal como el de un CCD y los de CMOS es otro factor principal que limita la precisión de la estimación de los centroides (Jia *et al.*, 2010).

El ruido sistemático es debido a la naturaleza del algoritmo del cálculo del centroide. Este ruido puede causar pérdida de precisión de arco-segundos por lo que es necesario analizar el error sistemático y diseñar un método de compensación para mejorar la precisión de la estimación de la posición de los centroides de las estrellas en la imagen.

Dispersión de la intensidad de las estrellas por filtro promedio

Bajo condiciones estáticas, la distribución de los puntos de las estrellas son generalmente representadas como una función Gaussiana de dos dimensiones utilizando un algoritmo de dispersión de la imagen utilizando un círculo de 3×3 o 5×5 con la finalidad de que la precisión del centroide de la estrella pueda ser mantenida dentro del nivel de subpixel (Zhang *et al.*, 2012b). El

proceso de filtrado promedio aritmético calcula el valor promedio de la imagen original $g(x, y)$ en el área definida por S_{xy} . El valor de la imagen restaurada f en cualquier punto (x, y) es simplemente el cálculo del promedio aritmético usando los píxeles en la región definida por S_{xy} (Gupta, 2011).

Reducción de ruido usando la convolución Gaussiana

Desenfocar una imagen es la manera más simple de suavizar una imagen; cada valor del pixel de la imagen es una suma ponderada de sus vecinas en la imagen original (Paris *et al.*, 2009). El comportamiento del filtro Gaussiano es el promedio ponderado de la intensidad de los píxeles adyacentes con una ponderación que decrementa dependiendo de la distancia espacial a la posición central \mathbf{p} . La ponderación para el pixel \mathbf{q} está definida por la Gaussiana $G_{\sigma}(\|\mathbf{p} - \mathbf{q}\|)$, en donde σ es un parámetro que define el tamaño de la vecindad. Los bordes dentro de la imagen son desenfocados con este procedimiento porque los píxeles tienen discontinuidades y todos son promediados (Paris *et al.*, 2009).

Reducción de ruido por difusión anisotrópica

La difusión anisotrópica (DA) es utilizada para reducir el ruido de las imágenes de las estrellas en el cielo (Mohammadi *et al.*, 2012). En el procesamiento de imágenes y máquinas de visión, DA o difusión de Perona–Malik es una técnica de reducción de ruido que preserva las partes más significativas del contenido de la imagen, tal como las líneas, orillas, esquinas y picos definidos (Arbabmir *et al.*, 2014).

Reducción de ruido por filtro bilateral

El filtro bilateral se define como un filtro promedio ponderante de los píxeles cercanos, de manera similar a la convolución Gaussiana. La diferencia es que el filtro bilateral toma en cuenta la diferencia en el valor con los vecinos para preservar los filos mientras se suaviza. La idea clave del filtro bilateral es que para que un pixel inflencie a otro pixel, éste debe no solo ocupar un lugar cercano pero también un valor similar (Paris *et al.*, 2009). La formalización de esta idea se remonta a la literatura de Yaroslavsky (Yaroslavsky y Yaroslavskij, 1985), Aurich y Weule (Aurich y Weule, 1995), Smith y Brady (Smith y Brady, 1997) y de Tomasi y Manduchi (Tomasi y Manduchi, 1998).

Reducción de ruido por filtro bilateral adaptativo

Este tipo de filtro se utiliza para suavizar una imagen con la ventaja de preservar la textura basada en la información la cual puede ser informativa (Anantrasirichai *et al.*, 2014).

El filtro bilateral normal utiliza parámetros fijos que ayudan a preservar textura pero deja una gran cantidad de ruido moteado y si remueve el ruido moteado también remueve la información de la textura. El filtro bilateral adaptativo se acerca más a una función Gaussiana pero ésta es cambiada adaptativamente dependiendo de la información local en el pixel y su vecindad (Anantrasirichai *et al.*, 2014)

3.3.2. Segmentación de la imagen

La técnica de segmentación de imágenes es una tarea importante y de desafío en el campo del análisis de imagen, visión computacional, reconocimiento de patrones, y también es la base para comprender imágenes así como el reconocimiento de objetos.

La segmentación de imagen es el proceso de particionar una imagen en múltiples segmentos, esto con el fin de cambiar la representación de una imagen en algo que sea más significativo y más fácil de analizar (Dass y Devi, 2012), cada uno de los segmentos representa algún tipo de información útil para el usuario en forma de color, intensidad o textura. De ahí la importancia de aislar los límites de cualquier imagen en forma de sus segmentos (Khan, 2013).

En otras palabras la segmentación de la imagen es un proceso de dividir la imagen en regiones distintas, con el objetivo de extraer del fondo al objeto de interés (Abdullah *et al.*, 2012).

Para el caso de un sensor de estrellas la segmentación se refiere en dividir la imagen en la información del primer plano frontal (las estrellas) y el fondo (Xu *et al.*, 2013).

A continuación se presentan algunas técnicas de segmentación de imagen.

Método del nivel de umbral fijo

Un método para separar las estrellas de una imagen del fondo es crear el histograma y poner un límite a la magnitud aparente con la que se desea trabajar (recordar que para nuestro caso es ≤ 6.0). Este método busca en todos los pixeles de la imagen si el nivel de la intensidad del pixel es mayor al umbral seleccionado, si es mayor a éste se almacena su valor de la intensidad y si es menor su valor se pone en cero.

Algoritmo de Otsu

El algoritmo de Otsu utiliza un valor de muestra promedio y además utiliza su desviación para calcular la dispersión (Yang *et al.*, 2012). Los píxeles de la imagen son divididos en dos clases C_0 y C_1 , los cuales son el primer plano frontal y el fondo debido al umbral t . En donde C_0 representa a los píxeles dentro de los niveles $[1, 2, \dots, t]$, en donde t es el nivel de umbral y C_1 denota a los píxeles dentro del nivel $[t + 1, t + 2, \dots, L]$, en donde L es el nivel de gris (Yang *et al.*, 2012).

Modificación del algoritmo de Otsu

Cuando la distribución de las clases C_0 y C_1 (primer plano frontal y el fondo) es asimétrico, el valor estimado de la mediana es más robusto comparado con el nivel de gris promedio. Se puede verificar que reemplazando al promedio por la mediana se obtiene un umbral óptimo t^* que es muy preciso a la presencia de una distribución fuertemente asimétrica de C_k comparada con los umbrales escogidos por el algoritmo de Otsu (Yang *et al.*, 2012).

Método del umbral multinivel

En el método del umbral multinivel se usa más de un valor del umbral. Este método fue desarrollado debido a que el rechazo por umbral fijo no siempre es bueno usarlo para la segmentación global. Arora *et al.* (2008) propuso un método de umbral multinivel que identifica los valores del umbral globalmente basados en la distribución del nivel de gris. Un algoritmo recursivo es aplicado para determinar la secuencia de valores del umbral que se basan en el promedio y la desviación estándar en cada etapa.

Método multiumbral

(Abdullah *et al.*, 2010) El método del rechazo por multiumbral utiliza una serie de valores del umbral y calcula el número total de blobs u objetos en una imagen para cada uno de los umbrales. Los valores del umbral pico son éstos con el número total de objetos más alto comparado a los umbrales vecinos.

Reducción de iluminación no homogénea por filtro Homomórfico

El filtrado Homomórfico está basado en el modelo de reflexión que toma en cuenta la iluminación y sus propiedades reflectivas, considera a los componentes de alta frecuencia y de baja frecuencia de la imagen (Wen-Cheng y Xiao-Jun, 2013).

Se necesitan conocer las características de los componentes de incidencia y de la reflexión en una imagen para diseñar un filtro adecuado para disminuir los componentes de baja frecuencia y aumentar los componentes de alta frecuencia (Wen-Cheng y Xiao-Jun, 2013). Por lo que se puede resolver los problemas de los campos de iluminación no uniformes, comprimiendo el intervalo dinámico y mejorando el contraste. El filtrado Homomórfico es un proceso de filtrado en el dominio de la frecuencia y que puede solucionar el problema de iluminación no uniforme (Delac *et al.*, 2006).

Rechazo por umbral adaptativo basado en la relación de señal a ruido pico

Este algoritmo utiliza un histograma de niveles de gris, un intervalo del umbral y la relación de señal a ruido pico (PSNR). El valor promedio del histograma de niveles de gris está fuertemente relacionado a la intensidad del pixel. Para evitar el efecto de luminosidad en la imagen se utiliza un valor promedio del histograma de niveles de gris para predeterminar el tipo de imagen: oscuro, medio o brillante. El valor medio corresponde al valor de los pixeles dominantes en la imagen (PirahanSiah *et al.*, 2014).

3.4. Agrupamiento de pixeles que representan estrellas

El agrupamiento de pixeles es el proceso de agrupar los datos multidimensionales similares a un número de grupo. Varias implementaciones han sido propuestas para etiquetar los componentes conectados en una imagen. Éstas se pueden dividir en cinco clases principales (He *et al.*, 2009):

1. algoritmo de un escaneo.
2. algoritmo de dos escaneos.
3. algoritmo multiescaneo.
4. algoritmo híbrido.
5. algoritmo tipo rastreo.

En el algoritmo de un escaneo, los pixeles de cada objeto son determinados concurrentemente con el escaneo de la imagen y una etiqueta única es asignada a cada objeto. El algoritmo de un solo escaneo fue propuesto para etiquetar a los componentes conectados. El algoritmo escanea la

imagen para encontrar los píxeles no etiquetados y sus vecindades son buscados recursivamente hasta encontrar todos los píxeles de los objetos (Abubaker *et al.*, 2007).

En el algoritmo de dos escaneos, en el primer escaneo las etiquetas provisionales son asignados a los objetos y en un segundo escaneo las etiquetas equivalentes son unidos.

En el algoritmo multiescaneo, la imagen es escaneado varias veces en las direcciones hacia adelante y hacia atrás para extraer las etiquetas y unir las etiquetas equivalentes.

El algoritmo híbrido escanea la imagen varias veces en la dirección hacia adelante y hacia atrás como el algoritmo multiescaneo. Sin embargo, similar al algoritmo de dos escaneos una tabla es empleada para resolver la equivalencia de etiquetado.

3.4.1. Agrupamiento de un escaneo

Para asignar una etiqueta a un píxel de una estrella, sus píxeles vecinos son analizados. Se unen las etiquetas equivalentes. De ahí, los siguientes tres estados son esperados en la arquitectura propuestas (Azizabadi *et al.*, 2014):

1. No existe etiquetas predefinidas (todas la etiquetas son cero) en la vecindad del píxel de la estrella actual. En ese caso se asigna una nueva etiqueta al píxel y se incrementa un contador de etiquetas, se almacenan las coordenadas del píxel y sus intensidades.
2. Si solamente existe una etiqueta predefinida en la vecindad del píxel actual, se asigna la misma etiqueta al píxel actual y se almacenan la información para el cálculo del centroide.
3. Si existen dos diferentes etiquetas en la vecindad del píxel actual, se asigna la etiqueta más pequeña al píxel actual y se unifican los bloques de las etiquetas.

3.4.2. Agrupamiento de dos escaneos por mínimos y máximos

Después del procedimiento de segmentación se necesita encontrar cuales píxeles tienen intensidad diferente de cero y agrupar los píxeles a las estrellas que le corresponde para después calcular el centroide de cada estrella.

Los píxeles correspondientes a una estrella se pueden observar en la imagen [21](#).

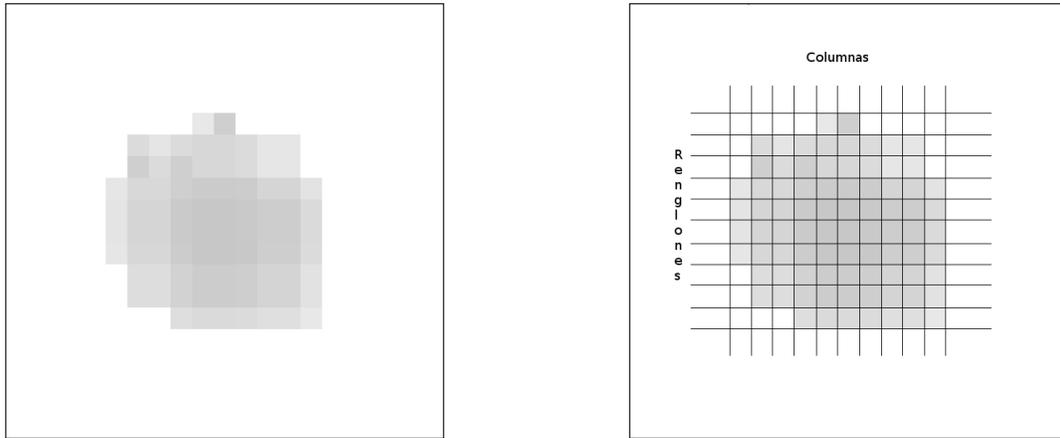


Figura 21: Imagen de una estrella con una ampliación de escala de 3000x. Misma imagen con división por renglones y columnas.

En la primera etapa solo se crean los arreglos de la imagen, en una segunda etapa se conoce a que grupo o estrella le corresponde cada uno de los arreglos, al final de ese procedimiento se pueden calcular los centroides. Cada centroide puede tener varios arreglos, cada arreglo tiene la siguiente estructura:

- (a) Número de columna del pixel de inicio de un centroide.
- (b) Número de columna del pixel final de un centroide.
- (c) Número de renglón.
- (d) Número de grupo del centroide.
- (e) Número de estrella correspondiente (éste no se obtiene en la primera parte del procedimiento de búsqueda de los centroides).

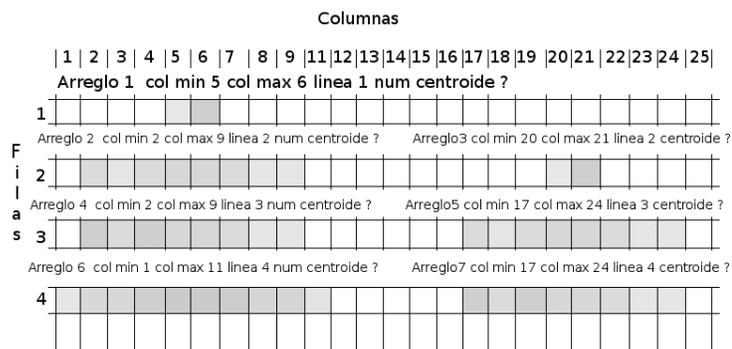


Figura 22: Imagen de dos estrellas con una ampliación de escala de 3000x seccionada por renglones y columnas con creación de arreglos.

Se crean arreglos con los pixeles que no tienen valor de intensidad cero ver figura 22, para que el pixel pertenezca al mismo grupo que se está trabajando se necesita que de un pixel a otro no exista un pixel con intensidad cero si por alguna casualidad hay un pixel con valor de intensidad cero, se crea otro arreglo con otro valor de grupo, y de esa misma manera se busca en todo el renglón, se trabaja de la misma manera para todos los renglones pero creando nuevos arreglos y nuevos números de grupos, cada renglón siempre crea un nuevo arreglo.

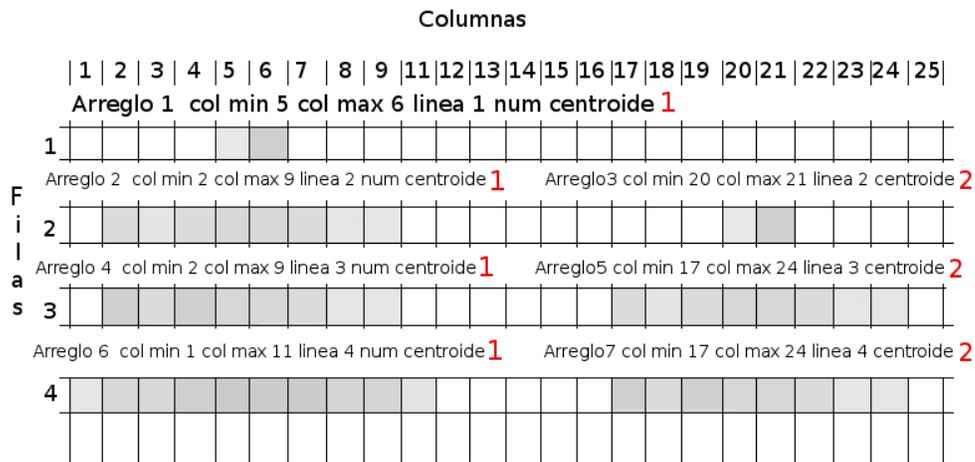


Figura 23: Imagen de dos estrellas con una ampliación de escala de 3000x seccionada por renglones, columnas con creación de arreglos y además con numeración en el campo de número de estrella.

Después del procedimiento anterior, ya solo se busca dentro los arreglos para obtener cada una de las estrellas dentro de la imagen de la siguiente forma:

Si el valor mínimo y máximo de un arreglo de grupo n se compara con otro valor mínimo y máximo de otro arreglo de grupo $n + 1$, siendo la única condición que el grupo n debe de pertenecer a un *renglón* y el otro grupo $n + 1$ sea del *renglón* + 1, ver Figura 23, se dice que el arreglo del grupo $n + 1$ son puntos que pertenecen a una misma estrella del grupo n con alguna de las siguientes condiciones:

(a) $\min_{x_{g2}} \geq \min_{x_{g1}}$ y $\max_{x_{g2}} \leq \max_{x_{g1}}$

(b) $\min_{x_{g1}} \geq \min_{x_{g2}}$ y $\max_{x_{g1}} \leq \max_{x_{g2}}$

(c) $\min_{x_{g2}} \geq \min_{x_{g1}}$ y $\max_{x_{g1}} \leq \max_{x_{g2}}$

Con estas condiciones se encuentran a cada una de las estrellas que existen en la imagen y se escribe la etiqueta del número de la estrella correspondiente a ese arreglo.

3.4.3. Agrupamiento circular

Un juego de pixeles adyacentes en el plano frontal de la imagen son llamados componentes conectados. Para encontrar los componentes conectados, uno tiene que agrupar a los pixeles vecinos en el plano frontal (Arbabmir *et al.*, 2014).

3.5. Cálculo del centroide

El proceso computacional que estima la posición de la estrella en el arreglo del plano focal es conocido como cálculo del centroide. Ésta es una etapa crucial ya que los errores en la estimación del centroide nos lleva a la precisión de apuntamiento total del sensor de estrellas (Katake, 2006).

Existen muchos algoritmos para calcular los centroides tales como los algoritmos de cálculos de centroides por polinomios, algoritmos basados en la regresión de vectores de mínimos cuadrados para compensar los errores sistemáticos (Yang *et al.*, 2011), interpoladores de B-splines (Li *et al.*, 2013), como un arreglo de puntos de masas, etc. La salida resultante del algoritmo del cálculo del centroide es una serie de coordenadas bidimensionales en el plano de la imagen teniendo el origen de la imagen en el centro. Este sistema permite que las coordenadas de las estrellas sean fácilmente convertidas a vectores unitarios en una etapa posterior.

3.5.1. Método del primer momento o centro de masa

El centro de masa o método del primer momento es una de las técnicas comúnmente usadas debido a su simplicidad y robustez.

Se utiliza el cálculo de la posición de la estrella de manera parecida a calcular el centro de masa utilizando técnicas de momento. La ecuación para el cálculo del centro de masa de la imagen que obtiene el centroide de la intensidad de la imagen es (Samaan, 2003):

$$\hat{x} = x_m + \frac{\sum_{i=1}^n \sum_{j=1}^n x_{ij} I_{ij}}{\sum_{i=1}^n \sum_{j=1}^n I_{ij}} \quad (11)$$

$$\hat{y} = y_m + \frac{\sum_{i=1}^n \sum_{j=1}^n y_{ij} I_{ij}}{\sum_{i=1}^n \sum_{j=1}^n I_{ij}} \quad (12)$$

Donde: (x_m, y_m) corresponde a la posición del pixel más brillante de la imagen, I_{ij} = valor de intensidad del pixel (i, j) , (x_{ij}, y_{ij}) = posición del pixel (i, j) .

3.5.2. Método de ajuste Gaussiana simétrico

El cálculo del método de ajuste Gaussiana simétrico es numéricamente más intenso pero es más preciso que el centro de masa. Éste consiste en fijar una función Gaussiana simétrica con un fondo plano para cada una de las coordenadas de las distribuciones marginales. La ecuación básica de la distribución marginal es (Stone, 1989):

$$M(x) = B + H \exp\left[-\frac{(x-x_c)^2}{2R^2}\right] \quad (13)$$

$$M(y) = B + H \exp\left[-\frac{(y-y_c)^2}{2R^2}\right] \quad (14)$$

En donde B es el nivel de fondo definido en la distribución marginal ($B = n_y b$, teniendo que n_y es el arreglo usado para crear la distribución marginal), R es el valor rms de la mitad del ancho de banda de la imagen ($1FWHM = 2.36R$ en donde $FWHM$ es el ancho completo en la mitad del máximo del espectro), y H es la altura de la imagen central en la distribución marginal, el cual puede también ser expresada con la relación (Stone, 1989):

$$H = \frac{N}{(2\pi)^{1/2}R} \quad (15)$$

La ecuación 15 es una aproximación paramétrica y requiere una solución para los cuatro parámetros expresados en las ecuaciones 13 y 14 las cuales pueden ser linealizadas con la expansión de Taylor y resueltas por mínimos cuadrados iterativos. El valor inicial de los parámetros fueron estimados de la distribución marginal y la convergencia fue definida en el punto cuando el cambio máximo del valor de x_c está abajo de ± 0.001 arco-segundos (Stone, 1989).

3.5.3. Método de ajuste Gaussiana circular

Para una Gaussiana circular, se necesitan conocer cuatro parámetros: las posiciones x y y , el ancho de la Gaussiana y la amplitud. Otros parámetros, tal como la brillantez total o integral de la Gaussiana, son una combinación del ancho y la amplitud, frecuentemente son usados en los algoritmos de seguimiento (Anthony y Granick, 2009).

$$I(x, y) = \frac{I_o}{2\pi\sigma_{PSF}^2} \exp\left[-\frac{(x-x_0)^2}{2\sigma_{PSF}^2}\right] \exp\left[-\frac{(y-y_0)^2}{2\sigma_{PSF}^2}\right] \quad (16)$$

Una función Gaussiana circular, sin la influencia del ruido, puede ser transformada en una ecuación lineal usando al algoritmo natural (Nightingale y Gordeyev, 2013):

$$c_1x_i + c_2y_j + c_3 \ln(I_{i,j}) + c_4 = (x_i^2 + y_j^2) \quad (17)$$

en donde c_1 , c_2 , c_3 y c_4 son coeficientes desconocidos definidos como:

$$c_1 = 2x'_c \quad (18)$$

$$c_2 = 2y'_c \quad (19)$$

$$c_3 = -2\sigma_w^2 \quad (20)$$

$$c_4 = -2\sigma_w^2 \ln(A) - x'^2_c - y'^2_c \quad (21)$$

La ecuación lineal puede ser resuelta usando una regresión de mínimos cuadrados lineal.

3.5.4. Método de la Convolución

El centroide por convolución es obtenido primero realizando una convolución discreta de dos dimensiones de una matriz de 15×15 que tiene una forma Gaussiana predeterminada con una matriz de 15×15 de la imagen, el centroide es estimado utilizando el cálculo del primer momento (Nightingale y Gordeyev, 2013).

3.5.5. Método del primer momento ponderado

Éste método utiliza una función ponderada w_{ij} con una forma Gaussiana la cual ajusta la posición del centro iterativamente basado en la iteración previa del cálculo del primer momento ponderado (Nightingale y Gordeyev, 2013).

$$\hat{x} = \frac{\sum_{i=1}^n \sum_{j=1}^n x_{ij} I_{ij} w_{ij}}{\sum_{i=1}^n \sum_{j=1}^n I_{ij} w_{ij}} \quad (22)$$

$$\hat{y} = \frac{\sum_{i=1}^n \sum_{j=1}^n y_{ij} I_{ij} w_{ij}}{\sum_{i=1}^n \sum_{j=1}^n I_{ij} w_{ij}} \quad (23)$$

La localización del centro de la función ponderada para la primera iteración es obtenida basada en el cálculo del primer momento inicial y la dispersión o el ancho Gaussiana, σ_w es estimada basada en la estimación de los puntos $FWHM$, en donde $\sigma_w \approx (FWHM/2.3548)$. El nuevo punto calculado llega a ser el centro de la función ponderada para la próxima iteración. Este proceso es repetido hasta que la posición del punto calculado en las direcciones x y y convergen dentro de 0.001 pixeles de la iteración previa (Nightingale y Gordeyev, 2013).

3.6. Corrección Gamma

La velocidad de cálculo del centroide depende del algoritmo usado, siendo el algoritmo del primer momento el más rápido, el mayor problema con este algoritmo es la menor precisión comparado con otros algoritmos. Existe una técnica para obtener una mayor precisión con los mismos algoritmos usados anteriormente tal como podremos ver en esta subsección.

La función de transformación también llamada corrección Gamma está dada por la expresión (Maini y Aggarwal, 2010):

$s = cI(x, y)^\gamma$, donde c es una constante que normalmente es 1 (Maini y Aggarwal, 2010), para varios valores de γ se obtienen diferentes niveles de intensificación (Nightingale y Gordeyev, 2013). En la transformación de la ley de potencia o corrección Gamma de cuarto orden, los pixeles individuales de la imagen son elevados a la cuarta potencia después de que la imagen ha sido normalizado de 0 a 1. Este método de intensificación de la imagen produce un punto más enfocado mientras reduce el impacto de la imagen ruidosa (Maini y Aggarwal, 2010).

El utilizar la corrección Gamma junto a los cálculos del método del primer momento o centro de masa es mucho más rápido de calcular que cualquiera otro método para obtener mayor precisión en el cálculo del centroide.

3.7. Introducción al catálogo de estrellas

Los catálogos de Hipparcos y Tycho fueron los productos primarios del satélite Hipparcos ¹ de la misión astrométrica de la Agencia Espacial Europea (ESA). Este satélite, el cual operó por 4 años, envió datos científicos de alta calidad de Noviembre de 1989 a Marzo de 1993 (Perryman *et al.*, 1997).

¹La palabra Hipparcos es un acrónimo para *High Precision PARallax COLlecting Satellite*, satélite de recolección de paralaje de alta precisión

Cada catálogo contiene una gran cantidad de datos fotométricos y astrométricos de alta calidad. También contiene datos de variabilidad de estrellas múltiples, dobles y mediciones fotométricas y astrométricas del sistema solar. El catálogo de Hipparcos tiene la información de más de 118,200 estrellas (Perryman *et al.*, 1997).

3.7.1. Catálogo HYG

El Catálogo de estrellas de HYG es un subconjunto de los datos de los 3 mayores catálogos: el catálogo de Hipparcos, el catálogo de las estrellas brillantes de Yale (5^a edición), y el catálogo de Gliese de las estrellas cercanas (3^a edición) (Mohammadnejad *et al.*, 2012).

El catálogo de Hipparcos es la colección de datos con las posiciones estelares de alta precisión, también contienen los paralajes particulares, los cuales son un punto inicial para obtener los datos de las distancias estelares. El Catálogo de las estrellas brillantes Yale contiene datos básicos esenciales de todas las estrellas observadas a simple vista, incluyen mucha información (tal como las letras griegas Bayer tradicionales y los números Flamsteed que sirven para identificar a las estrellas de forma distinta a uno numérico) los cuales no existen en otros catálogos. El catálogo Gliese es el catálogo más completo de las estrellas cercanas (estos se encuentran dentro de los 75 años luz con respecto al Sol). Este catálogo contiene muchas estrellas de muy baja intensidad no encontradas en el catálogo de Hipparcos (Nash, 2014).

Cada uno de los catálogos de estrellas que se pueden obtener tienen la característica que la información de las estrellas están compilados para una época en particular, para el caso del catálogo HYG su información es válida para la época 2000, por lo que si se necesita utilizar en un sensor de estrellas real se debe de trasladar a la época al cual va a ser usado.

Reducción de los datos del catálogo HYG

En la práctica, no todas las estrellas dentro del catálogo HYG son necesarias para obtener una estimación precisa de la actitud de una nave espacial. Los sensores de estrellas comerciales emplean catálogos de estrellas con un número pequeño de estrellas, en el orden de unos miles de estrellas y aún son capaces de obtener altas precisiones en la estimación de la actitud. La reducción del número de estrellas en el catálogo simplifica los cálculos requeridos por un algoritmo de identificación de estrellas.

Se utilizan solo las estrellas con una **magnitud aparente** igual o menor de 6.0 para una primera simulación y el FOV circular fue puesto a 12.4° de diámetro, esto se hizo porque con este

valor se garantiza que por lo menos existen 3 estrellas en la imagen².

También se usaron estrellas de **magnitud aparente** igual o menor de 5.5 para una segunda simulación y en este caso se utilizó el FOV circular de 15° de diámetro. Después de este primer filtro el catálogo HYG de 120,000 estrellas es reducido a 5,068 estrellas cuando se usa una magnitud igual o menor de 6.0, y para el caso de la magnitud igual o menor de 5.5 se reduce a 2,865 estrellas.

El segundo filtro que se aplicó a las estrellas del catálogo está relacionado a la distancia entre estrellas tal como es vista por un sensor de imágenes. Específicamente, si la imagen tiene dos estrellas separadas por menos de 2 píxeles, estas dos estrellas serán vistas por este sensor como una sola estrella. Por esta razón, es necesario garantizar que exista una separación entre estrellas de por lo menos dos píxeles, este paso se ejecuta en el plano xy para garantizar el funcionamiento correcto del filtro.

Suponiendo a un sensor de imagen con una resolución de $2,000 \times 2,000$ píxeles ($N_x \times N_y$) y después de tomar en cuenta el criterio de los dos filtros, el catálogo de estrellas fue reducido a 5024 estrellas cuando se usa la **magnitud aparente** igual o menor de 6.0 y de 2841 estrellas para cuando se usa una **magnitud aparente** igual o menor de 5.5. En el resto de esta tesis, el catálogo de estrellas HYG reducido será simplemente referido como el catálogo de estrellas. Este catálogo tiene los mismos campos que tiene el catálogo HYG.

3.8. Algoritmos de Identificación de Estrellas

La navegación celeste es una técnica ampliamente usada por naves espaciales modernas. En el caso de naves espaciales orbitando la tierra, esta técnica es empleada en conjunción con otros métodos de navegación autónoma tal como el Sistema de Posicionamiento Global (GPS acrónimo del inglés Global Positioning System), magnetómetros, giróscopos, etc. La navegación celeste es la única manera para navegar para naves viajando al espacio profundo.

Al inicio de la era espacial los sistemas de navegación celestes usaban el seguimiento de una estrella como sistema de determinación de la actitud, los sistemas modernos dependen de la identificación de grupos de estrellas que se buscan dentro del catálogo de estrellas. Los algoritmos de identificación de estrellas aparecieron en los inicios de 1970 y varias técnicas han sido propuestas desde entonces.

²Tres estrellas usadas como vértices es el mínimo número de vértices requerido para formar a un polígono.

Los algoritmos de identificación de estrellas se clasifican en dos categorías básicas, los algoritmos del sistema llamado perdido en el espacio, en los que no existe una información previa de la actitud del satélite, y los algoritmos recursivos, en los cuales se tiene una información previa de la actitud (Spratling y Mortari, 2009). Existen diferentes algoritmos de identificación de estrellas, de acuerdo a las diferentes estrategias que se escogen para determinar la actitud a bordo, tales como, el algoritmo de coincidencia de triángulos, algoritmos de coincidencia de cuadrículas, algoritmos de descomposición en valores singulares, algoritmos que aplican redes neuronales (Jing y Liang, 2012), algoritmos genéticos, métodos de decisiones difusas, algoritmos de coincidencia de patrones de línea difusas, etc. Algunos de estos métodos son tan complejos que no pueden ser usados en los sensores de estrellas a bordo de un satélite, esto es debido al hardware usado en los sistemas espaciales.

3.9. Transformación de las coordenadas xy en pixeles a coordenadas esféricas.

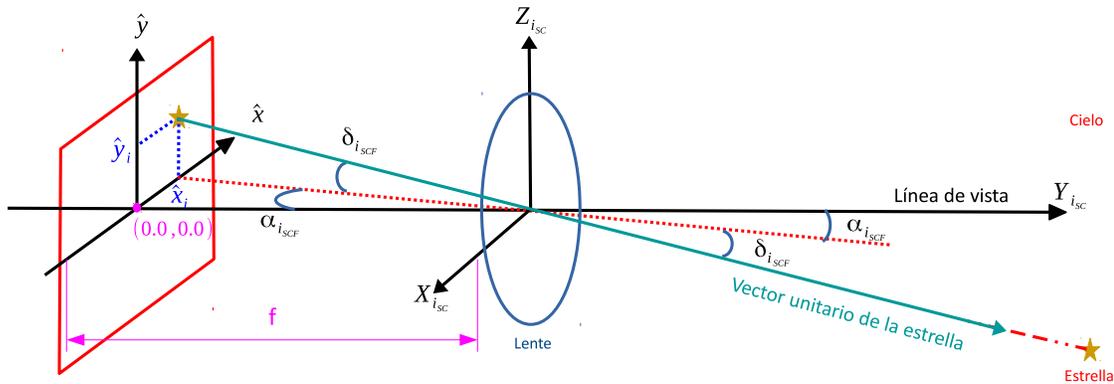


Figura 24: Geometría de un sensor de estrella (Wertz, 2012).

Considerando a los ejes \hat{x} y \hat{y} del eje de la línea de vista igual a cero en el cuadro de la imagen (SCF Star Coordinates Frame) y f la distancia focal tal como podemos observar en la Figura 24, las coordenadas en el cuadro del satélite son (Rad *et al.*, 2014),(Wertz, 2012):

$$\tan(\alpha_{iSCF}) = \frac{\hat{x}_i}{f}, \quad (24)$$

$$\tan(\delta_{iSCF}) = \frac{\hat{y}_i}{f} \cos(\alpha_{iSCF}), \quad (25)$$

por lo que:

$$\alpha_{iSCF} = \text{atan}\left(\frac{\hat{x}_i}{f}\right) \quad (26)$$

y

$$\delta_{i_{SCF}} = \text{atan}\left(\frac{\hat{y}_i}{f} \cos(\alpha_{i_{SCF}})\right). \quad (27)$$

Se tiene que (Rad *et al.*, 2014):

$$\vec{S}_{i_{SCF}} = \begin{pmatrix} X_{i_{SCF}} \\ Y_{i_{SCF}} \\ Z_{i_{SCF}} \end{pmatrix} = \begin{pmatrix} \cos(\alpha_{i_{SCF}}) \cos(\delta_{i_{SCF}}) \\ \sin(\alpha_{i_{SCF}}) \cos(\delta_{i_{SCF}}) \\ \sin(\delta_{i_{SCF}}) \end{pmatrix}. \quad (28)$$

Ya que la identificación de las estrellas ha sido realizada en la fase de identificación de polígonos, los vectores unitarios de todas las estrellas (CRF coordinates reference frame) del catálogo de estrella son (Wertz, 2012):

$$\vec{S}_{i_{CRF}} = \begin{pmatrix} x_{i_{CRF}} \\ y_{i_{CRF}} \\ z_{i_{CRF}} \end{pmatrix}_{CRF} = \begin{pmatrix} \cos(\alpha_{i_{CRF}}) \cos(\delta_{i_{CRF}}) \\ \sin(\alpha_{i_{CRF}}) \cos(\delta_{i_{CRF}}) \\ \sin(\delta_{i_{CRF}}) \end{pmatrix} \quad (29)$$

Los vectores de las estrellas del catálogo de estrellas son los vectores del cuadro de referencia \vec{F}_1 y los vectores de las estrellas identificadas en el sensor de estrellas son los vectores del cuadro del cuerpo \vec{F}_2 (Rad *et al.*, 2014).

3.10. Determinación de la actitud del satélite

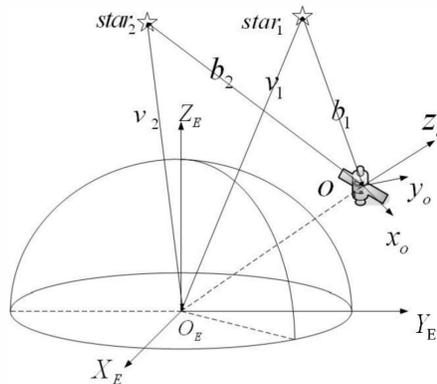


Figura 25: Vectores observados y sus vectores de referencias (Feng-gen *et al.*, 2011).

Existen varios métodos bien conocidos para calcular la determinación de la actitud con los que se obtienen los cuaterniones, éstos incluyen a los método-q de Davenport y el estimador de

cuaterniones QUEST ³. El método de la determinación de la actitud del vector de observación fue presentada por Wahba en 1965, el cual muestra que la matriz de la actitud óptima minimiza una función de pérdida. Basado en esto, Shuster y Oh propusieron los métodos QUEST y TRIAD. El método TRIAD utiliza solo dos vectores de observación y los vectores de referencia asociados con los que se tienen dos sistemas de coordenadas rectangulares y se obtiene a la matriz de transferencia entre ellos (Feng-gen *et al.*, 2011). La más grande ventaja del algoritmo TRIAD es la simplicidad de los cálculos, pero existe un problema de asimetría.

Uno de los métodos estadísticos más usados es el algoritmo de QUEST, y resuelve el problema de Wahba. Markley propuso la descomposición en valores singulares (SVD) y la matriz de la actitud óptima rápida (FOAM), los cuales son métodos basados en la descomposición de los valores singulares. Mortari propuso el método de Eluer-q resolviendo el problema de Wahba indirectamente, también desarrolló los métodos de estimación del cuaternión óptimo ESOQ ⁴ y ESOQ-2.

Los métodos más usados para un sensor de estrellas son el algoritmo de TRIAD, QUEST, ESOQ y ESOQ-2. El problema del método de TRIAD es la menor precisión de sus cálculos, esto debido a que solo utiliza dos vectores de medición, los métodos más precisos y rápidos son los de QUEST, ESOQ y ESOQ-2, de estos 3 últimos métodos, los de ESOQ y ESOQ-2 son más rápidos en un 10 % que el de QUEST (Markley y Mortari, 2000), aunque el de QUEST ha sido utilizado por más tiempo y su autor especifica que ha medida que las simulaciones de los tres métodos son implementadas, las velocidades son comparables entre sí, por lo que se debería de decir que sus velocidades de cálculos son iguales (Shuster, 2006).

La importancia de haber realizado este capítulo es que pudimos dividir las tres áreas importantes de investigación dentro de los sensores de estrellas los cuales son:

- Procesamiento de imágenes.
- Identificación de estrellas.
- Determinación de la actitud.

En nuestro caso el área para realizar la investigación es en el de identificación de las estrellas, el cual se presenta en el siguiente capítulo.

³QUEST de la palabra en inglés QUaternion ESTimation

⁴ESOQ de la palabra ESTimators of the Optimal Quaternion

Capítulo 4. Método de Identificación de Estrellas

En este capítulo se presenta todo el estudio realizado así como el desarrollo de un nuevo método de identificación de estrellas utilizando el estado del arte.

El principio de funcionamiento de los sensores de estrellas es asociar estrellas en grupos u objetos celestes en la vecindad, y entonces identificar a éstos como un todo. Se presentan un número de métodos que han sido publicados los cuales ilustran la variedad de diferentes técnicas usadas para identificar a las estrellas.

Algunos algoritmos utilizan los ángulos esféricos entre estrellas ¹ y el brillo de las estrellas. Una subcategoría de la clasificación anterior es la identificación de las estrellas no dimensional, para los cuales el valor exacto de la separación angular no es requerido sino que se utilizan sus valores normalizados para ser insensibles a la pobre calibración de la cámara o de los parámetros variantes en el tiempo de la calibración de la cámara.

Gotlieb propuso el método de coincidencia de las distancias angulares entre las estrellas observadas y las del catálogo de estrellas (Smart y Green, 1977). Junkins *et al.* (1978) fue capaz de identificar tripletes de estrellas calculando los ángulos entre los pares de estrellas, éste tuvo la limitación de requerir una estimación a priori de la actitud de la nave espacial antes de que fuera capaz de funcionar en tiempo real. La razón fue que Junkins había usado subcatálogos del cielo, cada uno representando a una porción del cielo, esto se hace para acelerar los cálculos (Spratling y Mortari, 2009). Groth (1986) desarrolló un algoritmo de coincidencia de patrones para comparar dos listas que incluyen a los triángulos formados con las tres coordenadas de las estrellas seleccionadas. Kosik (1991) diseñó un método de orientación de la distancia angular la cual es mejor técnica que el de los polígonos usada por Gotlieb. Baldini y Foggi (1993) propuso un método de identificación de estrellas multietapas que identifica a un número determinado de las estrellas más brillantes en una imagen dada, y mide la separación angular de la secuencia de cinco estrellas, después hace una búsqueda lineal de cada estrella en el catálogo el cual cae dentro un intervalo de tolerancia de las estrellas observadas (Spratling y Mortari, 2009).

Liebe (1995) desarrolló un método para la identificación de las estrellas en donde la distancia angular para la primera y la segunda estrella vecina es almacenada para todas las estrellas del cielo. Esto es, se almacena una estrella en el catálogo por su distancia angular a la primera estrella vecina, su distancia angular a la segunda estrella vecina y el ángulo interno entre las dos

¹el ángulo entre las líneas de vista de dos estrellas desde la perspectiva de una cámara

estrellas vecinas. Scholl M. S. (1995) publicó un método más sencillo, los ángulos interestelares fueron ordenados por su brillantez relativa, eliminando las permutaciones que existen cuando se considera el orden posible de las tres estrellas.

El algoritmo de Quine y Durrant-Whyte (1996) identifica una estrella particular y la magnitud de sus estrellas vecinas más cercanas. El número de mediciones independientes que se pueden hacer es independiente del número de estrellas consideradas por el algoritmo de identificación. Ya que las mediciones de la magnitud y separación son cuantitativas, cada medición independiente puede ser usada varias veces por una búsqueda en árbol binaria para subdividir un conjunto de estrellas.

Padgett y Kreutz-Delgado (1997) introdujo el método de rejilla asociado con un reconocimiento de patrón en donde el algoritmo propaga la posición de las estrellas vecinas como elementos dentro de la rejilla imprecisa. En este caso cada estrella está relacionada con un patrón bien definido que puede ser determinado por las estrellas circundantes. Lindsey *et al.* (1997) presentó un método para la identificación de las constelaciones de estrellas inspiradas por una técnica de redes neurales. Éste compara los vectores característicos derivados de los histogramas de distancias a múltiples estrellas alrededor de la estrella desconocida.

Mortari *et al.* (2001, 2004) propuso el algoritmo piramidal el cual utiliza un mínimo de cuatro estrellas para la extracción característica y la creación de patrones. En adición, este método utiliza una técnica de permutación óptima para explotar la habilidad del algoritmo para seleccionar cuales estrellas coinciden y una estrategia de búsqueda rápida llamada *k*-vector. Paladugu *et al.* (2003) presentó un estudio usando algoritmos genéticos para resolver el problema de reconocimientos de patrones de estrellas asociado a los sistemas de determinación de la actitud de un sensor de estrellas. El algoritmo genético minimiza la discrepancia entre las características de las estrellas dentro del campo de visión (FOV) actual y un FOV candidato seleccionado del mapeo de estrellas para determinar las coordenadas inerciales de la línea de vista del FOV. LIU *et al.* (2004) formuló un algoritmo de identificación de patrones de estrellas para el guiado celeste basados en polígonos convexos utilizando estrellas como vértices e independientes de sus magnitudes, almacenando los segmentos de líneas y ángulos interiores comenzando con el segmento más corto a lo largo de la dirección en sentido horario.

Zhang G. y J. (2008) propuso la identificación de estrellas autónomo del cielo completo. Un patrón de la estrella es generado en la forma de una característica radial y cíclica. Debido a que el

patrón es una característica fidedigna, es usado para la coincidencia inicial y el patrón cíclico es usado en la etapa de re-coincidencia. Kolomenkin *et al.* (2008) presentó un algoritmo basado en un esquema de votación geométrico en el cual un par de estrellas en el catálogo vota por un par de estrellas en la imagen si la distancia angular entre las estrellas de ambos pares son similares.

Jing y Liang (2012) presentó un método rápido de identificación de estrellas mejorado basados en redes neuronales (NN). El método propuesto está compuesto de tres niveles, siendo el primero la clasificación burda de las estrellas para la navegación, después el reconocimiento de patrones de estrellas basados en NN y la tercera es la validación final de los resultados del reconocimiento. La distancia angular del triángulo característico es empleado para la clasificación burda para activar las subredes para el reconocimiento de patrones de estrellas. Entonces el patrón de estrellas obtenido por el método de rejilla para la estrella principal es usada para identificar los patrones de estrellas.

Wei *et al.* (2009) propuso un algoritmo de identificación de estrellas basado en la transformación Log-Polar. Al inicio la transformación Log-Polar es usada para generar los patrones de estrellas. Después, los patrones de estrellas generados son codificados en cadenas de texto para el propósito de reconocimiento rápido y ahorro de memoria. Finalmente el reconocimiento de cadenas de texto modificado basado en el algoritmo de búsqueda Knuth—Morris—Pratt es usado para encontrar el reconocimiento entre las estrellas medidas y las estrellas del catálogo.

Para el caso de un campo de visión de $4^\circ \times 4^\circ$ y con una magnitud aparente máxima de 6,0, más del 70 % de las imágenes capturadas tienen 2 estrellas o menos (Ho, 2012).

En esta tesis se escogió un FOV circular de 12.4° de diámetro y con una magnitud aparente máxima de 6.0, debido a que con ese ángulo se pueden observar mínimo 3 estrellas y hasta un máximo de 59 estrellas, aunque pueden existir partes en el espacio que no se vean estrellas. Para el caso de magnitud aparente máxima de 5.5 se usó un FOV circular de 15° de diámetro.

4.1. Identificación de estrellas con polígonos

Si consideramos a las estrellas como puntos proyectados en el plano xy , cada estrella es igual a otra (de la misma manera que un punto no puede ser distinguido de otro). Por lo que se necesita agregar un contexto para distinguir a los puntos. La manera común para proporcionar el contexto a una estrella es asociar a ésta un grupo de objetos celestes vecinos. En este trabajo se procede de una manera similar, se identifica únicamente a una estrella por identificar su vecindad. Se

asocia un polígono a cada una de las estrellas del catálogo. Este polígono será formado de forma canónica, será similar en el sentido matemático a los polígonos que se obtienen de un sensor de imágenes.

De esta manera, si dos polígonos coinciden en ausencia de ruido se puede concluir que el grupo de estrellas asociados a éstas también van a coincidir. Este hecho será suficiente para identificar a las estrellas del catálogo. La técnica es derivada del método desarrollado por Chávez *et al.* (2013).

4.1.1. Invariante de semejanza para polígonos

Se definen a los puntos (x, y) en el plano de los números complejos correspondientes $z = x + jy$, en donde $j = \sqrt{-1}$. Un polígono en el plano complejo será un conjunto de puntos ordenados, en donde el *orden determina a los vértices consecutivos*. El corrimiento cíclico $(z_2, z_3, \dots, z_n, z_1)$, $(z_3, z_4, \dots, z_1, z_2)$, \dots , $(z_n, z_1, \dots, z_{n-2}, z_{n-1})$ y el ordenamiento inverso de los puntos $(z_n, z_{n-1}, \dots, z_2, z_1)$ tienen un ordenamiento diferente para el mismo polígono $(z_1, z_2, \dots, z_{n-1}, z_n)$. Una permutación general $p : \{1, 2, \dots, n\} \rightarrow \{1, 2, \dots, n\}$ puede formar a un polígono distinto $(z_{p(1)}, \dots, z_{p(n)})$ debido a que los vértices consecutivos no son los mismos. Un polígono canónico tiene un orden particular de los vértices.

Una transformación afín $f : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ puede ser escrita en términos de sumas y productos de números complejos tal como se puede observar en la siguiente ecuación:

$$f(z) = \alpha z + \beta \bar{z} + \gamma, \quad (30)$$

en donde $\alpha, \beta, \gamma \in \mathbb{C}$ y $|\alpha|^2 - |\beta|^2 = \det f \neq 0$. Aquí \bar{z} se refiere al complejo conjugado de z . Una transformación afín compleja o *transformación de semejanza* se refiere a las operaciones de escalamiento, rotación y desplazamiento.

En los polígonos dados $Z = (z_1, z_2, \dots, z_n) \in \mathbb{C}^n$ y $W = (w_1, w_2, \dots, w_n) \in \mathbb{C}^n$, el problema consiste en determinar si existe una transformación afín compleja f tal que $Z = f(W)$.

Podemos suponer que tenemos una colección de polígonos dada $Z_1, Z_2, \dots, Z_{\mathcal{N}}$ (esto corresponde a las estrellas del catálogo) y un polígono de búsqueda W , y se quiere saber cual de los Z_i son polígonos afín complejas de W . Utilizando una aproximación secuencial la solución puede ser encontrada en $O(\mathcal{N})$ operaciones. En general sin un índice, la complejidad depende linealmente

te del número de polígonos en la colección multiplicada por la complejidad de una coincidencia individual.

La aproximación de la coincidencia de polígonos bajo transformaciones de semejanza involucra la construcción de funciones escalares complejas $\varphi_\ell : \mathbb{C}^n \rightarrow \mathbb{C}$, $\ell = 1, \dots, \lfloor (n-1)/2 \rfloor$. La n -ésima potencia φ_ℓ^n es definida sobre el espacio del plano orientado n -gons, en el sentido que φ_ℓ^n es invariante bajo el corrimiento cíclico en las coordenadas \mathbb{C}^n , pero no es invariante bajo permutación general en las coordenadas (el cual genera otros polígonos). Estas funciones escalares son invariantes bajo transformaciones de semejanza actuando sobre los polígonos. Sin embargo la probabilidad que dos polígonos no similares que sean mapeadas al mismo escalar bajo cualquier función invariante es cero. Las funciones invariantes son analíticas; esto implica que una pequeña perturbación en un polígono nos lleva a un pequeño cambio controlado en la imagen correspondiente de la transformación de semejanza.

Con las herramientas anteriores se puede encontrar a todos los polígonos que coinciden en una colección y esto se puede hacer muy rápido después de mapear a todos los polígonos en la colección de números complejos. Es importante resaltar que todos los polígonos similares² serán mapeados al mismo número complejo φ_ℓ , el cual se llamará de ahora en adelante solo como *invariante*.

se supone $n \geq 3$.

Definición 4..1. Para cualquier $\ell = 1, \dots, \lfloor (n-1)/2 \rfloor$ debemos de considerar la función $\varphi_\ell : \mathbb{C}^n \rightarrow \mathbb{C} \cup \{\infty\}$ dado por

$$\varphi_\ell(z_1, \dots, z_n) = \frac{\sum_{k=1}^n \lambda^{\ell k} z_k}{\sum_{k=1}^n \lambda^{-\ell k} z_k}, \quad (31)$$

en donde $\lambda = e^{2\pi j/n}$ es la n sima raíz de la unidad.

Proposición 4..2. φ_ℓ es invariante bajo la acción de la transformación de semejanza en los polígonos con n vértices; esto es, si $\alpha, \gamma \in \mathbb{C}$ con $\alpha \neq 0$, entonces

$$\varphi_\ell(\alpha z_1 + \gamma, \alpha z_2 + \gamma, \dots, \alpha z_n + \gamma) = \varphi_\ell(z_1, z_2, \dots, z_n).$$

² Consideremos que dos polígonos son similares si alguno de éstos puede ser descrito como el resultado de una transformación de semejanza del otro.

Demostración.

$$\begin{aligned} \varphi_\ell(\alpha z_1 + \gamma, \alpha z_2 + \gamma, \dots, \alpha z_n + \gamma) &= \frac{\sum_{k=1}^n \lambda^{\ell k} (\alpha z_k + \gamma)}{\sum_{k=1}^n \lambda^{-\ell k} (\alpha z_k + \gamma)} = \\ &= \frac{\alpha \sum_{k=1}^n \lambda^{\ell k} z_k + \gamma \sum_{k=1}^n \lambda^{\ell k}}{\alpha \sum_{k=1}^n \lambda^{-\ell k} z_k + \gamma \sum_{k=1}^n \lambda^{-\ell k}} = \frac{\alpha \sum_{k=1}^n \lambda^{\ell k} z_k + \gamma \sum_{k=0}^{n-1} \lambda^{\ell k}}{\alpha \sum_{k=1}^n \lambda^{-\ell k} z_k + \gamma \sum_{k=0}^{n-1} \lambda^{-\ell k}} \\ &= \frac{\alpha \sum_{k=1}^n \lambda^{\ell k} z_k + \gamma(\lambda^{\ell n} - 1)/(\lambda^\ell - 1)}{\alpha \sum_{k=1}^n \lambda^{-\ell k} z_k + \gamma(\lambda^{-\ell n} - 1)/(\lambda^{-\ell} - 1)} = \\ &= \frac{\alpha \sum_{k=1}^n \lambda^{\ell k} z_k}{\alpha \sum_{k=1}^n \lambda^{-\ell k} z_k} = \frac{\sum_{k=1}^n \lambda^{\ell k} z_k}{\sum_{k=1}^n \lambda^{-\ell k} z_k} = \varphi_\ell(z_1, z_2, \dots, z_n). \end{aligned}$$

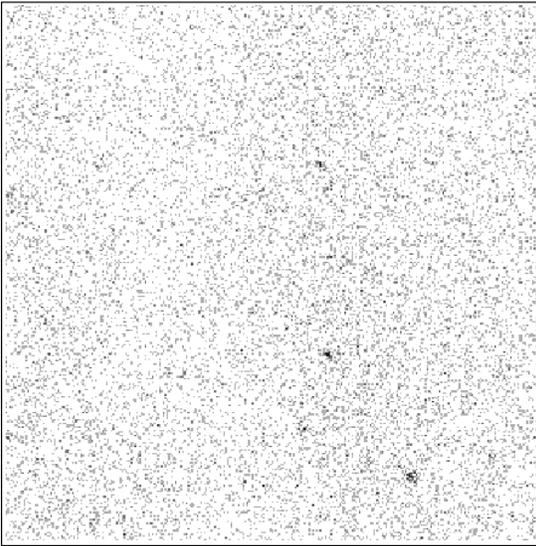


Figura 26: Imagen original de un lugar en el espacio y la imagen borrosa resultado del filtro promedio.

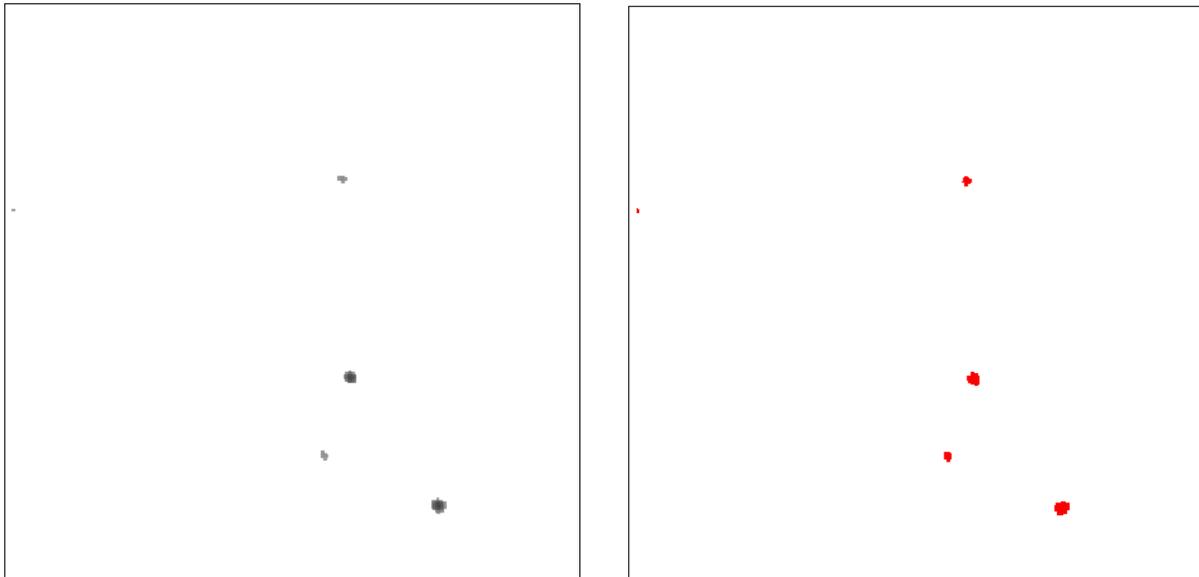


Figura 27: Visualización de centroides de las estrellas en la imagen de niveles de gris, e imagen blanco y rojo para mejorar la visualización de las estrellas dentro de la imagen.

Se verificó este método por simulación en Matlab, se utilizaron los centroides de la imagen de estrellas que se muestran en las Figuras 26 y 27, estos centroides son los vértices que forman al polígono, con los que se calculó la invariante y se comprobó que las invariantes calculadas no cambia de valor aún cuando se les aplica 20 diferentes escalamientos, rotaciones y desplazamientos al polígono original, tal como podemos observar en la Figura 28:

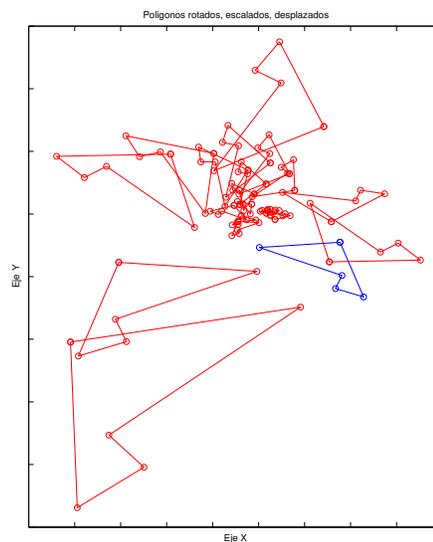


Figura 28: 20 diferentes escalamientos, rotaciones y desplazamientos al polígono original.

Se presenta la Tabla 6 de las invariantes calculadas para las transformaciones aplicadas.

Tabla 6: Resultados del cálculo de la Invariante.

# Transformación	Invariante calculada
1	$-1.07958463627345 + 1.37840028010926i$
2	$-1.07958463627345 + 1.37840028010926i$
3	$-1.07958463627345 + 1.37840028010926i$
4	$-1.07958463627345 + 1.37840028010926i$
5	$-1.07958463627345 + 1.37840028010926i$
6	$-1.07958463627345 + 1.37840028010926i$
7	$-1.07958463627345 + 1.37840028010926i$
8	$-1.07958463627345 + 1.37840028010926i$
9	$-1.07958463627345 + 1.37840028010926i$
10	$-1.07958463627345 + 1.37840028010926i$
11	$-1.07958463627345 + 1.37840028010926i$
12	$-1.07958463627345 + 1.37840028010926i$
13	$-1.07958463627345 + 1.37840028010926i$
14	$-1.07958463627345 + 1.37840028010926i$
15	$-1.07958463627345 + 1.37840028010926i$
16	$-1.07958463627345 + 1.37840028010926i$
17	$-1.07958463627345 + 1.37840028010926i$
18	$-1.07958463627345 + 1.37840028010926i$
19	$-1.07958463627345 + 1.37840028010926i$
20	$-1.07958463627345 + 1.37840028010926i$

4.2. Creación de la base de datos de invariante

Una vez que se conoce como calcular la invariante φ_ℓ , la próxima etapa consiste en generar la base de datos de las invariantes de cada uno de los polígonos asociados a todas las estrellas del catálogo. Si se tiene que s es una estrella del catálogo S , cada estrella en S tiene un número de identificación único el cual será llamado ID_s .

En el mundo real cualquier sistema de identificación de estrellas será afectado por varias fuentes de ruido posicionales, tales como: el ruido del detector y errores sistemáticos introducidos por los algoritmos de estimación de los centroides. En este trabajo se usaron las estrellas del catálogo S como vértices para crear a los polígonos. Se agregó un distinto nivel de ruido posicional distribuido uniformemente independiente a las coordenadas de cada estrella para simular a los errores introducidos por las fuentes de ruido mencionadas arriba, en este escrito a todos estos ruidos posicionales serán referidos solo como *ruido*.

Idealmente, en ausencia de ruido cada estrella s debe de tener un polígono único asociado a ésta, y cada polígono tiene una invariante φ_ℓ asociada a ésta también, el cual será llamada w^s . Debido al nivel de ruido introducido a las estrellas dentro del campo de visión se tendrá una forma diferente del polígono cada vez que se aplica un nivel de ruido distinto a las mismas estrellas y para cada polígono se tendrá un w^s diferente asociada a éste también. El conjunto de polígonos generará un área de invariantes $[w_{\min}^s, w_{\max}^s]$ o podrían generarse varias áreas de invariantes $[w_{\min_p}^s, w_{\max_p}^s]$, en donde $p = \{1, \dots, P\}$ y P es el número de polígonos con diferentes formas

asociadas a la misma estrella s , este fenómeno es una consecuencia del nivel de ruido introducido.

Las áreas de las invariantes $[w_{\min_p}^{s_1}, w_{\max_p}^{s_1}]$ asociadas a una estrella s_1 se pueden traslapar con otras áreas de invariantes $[w_{\min_p}^{s_2}, w_{\max_p}^{s_2}]$ asociadas a una diferente estrella s_2 . En el presente trabajo se le llamará *colisión* al hecho de que las áreas de dos o más invariantes se traslapan.

En la siguiente subsección se explicará como generar al polígono para cada estrella en el catálogo, y también se dará detalle sobre la generación de la base de datos de la invariante $(ID_s, [w_{\min_p}^s, w_{\max_p}^s])$. Es importante recalcar que después que el catálogo de estrellas ha sido determinado, la base de datos de la invariante necesita ser calculado solo una vez. Si el catálogo de estrellas es modificado o la apertura del FOV es cambiado, es necesario recalcar la base de datos de las invariantes.

4.2.1. Projectando estrellas de la esfera celeste al plano xy

La estrella bajo consideración será llamada la estrella central (CS), ésta se encuentra localizada en el centro del FOV circular. Tenemos que k_{FOV} es el número de estrellas que se encuentran dentro del FOV y que s_i es la i -ésima estrella dentro del FOV, en donde $i \in [1, \dots, k_{\text{FOV}}]$.

Todas las estrellas dentro del FOV son proyectadas en del plano xy de acuerdo a las siguientes ecuaciones (Jiancheng *et al.*, 2009) y (Smart y Green, 1977):

$$x_{s_i} = \frac{\cos(\delta_i) \sin(\alpha_i - \alpha_1)}{\sin(\delta_i) \sin(\delta_1) + \cos(\delta_i) \cos(\delta_1) \cos(\alpha_i - \alpha_1)} \quad (32)$$

$$y_{s_i} = \frac{\sin(\delta_i) \cos(\delta_1) - \cos(\delta_i) \sin(\delta_1) \cos(\alpha_i - \alpha_1)}{\sin(\delta_i) \sin(\delta_1) + \cos(\delta_i) \cos(\delta_1) \cos(\alpha_i - \alpha_1)} \quad (33)$$

δ_i y α_i son la declinación y la ascensión de las coordenadas celestes de la estrella bajo análisis (s_i), δ_1 y α_1 son las coordenadas esféricas de la dirección de la línea de vista del sensor localizada en el origen del plano xy , i.e. $(0, 0)$.

Para generar la base de datos, se introduce ruido a las posiciones de las estrellas proyectadas en el plano xy :

$$x'_{s_i} = x_{s_i} + \varepsilon_x \quad (34)$$

$$y'_{s_i} = y_{s_i} + \varepsilon_y \quad (35)$$

en donde ε_x y ε_y es el ruido distribuido uniformemente en el intervalo $[-\eta, \eta]$. La Figura 29 ilustra cómo las estrellas en la esfera celeste son proyectadas en el plano xy .

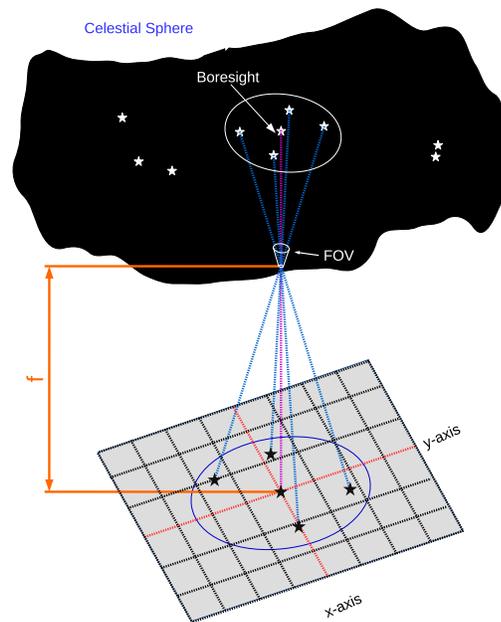


Figura 29: Estrellas de la esfera celeste proyectadas en el plano xy .

Como se mencionó anteriormente se introduce un nivel de ruido distinto a las coordenadas esféricas de cada una de las estrellas dentro del FOV. El nivel de ruido introducido es uniformemente distribuido en el intervalo de $[-\eta, \eta]$.

4.2.2. Creación de Polígonos con las proyecciones de las estrellas en el plano xy .

Una vez que el nivel de ruido ha sido agregado a las coordenadas esféricas de cada una de las estrellas dentro del FOV de 12.4 grados, se pueden formar los polígonos. Es posible que en una región particular del cielo con un FOV fijo, el número de estrellas vecinas k_{FOV} sea más pequeña que el número de vértices n predeterminado para formar a un polígono, en este caso se usará el número de estrellas vecinas disponibles para calcular la invariante, por lo que k_{FOV} será el número de vértices seleccionado para formar al polígono, y esto hace que se ajuste k al número de vértices en el polígono tal que $k = \min(n, k_{\text{FOV}})$.

La proyección de CS en el plano xy es siempre considerada como el primer vértice del polígono canónico y está localizado en el origen $x_{s_1}=0$ y $y_{s_1}=0$, que es $v_1 = (0, 0)$ en la Figura 30.

Las estrellas dentro del FOV son ordenados de forma ascendente de acuerdo a la distancia Euclidiana entre v_1 y la proyección en el plano xy de cada uno de las estrellas $k_{\text{FOV}} - 1$. Por lo que obtenemos el conjunto $\{v'_2, v'_3, \dots, v'_k\}$ en donde $v_2 = v'_2$ y es la estrella más cercana en la vecindad de v_1 , v'_3 es la segunda más cercana, y así todas las demás. Ahora se ordena

$\{v'_3, v'_4, \dots, v'_k\}$ por el ángulo θ orientado en sentido antihorario con respecto al vector $\overrightarrow{v_1 - v_2}$, en donde $\theta_1 < \theta_2 < \dots < \theta_{k-2}$.

Por lo que tenemos que los vértices que pertenecen al polígono son $\{v_1, v_2, v_3, \dots, v_k\}$ en donde $v_3 = v'_i \in \{v'_3, v'_4, \dots, v'_k\}$ con el ángulo más pequeño con respecto a $\overrightarrow{v_1 - v_2}$ y así sucesivamente. El método propuesto garantiza que la forma del polígono será siempre el mismo no importando el ángulo de rotación de la imagen. El procedimiento de ordenar angularmente usado es parecido a uno propuesto por Xie *et al.* (2012) y Xie y Wang (2012), sin embargo, en estos trabajos los autores no utilizan el conjunto de vértices como un polígono. El procedimiento propuesto está ilustrado en la Figura 30.

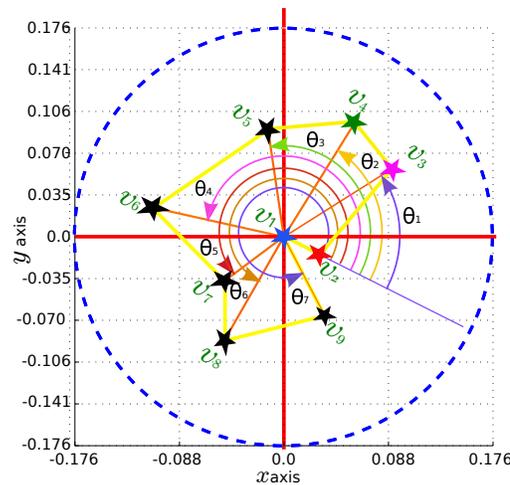


Figura 30: Orden de los vértices en el plano xy .

Como se mencionó anteriormente si se introduce un diferente nivel de ruido a cada una de las estrellas dentro del FOV, se crea un polígono, y si esto se hace varias veces, se crean distintos polígonos, los polígonos resultantes pueden tener casi la misma forma como se muestra en la Figura 31a o pueden tener formas completamente distintas tal como podemos observar en la Figura 31b, estos últimos polígonos serán referidos de ahora en adelante como *multipolígonos*.

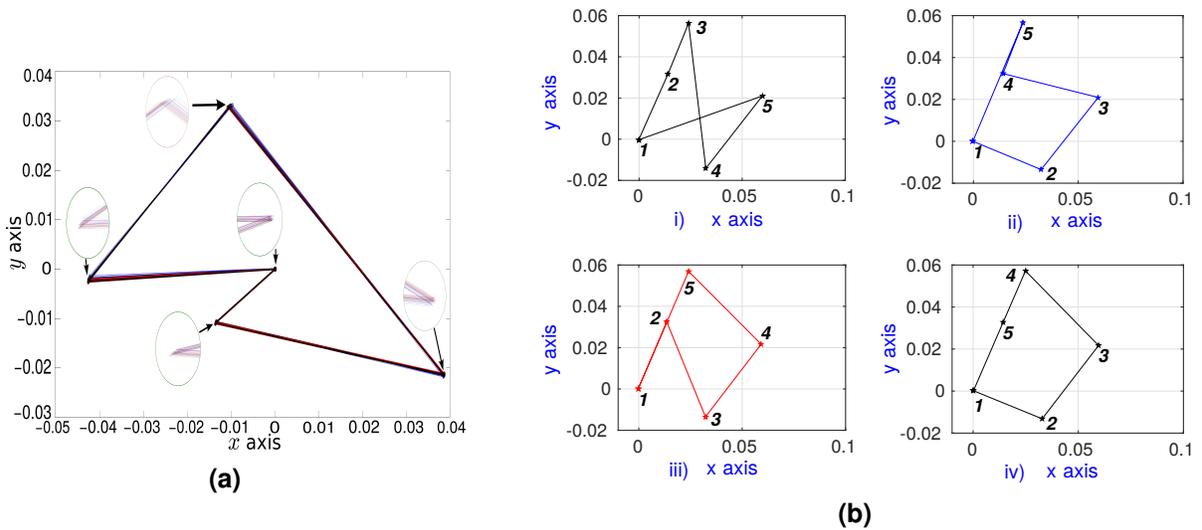


Figura 31: Polígonos ordenados por distancia Euclidiana y ángulo a) polígonos generados los cuales sus formas son muy parecidas b) cuatro diferentes formas de polígonos generados utilizando a la misma estrella central CS 87044.

En la Figura 31a podemos notar que las formas de los polígonos son muy parecidos a pesar del ruido introducido a todas las estrellas dentro del FOV. Para un área distinta del cielo tal como en la vecindad de la estrella CS 87044, podemos encontrar las estrellas tal como es mostrada en la Figura 31b. En esta área del espacio la forma del polígono varía como una función del ruido introducido, y debido a que el segundo vértice es siempre el más cercano a la estrella CS 87044 para este caso este vértice puede ser la estrella con ID 86529 o la estrella con ID 86040 tal como podemos ver en la Tabla 7. Dependiendo del ruido introducido podemos tener cuatro diferentes formas del polígono para el mismo juego de vértices tal como podemos ver en este ejemplo.

Tabla 7: Datos de cuatro diferentes polígonos con la misma estrella CS 87044 generada en la misma área del espacio.

Polígono original sin ruido					Figura 31b.i Primer polígono					Figura 31b.ii Segundo polígono				
ID CS	x_{pixel}	y_{pixel}	dist	ángulo	ID CS	x_{pixel}	y_{pixel}	dist	ángulo	ID CS	x_{pixel}	y_{pixel}	dist	ángulo
87044	0.00000	0.00000			87044	-0.00020	-0.00031			87044	-0.00027	0.00001		
86529	0.03259	-0.01349	0.03527	337.51	86040	0.01401	0.03162	0.03494	66	86529	0.03228	-0.01349	0.03524	337.47
85211	0.05985	0.02134	0.06354	19.62	85262	0.02400	0.05628	0.06155	66.84	85211	0.05954	0.02094	0.06336	19.29
86040	0.01444	0.03222	0.03531	65.86	86529	0.03238	-0.01417	0.03541	336.95	86040	0.01411	0.03233	0.03537	66.01
85262	0.02427	0.05684	0.06180	66.87	85211	0.05982	0.02100	0.06369	19.54	85262	0.02353	0.05659	0.06138	67.18

Figure 31b.iii Tercer polígono					Figure 31b.iv Cuarto polígono				
ID CS	x_{pixel}	y_{pixel}	dist	ángulo	ID CS	x_{pixel}	y_{pixel}	dist	ángulo
87044	-0.00025	0.00016			87044	0.00016	0.00015		
86040	0.01352	0.03245	0.03510	66.91	86529	0.03284	-0.01318	0.03529	337.8
86529	0.03225	-0.01350	0.04962	337.2	85211	0.05966	0.02170	0.04400	19.91
85211	0.05918	0.02165	0.04428	19.88	85262	0.02488	0.05720	0.04970	66.57
85262	0.02408	0.05691	0.04975	66.79	86040	0.01423	0.03266	0.02674	66.59

La forma del polígono depende de la distribución de las estrellas dentro del FOV circular y del nivel de ruido introducido a cada una de las estrellas. Cada polígono con forma diferente serán referenciados al número p y cada grupo de polígonos en el multipolígono generaran un área de invariantes distinta $[w_{\min_p}^s, w_{\max_p}^s]$, donde $p = \{1, \dots, P\}$. y P es el número de grupos de polígonos

con forma distinta para la misma estrella CS.

Si los vértices del polígono son ordenados solamente por distancias Euclidianas para crear a los polígonos, como se ilustra en la Figura 32, se obtienen mayor número de polígonos con formas completamente diferentes para la misma estrella CS comparados al número de polígonos creados cuando se ordenan los vértices por distancia Euclidiana y ángulo como es propuesto.

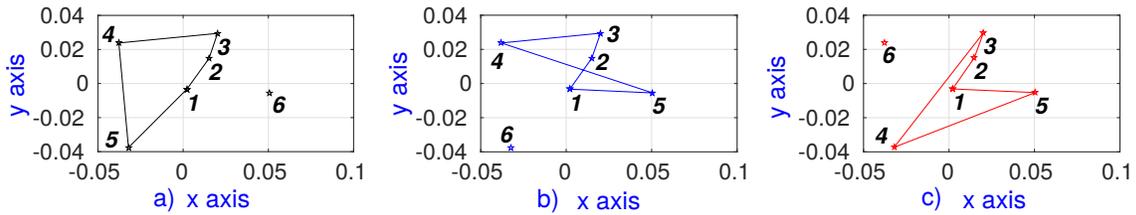


Figura 32: Figuras a), b) y c) son tres diferentes polígonos con la misma estrella CS 26529 generados en la misma área del cielo y ordenadas solamente con la distancia Euclidiana.

Tabla 8: Datos de tres diferentes polígonos con la misma estrella CS 26529 generados en la misma área del cielo.

Polígono original sin ruido				Figura 32.a Primer polígono				Figura 32.b Segundo polígono				Figura 32.c Tercer polígono			
ID CS	x_{pixel}	y_{pixel}	dist	ID CS	x_{pixel}	y_{pixel}	dist	ID CS	x_{pixel}	y_{pixel}	dist	ID CS	x_{pixel}	y_{pixel}	dist
26529	0.00000	0.00000	0.00000	26529	-0.00006	-0.00004	0.00000	26529	-0.00006	0.00009	0.00000	26529	0.00000	-0.00008	0.00000
26061	0.01286	0.01815	0.02224	26061	0.01281	0.01811	0.02225	26061	0.01261	0.01829	0.02217	26061	0.01301	0.01802	0.02229
25796	0.01807	0.03287	0.03751	25796	0.01792	0.03274	0.03739	25796	0.01799	0.03301	0.03754	25796	0.01802	0.03271	0.03741
27491	-0.03422	-0.03413	0.04833	26820	-0.04013	0.02701	0.04834	27491	-0.03426	-0.03405	0.04833	26820	-0.03992	0.02707	0.04828
26820	-0.04005	0.02719	0.04841	25749	0.04836	-0.00224	0.04847	25749	0.04827	-0.00205	0.04837	27491	-0.03424	-0.03433	0.04843
25749	0.04846	-0.00224	0.04851	27491	-0.03437	-0.03431	0.04849	26820	-0.04010	0.02729	0.04840	25749	0.04852	-0.00238	0.04858

En el caso en que el nivel de ruido es suficientemente alto para cambiar significativamente la distancia Euclidiana entre las estrellas vecinas y la estrella CS, como se muestra en la Tabla 8, un vértice que es parte de un polígono para cierto nivel de ruido se puede llegar a tener una distancia mayor que otra estrella que no es parte del polígono, por lo que este último llegará a ser parte del polígono. La Figura 32 ilustra los efectos del nivel de ruido introducido a la forma del polígono para la misma área del cielo.

4.2.3. Generación de base de datos de invariantes

Una vez que se conoce el procedimiento de calcular la invariante w^s de un polígono, la próxima etapa consiste en generar una base de datos de las invariantes de todos los polígonos asociados a todas las estrellas del catálogo. Almacenar todos los posibles valores de w^s después de agregar el nivel de ruido es inviable ya que existe un número infinito de diferentes polígonos que pueden ser creados. Debido a que la función de la invariante es analítica, por lo que pequeños cambios en la forma de los polígonos corresponde a un pequeño cambio en el valor de la invariante, lo que define a un área de invariantes en el plano complejo. Se puede explotar éste comportamiento del

área de la invariante por solo almacenar las dos esquinas que definen al área rectangular de la invariante. Estas esquinas se obtienen después de ejecutar al algoritmo un número muy grande e introduciendo diferentes niveles de ruido a cada una de las estrellas dentro del FOV, en este trabajo el programa se ejecutó un millón de veces y a cada estrella se le introdujo un nivel de ruido diferente cada vez que se ejecutaba al programa.

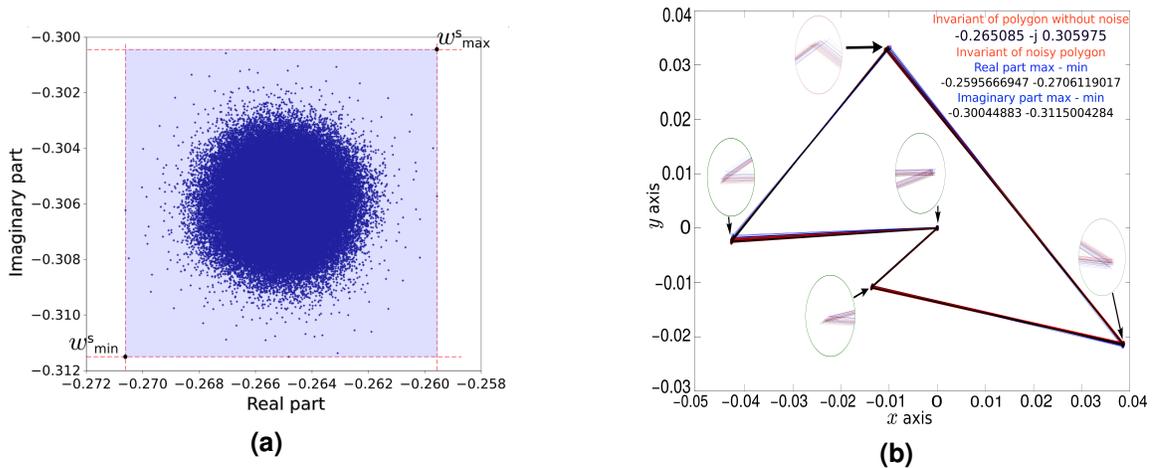


Figura 33: a) Comportamiento de la invariante debido al ruido introducido a los centroides de las estrellas, b) El conjunto de polígonos resultantes.

Tenemos que w_{\min}^s es la esquina inferior izquierda y w_{\max}^s es la esquina superior derecha que definen al área de la invariante rectangular. De esta manera, si los polígonos generados son cercanamente idénticos después de agregar el nivel de ruido, éstos son considerados dentro de un área de invariante solamente $[w_{\min_1}^s, w_{\max_1}^s]$, como es ilustrado por el área de color morado claro de la Figura 33.

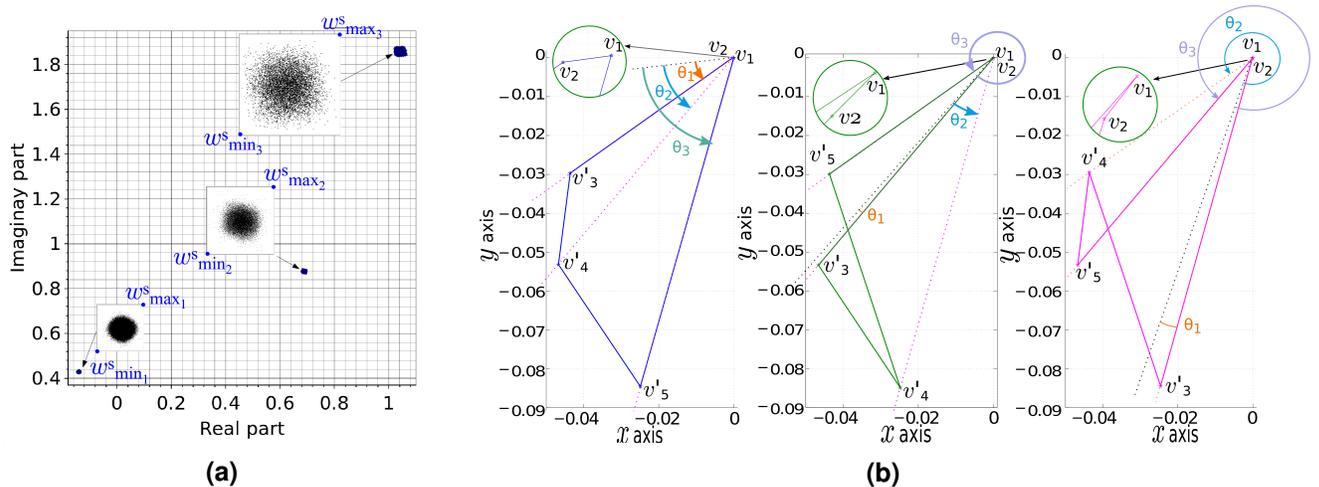


Figura 34: a) Comportamiento de la invariante debido al ruido introducido a los centroides de las estrellas, b) multi-polígonos resultantes.

En ciertos casos, a pesar de que la función de la invariante es analítica, el ruido introducido puede modificar cuales estrellas pueden ser usadas como vértices del polígono o el orden de los vértices y esto cambia dramáticamente la forma del polígono dando lugar a múltiples polígonos, los cuales llamaremos multipolígonos, los cuales están asociados a la misma CS. En consecuencia el multipolígono tiene P áreas de invariantes no traslapadas $[w_{\min_1}^s, w_{\max_1}^s], \dots, [w_{\min_P}^s, w_{\max_P}^s]$ asociadas a cada CS, este caso se puede observar en la Figura 34 que tiene 3 grupos de polígonos con forma distinta o sea $P = 3$.

Armado con las herramientas previamente descritas podemos proceder a diseñar un algoritmo para indexar todas las estrellas en el catálogo tal como el algoritmo 1.

Algoritmo 1 Indexando estrellas con ruido.

```

1: INPUT: Catálogo de estrellas  $S$ ,  $k_{FOV}$  número de estrellas vecinas,  $n$  número de vértices del polígono para calcular
   la invariante. cada  $s \in S$ 
2:  $P \leftarrow 0$  1 to 1,000,000  $k_{FOV} < n$ 
3:  $k = k_{FOV}$ 
4:  $k = n$ 
5:  $v_1 \leftarrow (Txy(s_1) + \varepsilon_{xy_1})$  { $Txy()$  función que transforma al plano  $xy$ }
6:  $ns_2 \leftarrow (Txy(s_2) + \varepsilon_{xy_2})$ 
7:  $ns_3 \leftarrow (Txy(s_3) + \varepsilon_{xy_3})$ 
8:  $\vdots$ 
9:  $ns_k \leftarrow (Txy(s_{(k)}) + \varepsilon_{xy_k})$  { $\varepsilon_{xy}$  función que introduce ruido}
10:  $\{v'_2, v'_3, \dots, v'_k\} \leftarrow \mathbf{k}\text{-NN}(ns)$ 
11:  $\{v'_i$  es la  $i$ -ésima estrella vecina más cercana de  $v_1$   $\}$ 
12:
13:  $\{v_2 \leftarrow v'_2\}$ 
14: Sort $\{v'_3, \dots, v'_k\}$  {Ordena por ángulo con respecto a  $v_2$ }
15: { en donde  $v_3, \dots, v_n = v'_i \in \{v'_3, v'_4, \dots, v'_k\}$  con }
16: { el ángulo más pequeño con respecto a  $v_1 - v_2$  }
17:
18:  $Z = (v_1, v_2, v_3, \dots, v_k)$  {Un polígono of  $k$  vértices}
19:  $w^s = \varphi_\ell(Z)$  { $\ell = [1, \dots, k/2]$ }
20: orden de los vértices actual  $\neq$  previo orden de vértices
21:  $P ++$  {Número de polígonos con diferentes}
   {orden de vértices para la}
   {misma CS ID}
22:
23:  $p = P$   $w^s \leq w_{\min}^s(p)$ 
24:  $w_{\min}^s(p) = w^s$   $w^s \geq w_{\max}^s(p)$ 
25:  $w_{\max}^s(p) = w^s$ 
26: Almacenar  $[ID_s, k, P, w_{\max}^s(1), w_{\min}^s(1), \dots, w_{\max}^s(P), w_{\min}^s(P)]$ 
27: { Esta tupla representa la estrella  $s$  }

```

Al principio uno espera que la base de datos de las invariantes completa contenga 5,024 polígonos después de ejecutar al Algoritmo 1, el cual es el número de estrellas en el catálogo, pero esta base de datos contiene más polígonos como un resultado del nivel de ruido agregado. El número total de polígonos depende del nivel de ruido y del número de vértices usados para formar a los polígonos. En esta tesis se generaron varias bases de datos dependientes del número de

vértices predeterminados $n \in \{3, \dots, 9\}$ para formar al polígono. La Tabla 9 muestra el número total de polígonos como una función del número de vértices n y del ruido introducido. En la primera simulación se decidió introducir un nivel de ruido de $\eta = 53 \mu\text{rad}$ y en una segunda simulación se duplicó el nivel de ruido a $\eta = 106 \mu\text{rad}$ con el fin de probar la robustez de esta propuesta. Estos valores son más altos que otros niveles de ruido previamente reportados en la literatura de algoritmos de identificación de estrellas que han estado en órbita en misiones en el espacio real (Mortari *et al.*, 2004).

Tabla 9: Número total de polígonos en la base de datos de invariantes como una función del número de vértices y el nivel de ruido utilizando estrellas con magnitud aparente ≤ 6.0 .

base de datos n vértices	#Polígonos sin ruido	#Polígonos $\pm 53 \mu\text{rad}$ de ruido	#Polígonos $\pm 106 \mu\text{rad}$ de ruido	#Polígonos $\pm 212 \mu\text{rad}$ de ruido	#Polígonos $\pm 318 \mu\text{rad}$ de ruido	#Polígonos $\pm 424 \mu\text{rad}$ de ruido
3	5,024	5266	5449	5818	6194	6535
4	5,024	5377	5651	6163	6775	7333
5	5,024	5446	5782	6430	7119	7822
6	5,024	5511	5940	6669	7510	8323
7	5,024	5590	6077	6889	7824	8746
8	5,024	5756	6304	7240	8281	9274
9	5,024	5717	6420	7442	8585	9831

Tabla 10: Número total de polígonos en la base de datos de invariantes como una función del número de vértices y el nivel de ruido utilizando estrellas con magnitud aparente ≤ 5.5 .

base de datos n vértices	#Polígonos sin ruido	#Polígonos $\pm 106 \mu\text{rad}$ de ruido	#Polígonos $\pm 212 \mu\text{rad}$ de ruido	#Polígonos $\pm 318 \mu\text{rad}$ de ruido	#Polígonos $\pm 424 \mu\text{rad}$ de ruido
3	2,841	3,062	3,248	3,456	3,639
4	2,841	3,170	3,430	3,723	3,975
5	2,841	3,274	3,594	3,920	4,273
6	2,841	3,360	3,743	4,158	4,575
7	2,841	3,448	3,902	4,398	4,968
8	2,841	3,558	4,068	4,619	5,236
9	2,841	3,608	4,152	4,757	5,391

Como se mencionó anteriormente, después de agregar diferentes niveles de ruido a las posiciones de las estrellas dentro del FOV y la misma CS se puede tener diferentes formas de los polígonos asociadas a éste. La Tabla 11 muestra el número máximo de polígonos P_{\max} que un multipolígono puede tener como una función del nivel de ruido y el número de vértices para la base de datos completa de las invariantes.

Tabla 11: Máximo número de polígonos P_{max} en la base de datos de invariantes completa que un multipolígono puede tener como una función del nivel de ruido y el número de vértices utilizando estrellas con magnitud aparente ≤ 6.0 .

base de datos n vértices	P_{max} $\pm 53 \mu\text{rad}$ de ruido	P_{max} $\pm 106 \mu\text{rad}$ de ruido	P_{max} $\pm 212 \mu\text{rad}$ de ruido	P_{max} $\pm 318 \mu\text{rad}$ de ruido	P_{max} $\pm 424 \mu\text{rad}$ de ruido
3	6	12	10	12	12
4	4	12	13	14	14
5	4	5	15	15	14
6	4	6	8	10	12
7	4	8	13	19	28
8	5	12	17	19	28
9	6	12	22	23	24

Tabla 12: Máximo número de polígonos P_{max} en la base de datos de invariantes completa que un multipolígono puede tener como una función del nivel de ruido y el número de vértices utilizando estrellas con magnitud aparente ≤ 5.5 .

base de datos n vértices	P_{max} $\pm 106 \mu\text{rad}$ de ruido	P_{max} $\pm 212 \mu\text{rad}$ de ruido	P_{max} $\pm 318 \mu\text{rad}$ de ruido	P_{max} $\pm 424 \mu\text{rad}$ de ruido
3	3	4	6	6
4	4	6	9	9
5	5	8	10	12
6	4	9	13	19
7	8	16	16	19
8	11	12	12	14
9	12	19	20	20

4.3. Imagen sintética de estrellas

Para evaluar la propuesta de identificación de estrellas se crearon *imágenes sintéticas* con las estrellas del catálogo. Estas imágenes se crearon proyectando las estrellas de un área de la esfera celeste en el plano xy , pero esta vez en pixeles. La lógica para hacerlo en pixeles es simular una imagen como si ésta fuera capturada con una cámara abordo de una nave espacial que utiliza un sensor de imagen con la resolución de (2000×2000) y un campo de visión circular (FOV) de 12.4° previamente descritas en la subsección 3.7.1. Sin perder generalidad, en este trabajo se supone que los centroides de cada estrella presente en la imagen ya han sido calculados. Las imágenes sintéticas son generadas directamente del catálogo de estrellas después de agregar ruido uniformemente distribuido a la posición de cada estrella. En esta sección se explica como las imágenes sintéticas son creadas y como los polígonos asociados a las estrellas en la imagen sintética son formados.

4.3.1. Creando imágenes sintéticas del catálogo de estrellas

Las imágenes sintéticas son generadas primeramente escogiendo una estrella del catálogo, la estrella seleccionada es considerada como la estrella central \widehat{CS} , la cual es colocada aleatoria-

mente en la vecindad del centro de la imagen sintética $\left(\frac{N_x}{2}, \frac{N_y}{2}\right)$. Es importante encontrar todas las estrellas que están localizados dentro del FOV circular de \widehat{CS} , el cual serán llamadas estrellas vecinas (NB). Después de encontrar las estrellas NB, se agrega ruido distribuido uniformemente a las posiciones de cada una de las estrellas encontradas.

Tenemos que \widehat{k}_{FOV} es el número de estrellas dentro del FOV de la imagen sintética y que \widehat{s}_i es la i -ésima estrella dentro del FOV, en donde $i \in [1, \dots, \widehat{k}_{FOV}]$. Las coordenadas celestes de \widehat{s}_i son convertidas a las coordenadas xy (Jiancheng F. y T., 2006; M. *et al.*, 2009) y tal como podemos observar en la Figura 35.

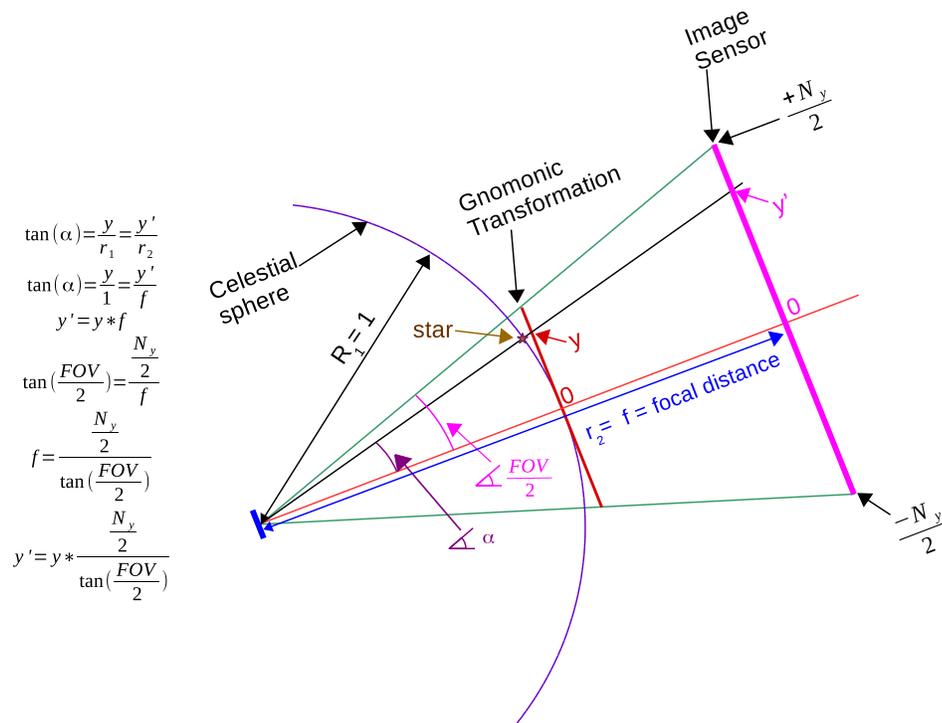


Figura 35: Gráfica de como las ecuaciones de transformación de coordenadas esféricas a píxeles son obtenidos.

Utilizando las ecuaciones (32) y (33) introducidos previamente en la sección 4.2 para después calcular la posición de las estrellas en la imagen sintética usando las ecuaciones (36) y (37) respectivamente.

$$\widehat{x}_{\widehat{s}_i} = \frac{N_x}{2} \cdot \frac{1}{\tan\left(\frac{FOV}{2.0}\right)} \cdot x_{\widehat{s}_i} + \frac{N_x}{2} + \varepsilon_{\widehat{x}} \quad (36)$$

$$\widehat{y}_{\widehat{s}_i} = \frac{N_y}{2} \cdot \frac{1}{\tan\left(\frac{FOV}{2.0}\right)} \cdot y_{\widehat{s}_i} + \frac{N_y}{2} + \varepsilon_{\widehat{y}} \quad (37)$$

en donde $\widehat{x}_{\widehat{s}_i}$ y $\widehat{y}_{\widehat{s}_i}$ están en píxeles, y $\varepsilon_{\widehat{x}}$, $\varepsilon_{\widehat{y}}$ son el nivel de ruido uniformemente distribuido introdu-

cido en el intervalo de $[-\hat{\eta}, \hat{\eta}]$. Como se comentó anteriormente se consideraron cinco escenarios de simulación, primero con un nivel de ruido de $\eta = 53 \mu\text{rad}$ esto es $\hat{\eta} \approx 0.5$ pixeles, después un nivel de ruido correspondiente a $\eta = 106 \mu\text{rad}$ esto es $\hat{\eta} \approx 1.0$ pixeles, seguido por incrementos de ruido de $106 \mu\text{rad}$ hasta $\eta = 424 \mu\text{rad}$ o $\hat{\eta} \approx 4$ pixeles.

La imagen sintética es rotada aleatoriamente multiplicando cada elemento en la imagen por una matriz de rotación de la forma tal como lo propone Luo *et al.* (2015)

$$R = \begin{pmatrix} \cos(\phi) & \sin(\phi) \\ -\sin(\phi) & \cos(\phi) \end{pmatrix} \quad (38)$$

en donde ϕ está uniformemente distribuido en el intervalo $[-\pi, \pi]$.

4.3.2. Creando polígonos usando estrellas como vértices en el plano de la imagen sintética

Es posible formar polígonos utilizando a las estrellas en la imagen sintética como vértices. El método que se usa para crear polígonos se describió en la subsección 4.2.2, y garantiza que la forma del polígono se mantiene igual no importando el ángulo de la rotación de la imagen bajo las mismas condiciones de niveles de ruido en cada una de las estrellas dentro del FOV.

Teniendo que \hat{n}' es el número de vértices seleccionados para formar al polígono, \hat{k} se ajusta al número de estrellas \hat{k}_{FOV} de la imagen actual si \hat{k}_{FOV} es menor que \hat{n}' tal que $\hat{k} = \min(\hat{n}', \hat{k}_{\text{FOV}})$, de la misma manera como fue explicado en la sección 4.2.2.

La proyección de $\widehat{\text{CS}}$ en el plano xy es la estrella más cercana al centro de la imagen, y siempre es considerado como el primer vértice $\hat{v}_1 = \hat{s}_{\widehat{\text{CS}}}$. En otras palabras $\widehat{\text{CS}}$ es la estrella que está localizado más cerca del origen $\hat{x} = 1000$ y $\hat{y} = 1000$. La imagen sintética es desplazada por multiplicar ésta por una matriz de rotación tal que la estrella CS sea puesta en el centro de una nueva imagen sintética (Ali *et al.*, 2006) y (Liebe, 2002). La matriz de rotación es obtenida de los vectores formados entre las coordenadas (x, y, z) de la estrella CS y las coordenadas del centro de la imagen sintética original.

Para encontrar los otros vértices se usa el mismo procedimiento descrito en la subsección 4.2.2, el procedimiento es ilustrado en la Figura 36.

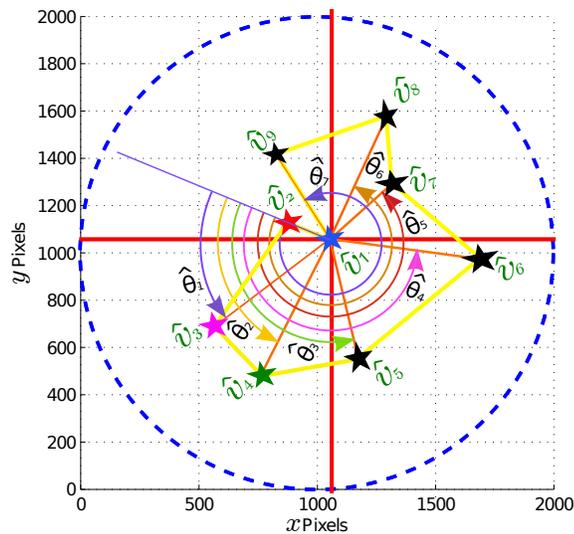


Figura 36: Orden de vértices en el plano de la imagen sintética.

Una vez creado el polígono en el plano de la imagen sintética se puede calcular su invariante. El procedimiento completo es descrito en el Algoritmo 2.

Algoritmo 2 CPIC(s, \hat{k}_{FOV}, n') función para crear un polígono y cálculo de la invariante.

- 1: INPUT: Imagen sintética \hat{s} , \hat{k}_{FOV} número de estrellas vecinas, n' número de vértices del polígono para calcular la invariante
 - 2: $\hat{n}' = nn(0) \hat{k}_{FOV} < \hat{n}'$
 - 3: $\hat{k} = \hat{k}_{FOV}$
 - 4: $\hat{k} = \hat{n}'$
 - 5: $\hat{v}_1 \leftarrow T P x y (T x y (\hat{s}_1)) + \varepsilon_{\hat{x}\hat{y}_1}$ {TPxy() transformar a xy pixeles }
 - 6: $\hat{n}s_1 \leftarrow T P x y (T x y (\hat{s}_2)) + \varepsilon_{\hat{x}\hat{y}_2}$
 - 7: $\hat{n}s_2 \leftarrow T P x y (T x y (\hat{s}_3)) + \varepsilon_{\hat{x}\hat{y}_3}$
 - 8: \vdots
 - 9: $\hat{n}s_{\hat{k}} \leftarrow T P x y (T x y (\hat{s}_{(\hat{k})})) + \varepsilon_{\hat{x}\hat{y}_{\hat{k}}}$
 - 10: { $\varepsilon_{\hat{x}\hat{y}}$ es el generador de ruido }
 - 11: $\{\hat{v}_2, \hat{v}_3, \dots, \hat{v}_{\hat{k}}\} \leftarrow \hat{k}\text{-NN}(\hat{n}s)$
 - 12: $\{\hat{v}_i$ es la estrella vecina i -ésima de \hat{v}_1 }
 - 13:
 - 14: **Sort**{ $\hat{v}_3, \hat{v}'_4, \dots, \hat{v}'_{\hat{k}}$ } { Ordenar por ángulo con respecto a \hat{v}_2 }
 - 15: $\hat{Z} = (\hat{v}_1, \hat{v}_2, \hat{v}_3, \dots, \hat{v}_{\hat{k}})$ { en donde $v_{3, \dots, \hat{k}} = v'_i \in \{v'_3, v'_4, \dots, v'_k\}$ }
 - 16: { con el ángulo más pequeño con respecto a $\overrightarrow{v_1 - v_2}$ }
 - 17: { \hat{Z} es un polígono de \hat{k} vértices }
 - 18: $w^{\hat{s}} = \varphi_{\ell}(\hat{Z})$ { $\ell = [1, 2, \dots, \hat{k}/2]$ } { $w^{\hat{s}}$ Invariante calculada }
-

4.4. Evaluación del algoritmo de identificación de estrellas

En las secciones previas se explica como calcular la invariante $w^{\hat{s}}$ del polígono asociado a la estrella \hat{s} en la imagen sintética y también como generar la base de datos de la invariante. En esta sección se validará el desempeño del algoritmo de identificación de estrellas y se presenta en detalle tres ejemplos para ilustrar el proceso de identificación.

4.4.1. Evaluando el desempeño del algoritmo de identificación de estrellas

El método de validación se realiza de la siguiente manera. Primero, se diseña un programa informático para generar la base de datos de invariantes como se describe en la sección 4.2. Para cada una de las estrellas $s \in S$ se introduce un nivel de ruido distinto, el polígono Z_s es creado y entonces la invariante asociada $w^s = \varphi_\ell(Z_s)$ es calculada. Este procedimiento es repetido 1 millón de veces para cada una de las estrellas en S . Por lo que se obtiene la tupla $(ID_s, k, P, w_{\max}^s(p), w_{\min}^s(p))$ las cuales son almacenadas para crear la base de datos de las invariantes completa.

En una etapa completamente separada se diseña un segundo programa de cómputo que solo tiene acceso a la base de datos de las invariantes. El procedimiento para evaluar al algoritmo de identificación de estrellas es conducido para cada una de las estrellas del catálogo. Durante la evaluación una imagen sintética es creada como se explica en la sección 4.3.1 la cual es rotada aleatoriamente por multiplicar cada uno de los elementos en la imagen por una matriz de rotación. Entonces el polígono \hat{Z} con $n_1 \in \{3, \dots, 9\}$ vértices es creado siguiendo las etapas indicadas en la subsección 4.3.2.

Después, se calcula la invariante $w^{\hat{s}} = \varphi_1(\hat{Z})$ del polígono de la imagen sintética. Entonces, se busca una coincidencia para $w^{\hat{s}}$ en la base de datos de la invariante correspondiente a n_1 vértices. En algunos casos una sola coincidencia es posible por lo que el proceso de identificación termina, sin embargo, en muchos casos varias colisiones ocurren como se va a mostrar más adelante, por lo que se necesita ejecutar de nuevo el algoritmo. Para tratar de encontrar una sola coincidencia en esta segunda iteración se modifica el número de vértices del polígono n_2 , en donde $n_2 \neq n_1$ y $n_2 \in \{3, \dots, 9\}$, mientras se mantiene la misma CS. Un segundo polígono es formado usando el mismo procedimiento descrito en la subsección 4.3.2. Por lo que una nueva invariante $w^{\hat{s}}$ es calculada, esta vez la búsqueda para una coincidencia en la base de datos de la invariante correspondiente a n_2 vértices no es conducida en la base de datos de la invariante completa, solo se hace en los polígonos en colisión que se obtuvieron anteriormente. En esta segunda iteración y de igual manera como la primera iteración se puede encontrar una sola coincidencia, pero en otros casos podemos seguir encontrando colisiones y consecuentemente el proceso es iterado m veces usando nuevos polígonos con diferentes número de vértices en cada iteración, teniendo que $n_m \neq n_1 \neq n_2, \dots, \neq n_{m-1}$, en donde $n_m \in \{3, \dots, 9\}$ y $m_{\max} = 7$. En principio es posible que aún puedan haber colisiones después de m iteraciones, en este caso el algoritmo no es capaz de identificar al polígono y consecuentemente ninguna de las estrella. El procedimiento se describe

en el algoritmo 3.

Algoritmo 3 búsqueda de polígono.

1: INPUT: Catálogo de estrellas S , n_m m secuencias del número de vértices en el polígono para calcular la invariante. **cada** $\hat{s} \in \hat{S}$
2: $(NB, \hat{k}_{FOV}) = \mathbf{Buscar}$ estrellas NB en el FOV para $\hat{CS}_{\hat{s}}$
3: $w^{\hat{s}} = CPIC(NB, \hat{k}_{FOV}, n_1)$
4: $ID_c(\text{number of collisions}) = \mathbf{Búsqueda}$ $w^{\hat{s}}$ en la base de datos completa número de colisiones encontrados > 0
5: $t = 2$ $t \leq m$
6: $w^{\hat{s}} = CPIC(NB, \hat{k}_{FOV}, n_t)$
7: $ID_c(\text{new number of collisions}) = \mathbf{Buscar}$ $w^{\hat{s}}$ en base de datos de invariantes
8: solamente estrellas $ID_c(\text{number of collisions})$ en colisiones nuevo número
de colisiones encontrados $== 0$
9: $w^{\hat{s}}$ **encontrar en basededatos de invariantes**
10: $t = m$
11: $t + +$
12:
13: $w^{\hat{s}}$ **encontrar en basededatos de invariantes**
14: $\{ID \text{ estrella encontrado, } ID \text{ estrellas en colisiones o no encontradas}\}$

4.4.2. Magnitud aparente ≤ 6.0

Para evaluar el comportamiento del algoritmo de identificación de estrellas, el procedimiento descrito en el algoritmo 3 se repite 10,000 veces para cada uno de los 5,024 objetos celestes del catálogo. Diferentes niveles de ruido es introducido cada vez que el algoritmo es ejecutado.

El resultado de la evaluación se muestra en la Tabla 13 y presenta el porcentaje de las colisiones después de cuatro iteraciones para ambos niveles de ruido introducidos; los cuales son ± 0.5 pixel y ± 1.0 pixel.

Tabla 13: Comportamiento del algoritmo de identificación de estrellas después de ejecutarlo 10,000 veces por cada una de las 5,024 estrellas del catálogo. La tabla presenta el porcentaje de colisiones para las primeras cuatro iteraciones. Se resaltan a los resultados correspondientes al número mayor de reconocimiento de polígonos.

n_1	Colisiones ruido de ± 0.5 pixel	Colisiones ruido de ± 1.0 pixel	n_2	Colisiones ruido de ± 0.5 pixel	Colisiones ruido de ± 1.0 pixel	n_3	Colisiones ruido de ± 0.5 pixel	Colisiones ruido de ± 1.0 pixel	n_4	Colisiones ruido de ± 0.5 pixel	Colisiones ruido de ± 1.0 pixel
3	58.284	0.000	4	99.973131	95.755	5	99.999560	99.872942	6	99.999168	99.999865
3	58.284	0.000	4	99.973131	95.755	5	99.999560	99.872942	7	99.999130	99.999885
3	58.284	0.000	4	99.973131	95.755	5	99.999560	99.872942	8	99.999150	99.979944
3	58.284	0.000	4	99.973131	95.755	5	99.999560	99.872942	9	99.999285	99.999863
3	58.284	0.000	4	99.973131	95.755	6	99.999532	99.958521	7	99.999080	99.999871
3	58.284	0.000	4	99.973131	95.755	6	99.999532	99.958521	8	99.999126	99.999883
3	58.284	0.000	4	99.973131	95.755	6	99.999532	99.958521	9	99.999275	99.999847
3	58.284	0.000	4	99.973131	95.755	7	99.999552	99.998881	8	99.999152	99.999857
3	58.284	0.000	4	99.973131	95.755	7	99.999552	99.998881	9	99.999285	99.999837
3	58.284	0.000	4	99.973131	95.755	8	99.999556	99.935098	9	99.999264	99.999889
3	58.284	0.000	4	99.973131	95.755	9	99.999640	99.900195			
3	58.284	0.000	5	99.999043	96.422	6	99.999112	99.917186	7	99.998802	99.999839
3	58.284	0.000	5	99.999043	96.422	6	99.999112	99.917186	8	99.998796	99.999871
3	58.284	0.000	5	99.999043	96.422	6	99.999112	99.917186	9	99.998861	99.999831
3	58.284	0.000	5	99.999043	96.422	7	99.999108	99.890880	8	99.998770	99.999883
3	58.284	0.000	5	99.999043	96.422	7	99.999108	99.890880	9	99.998863	99.999851
3	58.284	0.000	5	99.999043	96.422	8	99.999164	99.967604	9	99.998915	99.999843
3	58.284	0.000	5	99.999043	96.422	9	99.999281	99.933426			
3	58.284	0.000	6	99.999791	97.434	7	99.999120	99.901387	8	99.998746	99.999827
3	58.284	0.000	6	99.999791	97.434	7	99.999120	99.901387	9	99.998931	99.999829
3	58.284	0.000	6	99.999791	97.434	8	99.999150	99.971300	9	99.998859	99.999819
3	58.284	0.000	6	99.999791	97.434	9	99.999228	99.937297			
3	58.284	0.000	7	99.995474	96.331	8	99.999114	99.960008	9	99.998991	99.999807
3	58.284	0.000	7	99.995474	96.331	9	99.999210	99.980667			
3	58.284	0.000	8	99.999584	95.261	9	99.999271	99.847265			
3	58.284	0.000	9	99.999682	94.808						
4	78.707	44.160	5	99.997526	99.608	6	99.999220	99.969194	7	99.998684	99.978688
4	78.707	44.160	5	99.997526	99.608	6	99.999220	99.969194	8	99.998768	99.978666
4	78.707	44.160	5	99.997526	99.608	6	99.999220	99.969194	9	99.998861	99.999695
4	78.707	44.160	5	99.997526	99.608	7	99.999202	99.978513	8	99.998746	99.978551
4	78.707	44.160	5	99.997526	99.608	7	99.999202	99.978513	9	99.998790	99.999674
4	78.707	44.160	5	99.997526	99.608	8	99.999154	99.958748	9	99.998879	99.999707
4	78.707	44.160	5	99.997526	99.608	9	99.999279	99.999753			
4	78.707	44.160	6	99.919916	99.766	7	99.999226	99.977108	8	99.998857	99.976943
4	78.707	44.160	6	99.919916	99.766	7	99.999226	99.977108	9	99.998796	99.999672
4	78.707	44.160	6	99.919916	99.766	8	99.999158	99.976988	9	99.998836	99.999689
4	78.707	44.160	6	99.919916	99.766	9	99.999218	99.999713			
4	78.707	44.160	7	99.999598	99.936	8	99.999244	99.976990	9	99.999003	99.999682
4	78.707	44.160	7	99.999598	99.936	9	99.999254	99.999703			
4	78.707	44.160	8	99.999562	99.896	9	99.999327	99.999709			
4	78.707	44.160	9	99.999739	99.827						
5	88.144	47.419	6	99.999136	99.714	7	99.998861	99.954138	8	99.998340	99.953840
5	88.144	47.419	6	99.999136	99.714	7	99.998861	99.954138	9	99.998629	99.999691
5	88.144	47.419	6	99.999136	99.714	8	99.998668	99.954053	9	99.998923	99.999638
5	88.144	47.419	6	99.999136	99.714	9	99.998941	99.999680			
5	88.144	47.419	7	99.999168	99.749	8	99.998726	99.954057	9	99.998505	99.953844
5	88.144	47.419	7	99.999168	99.749	9	99.998804	99.999731			
5	88.144	47.419	8	99.999158	99.852	9	99.998967	99.999715			
5	88.144	47.419	9	99.999268	99.815						
6	87.835	46.157	7	99.986266	99.729	8	99.986164	99.935201	9	99.998509	99.999648
6	87.835	46.157	7	99.986266	99.729	9	99.998911	99.999735			
6	87.835	46.157	8	99.982904	99.797	9	99.998949	99.999721			
6	87.835	46.157	9	99.999363	99.890						
7	89.255	48.034	8	99.974769	99.687	9	99.998861	99.979793			
7	89.255	48.034	9	99.999309	99.912						
8	90.317	46.237	9	99.989459	99.635						
9	90.318	48.328									

El porcentaje mostrado en las tablas fueron calculados usando la siguiente ecuación

$$\frac{\# \text{ de estrellas en colision}}{5,024 \times 10,000} \times 100. \quad (39)$$

También se realizaron simulaciones con ruido de ± 2.0 pixeles de ruido, ± 3.0 pixeles de ruido y de ± 4.0 pixeles de ruido, se muestran los resultados en las Tablas 14 y 15 y se resaltan los mejores resultados.

Tabla 15: Comportamiento del algoritmo de identificación de estrellas después de ejecutarlo 10,000 veces por cada una de las 5,024 estrellas del catálogo. La tabla presenta el porcentaje de reconocimiento de polígonos para la sexta y séptima iteración con ruido de 3.0 píxeles y 4.0 píxeles. Se resaltan a los resultados correspondientes al número mayor de reconocimiento de polígonos.

n_1	n_2	n_3	n_4	n_5	n_6	Ruido 3.0	Ruido 4.0	n_7	Ruido 3.0	Ruido 4.0	n_8	Ruido 4.0
3	4	5	6	7	8	99.918033	95.137410	9	99.961176	97.400161	12	99.681530
3	4	5	6	7	9	99.933562	95.429339					
3	4	5	7	8	9	99.937904	93.404999					
3	4	6	7	8	9	99.944935	94.060538					
3	5	6	7	8	9	99.920508	94.436284					
4	5	6	7	8	9	99.932335	96.732849					

De los resultados presentados en la tabla 13, se puede observar que cuando se introduce un nivel de ruido de ± 0.5 pixel solo se necesitan dos iteraciones para identificar al 99.999 % de los polígonos creados usando las estrellas del catálogo como vértices. Para el caso en donde el ruido introducido es incrementado a ± 1.0 pixel se requiere de tres iteraciones para identificar al 99.999 % de los polígonos creados usando las estrellas del catálogo como vértices.

Si el nivel de ruido se incrementa a 2 píxeles el mejor porcentaje de reconocimientos de polígonos es de 99.9 % usando 4 iteraciones. Si el nivel de ruido se incrementa a 3 píxeles el mejor porcentaje de reconocimientos de polígonos es de 99.9 % usando 5 iteraciones y si éste se incrementa a 4 píxeles el mayor porcentaje de reconocimientos es de 99.68 % después de 8 iteraciones, tal como podemos observar en la Tablas 15 y 14.

Respecto al tiempo de ejecución después de correr al método propuesto 50.24 millones de veces con un nivel de ruido de 4 píxeles el tiempo medio para identificar un polígono es de 183.64 μs con $\sigma = 51.11 \mu s$. Estos resultados fueron obtenidos usando una computadora Alienware™ corriendo Xubuntu Linux 16.04 con un CPU Intel® Core™ i7-4700MQ a 2.40GHz. El algoritmo fue programado en el lenguaje C sin multithreading. El programa para generar las bases de datos de las invariantes fue programado en lenguaje C usando 8 multithreads. El tiempo para generar la base de datos es aproximadamente 70 horas de tiempo de procesamiento para generar la base de datos, esto para un nivel de ruido dado, usando el mismo hardware descrito anteriormente.

Es importante mencionar que incrementar el número de iteraciones mejora la proporción de reconocimiento, de ahí que hay que seleccionar el número de vértices n_1, n_2, \dots, n_m que presente el mejor comportamiento al reconocimiento de los polígonos. Por lo que cuando el nivel de ruido introducido es de ± 0.5 pixel, Las mejores secuencias de los números de vértices para formar los

polígonos son: 3-5, 3-6, 3-8, 3-9, 4-7, 4-8, 4-9, 5-6, 5-7, 5-8, 5-9, 6-9 y 7-9. Adicionalmente, cuando el nivel de ruido introducido es de ± 1.0 pixel, las mejores secuencias del número de vértices para generar los polígonos son: 4-5-9, 4-6-9, 4-7-9, 4-8-9, 5-6-9, 5-7-9, 5-8-9, 6-7-9 y 6-8-9. Se presenta la Tabla 16 con el porcentaje de colisiones y polígonos no reconocidos con ruido en la imagen los cuales se corren 10,000 veces para cada una de las 5024 estrellas del catálogo de estrellas.

Tabla 16: Porcentaje de colisiones y polígonos no reconocidos con ruido en la imagen (10,000 veces para cada una de las 5,024 estrellas del catálogo).

#	Ruido +/- 0.5 pixel						Ruido +/- 1.0 pixel					
	Colisiones			#Pol. no reconocidos			Colisiones			#Pol. no reconocidos		
<i>m</i>	min	promedio	max	min	promedio	max	min	promedio	max	min	promedio	max
1	9.68165	16.7343	41.71599	0.000000	0.00003	0.00005	51.67208	59.95224	100.0000	0.000000	0.000009	0.000018
2	0.00000	0.00841	0.079618	0.000018	0.00054	0.00086	0.063511	1.294058	5.192024	0.000060	0.000082	0.000117
3	0.00000	0.00036	0.012645	0.000360	0.00085	0.00133	0.000133	0.040307	0.152604	0.000078	0.000124	0.000167
4	0.00000	0.00000	0.000000	0.000715	0.00110	0.00166	0.000000	0.005701	0.045965	0.000111	0.000162	0.000225

Se realizó una prueba de repetibilidad para el caso de ± 1.0 pixel de ruido y con la secuencia del número de vértices para generar los polígonos de 6-7-9 y 5, este último número de vértices se utiliza como invariante de verificación, toda estas secuencias nos sirve para el reconocimiento de los 5024 polígonos que se pueden generar dentro del catálogo de estrellas y con 10,000 veces que se ejecuta el algoritmo para cada una de las estrellas del catálogo, la cual llamaremos simulación completa. Teniendo como resultado el 99.999684 % de promedio con una variación de +0.000071 %, -0.000112 %, para encontrar el valor promedio y su variación se ejecutó 17 veces la simulación completa.

En las Tablas 13 a 15 los cálculos de las invariantes siempre fueron realizados con el valor de $\ell = 1$. De acuerdo a los resultados de las experimentaciones que se hicieron con los valores de ℓ , se encontró que para los números de vértices impares de los polígonos la restricción de ℓ se amplia a $\ell = 1, \dots, \lfloor (n-1) \rfloor$ y para los números pares de vértices de los polígonos la restricción de ℓ si se apega a $\ell = 1, \dots, \lfloor (n-1)/2 \rfloor$ comentado anteriormente en la sección 4.1.1.

Se hicieron cambios al valor de ℓ para hacer los cálculos de las invariantes de los polígonos con número de vértices de 5, 7, 9, debido a que estos números de vértices de los polígonos nos proporcionan $(n-1)$ distintos ℓ con los que se pueden trabajar y poder hacer comparaciones con las simulaciones anteriores. Podemos ver los resultados en las Tablas 17, 18 y 19 en todos los casos el algoritmo fue corrido 10,000 veces a cada una de las estrellas del catálogo.

Tabla 17: La tabla presenta el porcentaje de polígonos reconocidos usando polígonos de 5 lados con nivel de ruido introducido de ± 1.0 .

ℓ_1	Polígonos reconocidos	ℓ_2	Polígonos reconocidos	ℓ_3	Polígonos reconocidos	ℓ_4	Polígonos reconocidos
1	47.362020 %	2	99.573272 %	3	99.661851 %	4	99.704357 %
1	47.362020 %	2	99.573272 %	4	99.625161 %		
1	47.362020 %	3	99.625360 %	4	99.678392 %		
1	47.362020 %	4	51.716988 %				
2	26.531592 %	3	31.736736 %	4	99.634031 %		
2	26.531592 %	4	99.548256 %				
3	0.000096 %	4	99.597564 %				
4	0.000034 %						

Tabla 18: La tabla presenta el porcentaje de polígonos reconocidos usando polígonos de 7 lados con nivel de ruido introducido de ± 1.0 .

ℓ_1	Polígonos reconocidos	ℓ_2	Polígonos reconocidos	ℓ_3	Polígonos reconocidos	ℓ_4	Polígonos reconocidos
1	48.006065 %	2	99.082932 %	3	99.942713 %	4	99.943760 %
1	48.006065 %	2	99.082932 %	3	99.942713 %	5	99.963788 %
1	48.006065 %	2	99.082932 %	3	99.942713 %	6	99.942936 %
1	48.006065 %	2	99.082932 %	4	99.940818 %	5	99.961994 %
1	48.006065 %	2	99.082932 %	4	99.767170 %	6	99.940693 %
1	48.006065 %	2	99.082932 %	5	99.767124 %	6	99.785868 %
1	48.006065 %	2	99.082932 %	6	99.228973 %		
1	48.006065 %	3	99.507395 %	4	99.585370 %	5	99.963750 %
1	48.006065 %	3	99.507395 %	4	99.585370 %	6	99.604729 %
1	48.006065 %	3	99.507395 %	5	99.963000 %	6	99.962970 %
1	48.006065 %	3	99.507395 %	6	99.546047 %		
1	48.006065 %	4	99.520832 %	5	99.961029 %	6	99.961043 %
1	48.006065 %	4	99.521113 %	6	99.576959 %		
1	48.006065 %	5	99.765888 %	6	99.784624 %		
1	48.006065 %	6	52.243125 %				
2	0.000098 %	3	98.998461 %	4	99.217158 %	5	99.674459 %
2	0.000098 %	3	98.998461 %	4	99.217158 %	6	99.943324 %
2	0.000098 %	3	98.998461 %	5	99.597060 %	6	99.963740 %
2	0.000098 %	3	98.998461 %	6	99.942910 %		
2	0.000098 %	4	99.134622 %	5	99.646194 %	6	99.961937 %
2	0.000098 %	4	99.134622 %	6	99.940818 %		
2	0.000098 %	5	31.918941 %	6	99.782988 %		
2	0.000098 %	6	99.033350 %				
3	0.000000 %	4	29.128073 %	5	99.556555 %	6	99.963607 %
3	0.000000 %	4	29.128073 %	6	99.595882 %		
3	0.000000 %	5	99.438075 %	6	99.963264 %		
3	0.000000 %	6	99.535579 %				
4	22.819676 %	5	99.513603 %	6	99.961131 %		
4	22.819676 %	6	99.567500 %				
5	0.000322 %	6	99.779988 %				
6	48.214110 %						

Tabla 19: La tabla presenta el porcentaje de polígonos reconocidos usando polígonos de 9 lados con nivel de ruido introducido de ± 1.0 .

ℓ_1	Polígonos reconocidos	ℓ_2	Polígonos reconocidos	ℓ_3	Polígonos reconocidos	ℓ_4	Polígonos reconocidos
1	48.291877 %	2	99.918903 %	3	99.999699 %	4	99.999719 %
1	48.291877 %	2	99.918903 %	3	99.999699 %	5	99.999682 %
1	48.291877 %	2	99.918903 %	3	99.999699 %	6	99.999739 %
1	48.291877 %	2	99.918903 %	3	99.999699 %	7	99.999733 %
1	48.291877 %	2	99.918903 %	3	99.999699 %	8	99.999703 %
1	48.291877 %	2	99.918903 %	4	99.999743 %	5	99.999664 %

Continúa en la siguiente página

Tabla 19 – Continuación de la página anterior

ℓ_1	Polígonos reconocidos	ℓ_2	Polígonos reconocidos	ℓ_3	Polígonos reconocidos	ℓ_4	Polígonos reconocidos
1	48.291877 %	2	99.918903 %	4	99.999743 %	6	99.999699 %
1	48.291877 %	2	99.918903 %	4	99.999743 %	7	99.999729 %
1	48.291877 %	2	99.918903 %	4	99.999743 %	8	99.999715 %
1	48.291877 %	2	99.918903 %	5	99.999735 %	6	99.999709 %
1	48.291877 %	2	99.918903 %	5	99.999735 %	7	99.999713 %
1	48.291877 %	2	99.918903 %	5	99.999735 %	8	99.999699 %
1	48.291877 %	2	99.918903 %	6	99.999715 %	7	99.999701 %
1	48.291877 %	2	99.918903 %	6	99.999715 %	8	99.999703 %
1	48.291877 %	2	99.918903 %	7	99.931539 %	8	99.943268 %
1	48.291877 %	2	99.918903 %	8	99.931121 %		
1	48.291877 %	3	99.793000 %	4	99.999670 %	5	99.999731 %
1	48.291877 %	3	99.793000 %	4	99.999670 %	6	99.999652 %
1	48.291877 %	3	99.793000 %	4	99.999670 %	7	99.999660 %
1	48.291877 %	3	99.793000 %	4	99.999670 %	8	99.999676 %
1	48.291877 %	3	99.793000 %	5	99.999707 %	6	99.999678 %
1	48.291877 %	3	99.793000 %	5	99.999707 %	7	99.999682 %
1	48.291877 %	3	99.793000 %	5	99.999707 %	8	99.999680 %
1	48.291877 %	3	99.793000 %	6	99.799072 %	7	99.999721 %
1	48.291877 %	3	99.793000 %	6	99.799072 %	8	99.821471 %
1	48.291877 %	3	99.793000 %	7	99.999743 %	8	99.999741 %
1	48.291877 %	3	99.793000 %	8	99.821212 %		
1	48.291877 %	4	99.885633 %	5	99.892677 %	6	99.999747 %
1	48.291877 %	4	99.885633 %	5	99.892677 %	7	99.999715 %
1	48.291877 %	4	99.885633 %	5	99.892677 %	8	99.893364 %
1	48.291877 %	4	99.885633 %	6	99.999711 %	7	99.999705 %
1	48.291815 %	4	99.885633 %	6	99.999711 %	8	99.999680 %
1	48.291815 %	4	99.885633 %	7	99.999729 %	8	99.999701 %
1	48.291815 %	4	99.885633 %	8	99.891982 %		
1	48.291815 %	5	99.784554 %	6	99.999721 %	7	99.999680 %
1	48.291815 %	5	99.784554 %	6	99.999721 %	8	99.999686 %
1	48.291815 %	5	99.784554 %	7	99.999749 %	8	99.999717 %
1	48.291815 %	5	99.784554 %	8	99.789863 %		
1	48.291815 %	6	99.617289 %	7	99.999715 %	8	99.999715 %
1	48.291815 %	6	99.617289 %	8	99.661117 %		
1	48.291815 %	7	99.750183 %	8	99.778830 %		
1	48.291815 %	8	54.795253 %				
2	0.001965 %	3	99.763535 %	4	99.999755 %	5	99.999723 %
2	0.001965 %	3	99.763535 %	4	99.999755 %	6	99.999695 %
2	0.001965 %	3	99.763535 %	4	99.999755 %	7	99.999664 %
2	0.001965 %	3	99.763535 %	4	99.999755 %	8	99.999721 %
2	0.001965 %	3	99.763535 %	5	99.999729 %	6	99.999731 %
2	0.001965 %	3	99.763535 %	5	99.999729 %	7	99.999711 %
2	0.001965 %	3	99.763535 %	5	99.999729 %	8	99.999741 %
2	0.001965 %	3	99.763535 %	6	99.794554 %	7	99.866379 %
2	0.001965 %	3	99.763535 %	6	99.794554 %	8	99.999709 %
2	0.001965 %	3	99.763535 %	7	99.857576 %	8	99.999711 %
2	0.001965 %	3	99.763535 %	8	99.999743 %		
2	0.001965 %	4	99.632004 %	5	99.747753 %	6	99.999731 %
2	0.001965 %	4	99.632004 %	5	99.747753 %	7	99.892671 %
2	0.001965 %	4	99.632004 %	5	99.747753 %	8	99.999763 %
2	0.001951 %	4	99.632004 %	6	99.999733 %	7	99.999713 %
2	0.001965 %	4	99.632004 %	6	99.999733 %	8	99.999682 %
2	0.001965 %	4	99.632004 %	7	99.818983 %	8	99.999711 %
2	0.001965 %	4	99.632004 %	8	99.999733 %		
2	0.001965 %	5	99.569297 %	6	99.999723 %	7	99.999729 %
2	0.001965 %	5	99.569297 %	6	99.999723 %	8	99.999670 %
2	0.001965 %	5	99.569297 %	7	99.761164 %	8	99.999737 %
2	0.001965 %	5	99.569297 %	8	99.999767 %		
2	0.001965 %	6	99.698029 %	7	99.809279 %	8	99.999689 %
2	0.001965 %	6	99.698029 %	8	99.999697 %		
2	0.001965 %	7	32.519174 %	8	99.922605 %		
2	0.001965 %	8	99.909228 %				
3	14.064934 %	4	99.719829 %	5	99.817303 %	6	99.854887 %
3	14.064934 %	4	99.719829 %	5	99.817303 %	7	99.999640 %

Continúa en la siguiente página

Tabla 19 – Continuación de la página anterior

ℓ_1	Polígonos reconocidos	ℓ_2	Polígonos reconocidos	ℓ_3	Polígonos reconocidos	ℓ_4	Polígonos reconocidos
3	14.064934 %	4	99.719829 %	5	99.817303 %	8	99.999691 %
3	14.064934 %	4	99.719829 %	6	99.757940 %	7	99.999632 %
3	14.064934 %	4	99.719829 %	6	99.757940 %	8	99.999719 %
3	14.064934 %	4	99.719829 %	7	99.999626 %	8	99.999707 %
3	14.064934 %	4	99.719829 %	8	99.999719 %		
3	14.064934 %	5	99.501955 %	6	99.607556 %	7	99.999431 %
3	14.064934 %	5	99.501955 %	6	99.607594 %	8	99.999689 %
3	14.064934 %	5	99.501955 %	7	99.999425 %	8	99.999705 %
3	14.064934 %	5	99.501955 %	8	99.999711 %		
3	14.064934 %	6	25.179682 %	7	99.234956 %	8	99.999691 %
3	14.064934 %	6	25.179682 %	8	99.820758 %		
3	14.064934 %	7	99.051748 %	8	99.999723 %		
3	14.064934 %	8	99.820261 %				
4	9.899331 %	5	18.073059 %	6	99.701634 %	7	99.999624 %
4	9.899331 %	5	18.073059 %	6	99.701650 %	8	99.999684 %
4	9.899331 %	5	18.073059 %	7	98.562412 %	8	99.999729 %
4	9.899331 %	5	18.073059 %	8	99.870545 %		
4	9.899331 %	6	99.587215 %	7	99.999636 %	8	99.999693 %
4	9.899331 %	6	99.587215 %	8	99.999729 %		
4	9.899331 %	7	98.258141 %	8	99.999727 %		
4	9.899331 %	8	99.849592 %				
5	0.000000 %	6	99.371622 %	7	99.999423 %	8	99.999695 %
5	0.000000 %	6	99.371622 %	8	99.999765 %		
5	0.000000 %	7	98.163547 %	8	99.999729 %		
5	0.000000 %	8	99.744275 %				
6	0.000000 %	7	99.079992 %	8	99.999715 %		
6	0.000000 %	8	99.624701 %				
7	0.000000 %	8	99.757956 %				
8	0.147448 %						

Si se comparan las Tablas 13 con ℓ fijo a 1 versus las Tablas 17, 18 y 19, los cuales cambian el valor de ℓ para cada iteración del algoritmo, podemos observar que los mejores resultados para tener el mayor reconocimiento de polígonos con ruido de ± 1 pixel es con las secuencias de números de vértices para 3 iteraciones tal como podemos observar en los resultados marcados con colores morado claro de la Tabla 13 y verde claro de la Tabla 19.

Tabla 20: Comportamiento del algoritmo de identificación de estrellas después de ejecutarlo 10,000 veces por cada una de las 5,024 estrellas del catálogo. La tabla presenta el porcentaje de reconocimiento de polígonos para la séptima iteración con ruido de 4.0 píxeles. Para esta prueba solo se cambia el valor de ℓ para polígonos de 9 vértices.

ℓ_1	ℓ_2	ℓ_3	ℓ_4	ℓ_5	ℓ_6	ℓ_7	Reconoce polígonos Ruido 4.0
1	2	3	4	5	6	7	50.153258 %
1	2	3	4	5	6	8	45.033564 %
1	2	3	4	5	7	8	53.582536 %
1	2	3	4	6	7	8	54.811150 %
1	2	3	5	6	7	8	55.035401 %
1	2	4	5	6	7	8	40.801638 %
1	3	4	5	6	7	8	48.702016 %
2	3	4	5	6	7	8	38.315374 %

Comparando el resultado del algoritmo con 7 iteraciones y cambiando el número de vértices para formar al polígono con ruido de 4 píxeles tal como podemos ver en la Tabla 15 versus el resultado de la Tabla 20 que para 7 iteraciones se cambia el valor de ℓ para polígonos de 9 vértices, esto es 97.4 % versus 55.035401 %, podemos concluir que el mejor comportamiento del algoritmo es cuando cambiamos el número de vértices en vez de cambiar ℓ .

4.4.2.1. Magnitud aparente ≤ 5.5

Si la magnitud aparente usada es ≤ 5.5 la ecuación utilizada es:

$$\frac{\# \text{ of polygons recognized}}{2,841 \times 10,000} \times 100. \quad (40)$$

Tabla 21: Comportamiento del algoritmo de identificación de estrellas después de ejecutarlo 10,000 veces por cada una de las 2,841 estrella del catálogo.

n_1	n_2	Ruido 1 pixel	n_3	Ruido 1 pixel	Ruido 2 pixeles	n_4	Ruido 2 pixeles	n_5	Ruido 3 pixeles	n_6	Ruido 3 pixeles	Ruido 4 pixeles	n_7	Ruido 4 pixeles
3	4	99.7144	5	99.9991	99.3434	6	99.8015	7	99.6146	8	99.7122	98.0938	9	99.4425
3	4	99.7144	5	99.9991	99.3434	6	99.8015	7	99.6146	8	99.7300	97.0752		
3	4	99.7144	5	99.9991	99.3434	6	99.8015	8	99.6608	9	99.7486	97.5468		
3	4	99.7144	5	99.9991	99.3434	6	99.8015	9	99.6335					
3	4	99.7144	5	99.9991	99.3434	7	99.8579	8	99.6743	9	99.7511	98.7526		
3	4	99.7144	5	99.9991	99.3434	7	99.8579	9	99.6776					
3	4	99.7144	5	99.9991	99.3434	8	99.8579	9	99.6279					
3	4	99.7144	5	99.9991	99.3434	9	99.8364							
3	4	99.7144	6	99.9991	99.4306	7	99.8507	8	99.6059	9	99.7486	98.3396		
3	4	99.7144	6	99.9991	99.4306	7	99.8507	9	99.6599					
3	4	99.7144	6	99.9991	99.4306	8	99.8520	9	99.7376					
3	4	99.7144	6	99.9991	99.4306	9	99.8138							
3	4	99.7144	7	99.9990	99.8123	8	99.8516	9	99.6625					
3	4	99.7144	7	99.9990	99.8123	9	99.8529							
3	4	99.7144	8	99.9989	99.6939	9	99.8528							
3	4	99.7144	9	99.9990	99.6951									
3	5	99.7910	6	99.9989	99.5022	7	99.8542	8	99.6093	9	99.7236	98.9106		
3	5	99.7910	6	99.9989	99.5022	7	99.8542	9	99.5707					
3	5	99.7910	6	99.9989	99.5022	8	99.8540	9	99.5727					
3	5	99.7910	6	99.9989	99.5022	9	99.8263							
3	5	99.7910	7	99.9989	99.8538	8	99.8536	9	99.7225					
3	5	99.7910	7	99.9989	99.8538	9	99.8552							
3	5	99.7910	8	99.9989	99.7785	9	99.8540							
3	5	99.7910	9	99.9989	99.7632									
3	6	99.8159	7	99.9989	99.7209	8	99.8230	9	99.5743					
3	6	99.8159	7	99.9989	99.7209	9	99.8447							
3	6	99.8159	8	99.9988	99.7061	9	99.8448							
3	6	99.8159	9	99.9989	99.4241									
3	7	99.9302	8	99.9989	99.7899	9	99.8450							
3	7	99.9302	9	99.9989	99.7450									
3	8	99.9598	9	99.9989	99.7274									
3	9	99.9631												
5	7	99.9088	8	99.9110	99.5888	9	99.6041							
5	7	99.9088	9	99.9112	99.6042									

Para un nivel de ruido de 1 pixel, el porcentaje de reconocimiento de polígonos es de 99.9 % después de realizar dos iteraciones. Si el nivel de ruido se incrementa a 2 píxeles el porcentaje de 99.9 % se obtienen después de tres iteraciones. Si el nivel de ruido se incrementa a 3 píxeles el porcentaje de reconocimiento máximo es de 99.9 % después de 5 iteraciones y finalmente si el nivel de ruido se incrementa a 4 píxeles el porcentaje de reconocimiento de polígonos máximo es de 99.4 % después de 7 iteraciones. El tiempo de procesamiento medio del algoritmo para un nivel

de ruido de 4 píxeles es de $64.21 \mu\text{s}$ con $\sigma = 34.12 \mu\text{s}$, para el nivel de ruido de 1 píxel el tiempo de procesamiento medio es de $32.37 \mu\text{s}$ con $\sigma = 19.81 \mu\text{s}$.

4.4.3. Validación del polígono reconocido

Cabe resaltar que inicialmente no existía un procedimiento que comprobara que el polígono encontrado fuera o no fuera válido, por lo que el programa podía habernos entregado un polígono encontrado que fuera falso. Eso quiere decir que si se reconoce un polígono no podemos asegurar que ese polígono sea el mismo que corresponde a la estrella en estudio, por lo que a continuación se introduce un mecanismo de verificación. Este sistema de comprobación es realizado con las iteraciones con números de vértices distintos que no hayan sido usados anteriormente para generar al polígono con el fin de verificar que el polígono es válido. Para el caso de encontrar una coincidencia después de $M \leq m$ iteraciones, se puede comprobar que el algoritmo propuesto ha encontrado un polígono válido por crear nuevos polígonos, usando a la misma estrella central y evaluando los $m_{\text{max}} - M$ invariantes remanentes y confirmar que todos los valores de las invariantes corresponden al polígono encontrado. El procedimiento descrito se muestra en el Algoritmo 4. Para cualquier situación del algoritmo es mejor tener el resultado de un polígono no reconocido que un polígono falso.

Algoritmo 4 Reconocimiento de polígonos.

- 1: INPUT: Catálogo de estrella S , n_m m secuencias del número de vértices en el polígono para calcular la invariante.
Cada $\hat{s} \in \hat{S}$
 - 2: $(\text{NB}, \hat{k}_{\text{FOV}}) = \text{Buscar NB estrellas en el FOV para } \hat{CS}_{\hat{s}}$
 - 3: $w^{\hat{s}} = \text{CPIC}(\text{NB}, \hat{k}_{\text{FOV}}, n_1)$
 - 4: $ID_c(\text{number of collisions}) = \text{Buscar } w^{\hat{s}}$ en la base de datos completa de la invariante número de colisiones reconocidos > 0
 - 5: $t = 2$ $t \leq m$
 - 6: $w^{\hat{s}} = \text{CPIC}(\text{NB}, \hat{k}_{\text{FOV}}, n_t)$
 - 7: $ID_c(\text{new number of collisions}) = \text{Buscar } w^{\hat{s}}$ en base de datos
solo estrellas $ID_c(\text{number of collisions})$ en colisiones nuevo número de colisiones reconocidos $== 0$ todas $w^{\hat{s}}$ de las m iteraciones están en su correspondiente base de datos de invariante
 - 8: *ID estrella encontrada*
 - 9: *ID estrella no encontrada*
 - 10: $t = m$
 - 11: $t++$
 - 12: todas las $w^{\hat{s}}$ de m iteraciones están en su base de datos de invariante correspondiente
 - 13: *ID estrella encontrada*
 - 14: *ID estrella no encontrada* número de colisiones encontradas > 0
 - 15: *ID estrellas en colisión*
 - 16: $\{\text{Output: ID estrella encontrada, ID estrellas en colisión o estrella no encontrada}\}$
-

En ciertos casos existen objetos en el FOV que pueden ser confundidos por un sensor de estrellas como un cuerpo celeste que no son partes del catálogo, tales como los planetas satélites

artificiales, basura espacial y lo más común, estrellas no incluidas en el catálogo Mortari *et al.* (2004) los cuales llamaremos estrellas falsas. El mal funcionamiento de los sensores CMOS tal como los hot pixels o los hoy spots de los CCD's Jørgensen (2000) también son fuentes de estrellas falsas, otros ejemplos son las reflexiones en la lente debido a los cuerpos celestes brillantes Steyn *et al.* (1997). Los sensores de estrellas también pueden ser afectadas por la desaparición de estrellas válidas debido a los efectos de las estrellas variables en el cual su magnitud toma valores abajo del umbral de la detección del sensor y también por los algoritmos de filtrado de la imagen.

En las simulaciones, el algoritmo propuesto verifica que si un polígono coincide, éste siempre corresponde a la estrella bajo análisis, esto es, no existen reconocimiento de polígonos falsos. Los sistemas de identificación de estrellas reales experimentan situaciones en donde estrellas que no existen en el catálogo de estrellas aparecen en la imagen o algunas estrellas del catálogo desaparecen de la imagen. En este caso el algoritmo propuesto tiene un comportamiento robusto tal como se puede ver en la Tabla 22. El método en la forma actual no puede identificar a todos los polígonos cuando las estrellas aparecen o desaparecen de la imagen, pero tampoco identifica a un polígono que no es de la estrella bajo estudio. Todos los polígonos identificados son polígonos válidos debido a que en la base de datos de invariantes si existen casos en que una estrella perteneciente al polígono cambia de posición con otra estrella que no está en el polígono, esto es debido al nivel de ruido introducido y de esa forma tendría el mismo comportamiento cuando la estrella desaparece de la imagen.

Tabla 22: Rendimiento del algoritmo de identificación de estrellas después de ejecutarlo 10,000 veces a cada una de las 5,024 estrellas del catálogo cuando se pierde una o dos estrellas o si una estrella que no se encuentra en el catálogo es introducida en la imagen.

	desaparece 1 estrella	desaparecen 2 estrellas	1 estrella falsa es introducida
Colisiones	0.000 %	0.000 %	0.000 %
Polígonos falsos	0.000 %	0.000 %	0.000 %
Polígonos reconocidos	0.878 %	0.005 %	0.000 %
Polígonos no reconocidos	99.122 %	99.995 %	100.000 %

Los Rayos Cósmicos o impactos de iones pesados pueden provocar eventos tal como los SEU (Single event upsets), los cuales pueden interrumpir la operación de la electrónica sensible. En este caso un efecto de radiación temporal se observa en el sensor de estrellas. Cuando las partículas pegan al CCD éstos generan electrones, los cuales cargan los pixeles como los fotones lo hacen regularmente, produciendo una imagen tal como lo haría una estrella brillante Brekke *et al.* (2004), este efecto se puede observar en la Figura 37 . En el caso de que sucedan

uno o muchos SEU's y provoquen estrellas falsas como vértices dentro del polígono, el método propuesto no reconocerá al polígono.

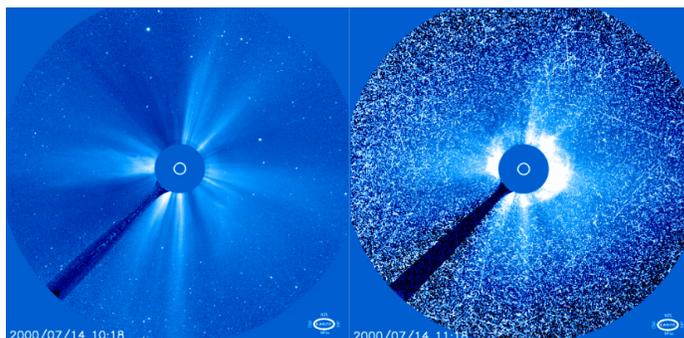


Figura 37: Imagen normal. Efecto de la radiación en el CCD. (ESA, 2008)

4.4.4. Comparación con otro método

En esta subsección se compara nuestro método con una técnica de identificación de estrellas desarrollada por Wei *et al.* (2009), su proposición está basada en la Transformación Log Polar (LPT acrónimo de las palabras en inglés). En ese desarrollo cada punto de interés en el plano cartesiano xy es transformado a un punto en el plano log-polar y éste es invariante a la transformación de semejanza. En el método que se propone en esta tesis todos los puntos de interés del plano xy son mapeados a un solo punto en el plano complejo y éste también es invariante a la transformación de semejanza. Para este comparativo con respecto al método LPT, en vez de usar un sensor de imagen con resolución de 2048×2048 se utiliza una resolución de 1024×1024 utilizando el mismo FOV de 12.4° . La Tabla 23 presenta los resultados de la comparación para ambos métodos para estrellas con una magnitud aparente ≤ 6.0 . Tal como podemos ver de los resultados, la propuesta presentada en esta tesis presenta un ligero mejor desempeño que el método de Wei *et al.* aún para niveles de ruido de hasta $\sigma = 2.0$ píxeles.

Tabla 23: Comparación entre método propuesto y método de Wei *et al.*'s Wei *et al.* (2009).

Ruido σ	Método propuesto		Método de Wei <i>et al.</i> 's		
	Ruido promedio en píxeles	Tiempo promedio (ms)	Porcentaje de identificación	Tiempo promedio (ms)	Porcentaje de identificación
0.0		0.044 ¹	100.00 %	15.7 ²	100.00 %
0.4		0.051 ¹	100.00 %	15.7 ²	99.90 %
0.8		0.067 ¹	99.99 %	15.7 ²	99.60 %
1.6		0.098 ¹	99.76 %	15.7 ²	98.90 %
2.0		0.118 ¹	98.86 %	15.7 ²	98.20 %
2.0		4.540 ³	98.86 %	15.7 ²	98.20 %

¹ Intel i7-4700MQ 2.4 GHz. ² Pentium III 800 MHz. ³ ARM iMX.6 396MHz.

La Tabla 24 presenta una comparación de tiempo de identificación de una estrella entre el método propuesto y otros algoritmos que han sido publicados.

Tabla 24: Comparación de algoritmos de identificación de estrellas.

Método	CPU	Freq.	Numero de estrellas en catálogo	Magnitud Aparente	Tiempo requerido para identificar estrellas(ms)	Referencia
Algoritmo Grid Modificado FOV 6°	Pentium 4	3.06 GHz	9110	≤ 8.0	19.100	Lee y Bang (2007)
Algoritmo Grid Modificado FOV 8°					51.100	
Algoritmo Grid Modificado FOV 10°					105.400	
Algoritmo Grid Modificado FOV 12°					142.900	
Identificación estrellas Piramidal Pirámide	Pentium III	450 MHz	3694	≤ 5.8	1000.000	Mortari <i>et al.</i> (2004)
M. Grid	Pentium (R) Dual-Core	3.20 GHz	6685	≤ 6.5	27.500	Luo <i>et al.</i> (2015)
LPT					394.600	
					73.800	
Algoritmo patrón vector uni-dimensional					7.800	
Liebe					13.900	
Votación geométrico		2.99 GHz	1631	≤ 5.0	73.330	Pham <i>et al.</i> (2013)
Método piramidal					24.410	
Distancias plana con búsqueda de árbol					4.850	
Transformación K-L					12.500	
Triángulo	Pentium Dual-Core	3.2 GHz	2859	≤ 5.5	18.800	Zhao <i>et al.</i> (2016)
SVD					23.600	
Log Polar	Pentium III	800 MHz		≤ 6.0	15.700	Wei <i>et al.</i> (2009)
Grid					10.500	
Nuestro método	Intel i7-4700MQ	2.4 GHz	2841	≤ 5.5	0.064	
	ARM i.MX6	1GHz			0.900	
	ARM i.MX6	396 MHz			1.900	

4.4.5. Casos de estudio

Se presentan tres ejemplos como comprobación de los procedimientos, en el primer ejemplo se hace sobre la zona del cielo con mayor número de estrellas, a la base de datos y a las estrellas no se les introduce ruido. El segundo ejemplo es en la zona de menor número de estrellas y al igual que el primer ejemplo no tiene ruido introducido, y en el último ejemplo para una zona aleatoria del cielo, tanto a la base de datos como a las estrellas dentro del campo de visión si tienen ruido introducido.

4.4.5.1. Caso de estudio usando la estrella CS ID 20155 sin ruido

Para este ejemplo se utiliza a la CS ID 20155 como estrella central, el cual tiene 41 estrellas dentro del campo de visión y se usan polígonos de 9 vértices, las estrellas vecinas que forman a este polígono son:

Tabla 25: Coordenadas esféricas de CS ID 20155 y de sus estrellas vecinas.

star ID	asc. rad.	dec. rad.
20155	1.133562	0.272754
20211	1.137104	0.263465
20680	1.163289	0.256804
20661	1.162152	0.272590
20433	1.149376	0.292818
20490	1.152336	0.304458
20405	1.147270	0.306175
20169	1.134297	0.244960
20350	1.143446	0.245693

Se proyecta a cada una de las estrellas al plano xy utilizando las ecuaciones 32 y 33 se calcula la invariante w_s para generar la base de datos, el valor de la invariante w_s para este polígono en particular es el número complejo $0.491681 - j0.027945$, siendo que $j = \sqrt{-1}$ es la unidad imaginaria, el valor de w_s se almacena en la base de datos de la invariante. Se muestran algunos valores de invariantes calculados y almacenados en la base de datos que se encuentran ordenados por la parte real y después por la parte imaginaria tal como se muestra en la tabla 26.

Tabla 26: Parte de las invariantes almacenadas en la base de datos de invariantes.

star ID	# estrellas NB	Vértices Polígono	Parte Real Invariante	Parte Imag. Invariante
97774	21	9	0.4691435384	-0.0215473926
111147	11	9	0.4772969671	0.1829314917
24516	21	9	0.4875978155	-0.7584331867
112826	12	9	0.4884443618	0.1793643645
97335	9	9	0.4891462132	-0.0257052012
20155	41	9	0.491680794	-0.0279451468
107213	13	9	0.4951482314	0.0772224181
76616	15	9	0.501877282	-0.155872621
40242	48	9	0.5160926526	-0.9628223352
61834	46	9	0.5163644043	0.5370181319
1977	20	9	0.519964102	-0.6127294464

La siguiente etapa es crear la imagen sintética con las ecuaciones 36 y 37.

Tabla 27: Centroides en la imagen sintética para CS ID 20155.

star ID	pixel \hat{x}	pixel \hat{y}
20155	1000.000000	1000.000000
20211	928.762123	943.212931
20680	944.902039	702.788299
20661	1079.280751	759.196768
20433	1219.513091	925.905736
20490	1329.458848	935.436911
20405	1330.242697	982.666266
20169	759.161219	913.234108
20350	791.389757	837.821630

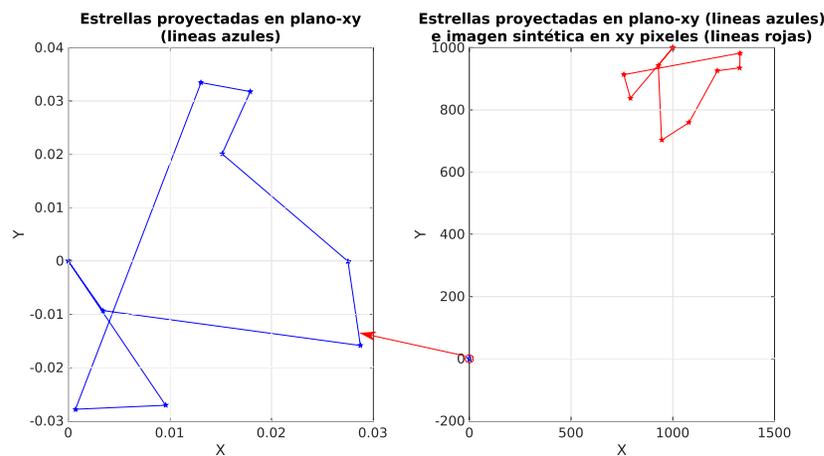


Figura 38: a) Estrellas proyectadas en el plano xy amplificadas de la figura de la derecha, b) Estrellas proyectadas en el plano xy y los centroides $\hat{x}\hat{y}$ de la imagen sintética para la estrella ID 20155 como estrella central.

En la Figura 38-b se muestran dos polígonos para la estrella con CS ID 20155, el punto pequeño es el polígono de las estrellas proyectadas en el plano xy y el otro polígono es el formado por las estrellas en la imagen sintética. La Figura 38-a es el polígono de estrellas proyectadas en el plano xy amplificada de la Figura 38-b. Como se puede observar el polígono de las estrellas

proyectadas en el plano xy tiene diferente escala que la del polígono de la imagen sintética, los orígenes de cada uno son distintos y también están rotadas con ángulos distintos, aunque ambos polígonos tienen la misma forma.

La invariante calculada w_s de la imagen sintética tiene el valor del número complejo $0.491681 - j0.027945$. Si buscamos la parte real 0.491681 en la tabla 26 de la base de datos de las invariantes, se encuentra una sola estrella asociada con el mismo valor de la parte real w_s , la star ID de esta estrella es:

la CS ID 20155.

Con lo que podemos concluir que es el mismo polígono asociado con la CS ID 20155 que es la estrella central que se planteó al inicio de este ejemplo.

4.4.5.2. Caso de estudio usando la estrella con ID 4842 sin ruido

Para este segundo ejemplo se utiliza a la CS ID 4842 y al igual que el ejemplo anterior se crean polígonos de 9 lados, aunque para esta estrella central solo tiene dos estrellas vecinas por lo que se va a crear un polígono de 3 vértices solamente, las estrellas vecinas dentro del mismo campo de visión son:

Tabla 28: Las coordenadas esféricas para la estrella CS ID 4842.

star ID	asc. rad.	dec. rad.
4842	0.272448	-0.550686
4567	0.255717	-0.512384
6488	0.364407	-0.540103

Proyectando a cada estrella al plano xy utilizando las ecuaciones anteriores 32 y 33 se calcula la invariante w_s para generar la base de datos, el valor de la invariante w_s para este polígono en particular es el número complejo $-1.678355 + j1.196784$. Esta invariante w_s es almacenada en la base datos de invariantes. A continuación se presenta otra parte de la base de datos en la Tabla 29.

Tabla 29: Otra parte de la base de datos de invariantes.

Star ID	# Estrellas NB	Vértices Polígono	parte Real Invariante	parte Imag. Invariante
17485	23	9	-7.7904800773	4.8180298668
60554	17	9	-2.3235576441	1.0411141556
4842	3	3	-1.6783549167	1.19678408
62305	7	7	-1.5498905799	-0.0132164407
74948	23	9	-1.5231251797	-0.29382909
92870	10	9	-1.5132626444	0.6990981069
71533	14	9	-1.4542293583	-0.5209098948
7984	5	5	-1.398465376	-0.1609823187
36672	42	9	-1.3978902376	-0.0786773247
84720	12	9	-1.2091218089	-0.1543426652
20304	18	9	-1.1544434065	-0.0425780132

Se crea una imagen sintética con los puntos calculados con las ecuaciones 36 y 37.

Tabla 30: Centroides en la imagen sintética para la estrella CS ID 4842.

star ID	pixel \hat{x}	pixel \hat{y}
4842	1000.000000	1000.000000
4567	705.241613	765.034200
6488	702.680778	1668.548921

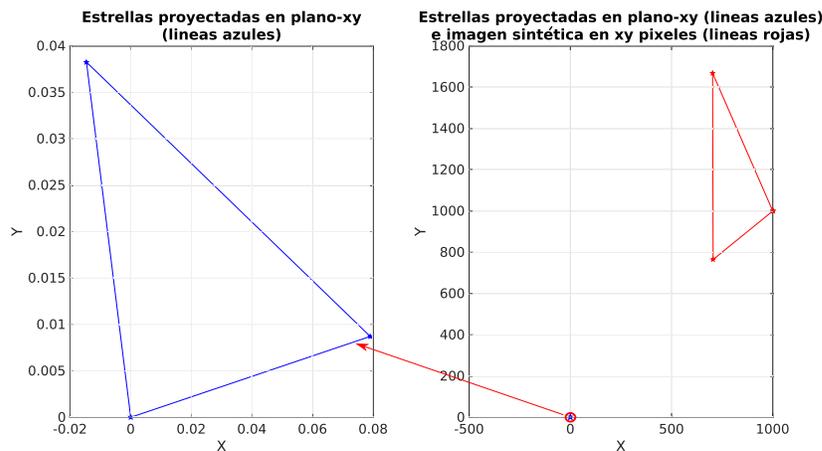


Figura 39: a) Estrellas proyectadas en el plano xy amplificadas de la figura de la derecha, b) Estrellas proyectadas en el plano xy y los centroides $\hat{x}\hat{y}$ de la imagen sintética para la estrella ID 4842 como estrella central.

En la Figura 39-b se muestran dos polígonos para la estrella CS ID 4842 como la estrella central, el punto pequeño es el polígono de las estrellas proyectadas en el plano xy y el otro polígono es el formado por las estrellas en la imagen sintética. La Figura 39-a es el polígono de estrellas proyectadas en el plano xy amplificadas de la Figura 39-b, El polígono de las estrellas proyectadas en el plano xy tiene diferente escala que la del polígono de la imagen sintética, los orígenes de cada uno son distintos y también están rotadas con ángulos diferentes aunque ambos polígonos tienen la misma forma.

La invariante calculada $w_{\hat{s}}$ es el número complejo $-1.678355 + j1.196784$. Si se busca la parte real -1.678355 en la base de datos de la invariante de la tabla 29, encontramos a una sola estrella con la misma parte real de la invariante $w_{\hat{s}}$ en la base de datos de las invariantes y ésta es la estrella CS ID 4842. Con lo que podemos concluir que es el mismo polígono asociado con la CS ID 4842 que es la estrella que se planteó al inicio.

4.4.5.3. Caso de estudio usando la estrella con ID 521 con introducción de ruido de ± 1.0 pixeles

En este ejemplo se utiliza a la estrella con ID 521 como estrella central (CS). Cuando se usa ésta estrella también existen otras 24 estrellas dentro del campo de visión (FOV). La Tabla 31 muestra las 25 coordenadas esféricas de todas las estrellas dentro de los 12.4 grados del FOV como éstos aparecen en el catálogo de estrellas. Para el presente ejemplo se utilizó las bases de datos de las invariantes correspondientes a la secuencia $n_1 = 3$, $n_2 = 5$ y $n_3 = 6$. Es importante mencionar que las base de datos ya han sido generadas como se indica en la sección 4.2.

Tabla 31: Coordenadas esféricas de las estrellas dentro del FOV para la estrella con ID 521 como estrella central.

Estrella ID	RA (radianes)	Dec (radianes)
521	0.027561	-0.856524
88	0.004697	-0.851893
107	0.005826	-0.878553
763	0.041060	-0.798443
2377	0.132796	-0.841509
2466	0.137077	-0.851782
2705	0.150371	-0.914083
2796	0.155710	-0.837774
116236	6.186536	-0.793991
116948	6.227625	-0.876617
117866	6.278513	-0.920588

La próxima etapa consiste en proyectar todas las estrellas dentro del plano de la imagen sintética usando el algoritmo 2 y agregando un nivel ruido dentro de ± 1.0 pixel.

Desde este instante y en adelante la imagen resultante es fija y será usada en el resto del proceso de identificación. Las coordenadas de todas las estrellas en el FOV serán usadas para crear la imagen sintética con las coordenadas de las estrellas ruidosas, es importante comentar que ninguna información del catálogo de estrellas son usadas durante el proceso para identificar las estrellas. Sin embargo, la identificación se lleva a cabo usando la lista de coordenadas en pixeles con introducción de ruido de la imagen sintética.

Usando la imagen sintética como entrada, un polígono de $n_1 = 3$ vértices es creado como se explica en la subsección 4.3.2 y es ilustrado en la Figura 40.

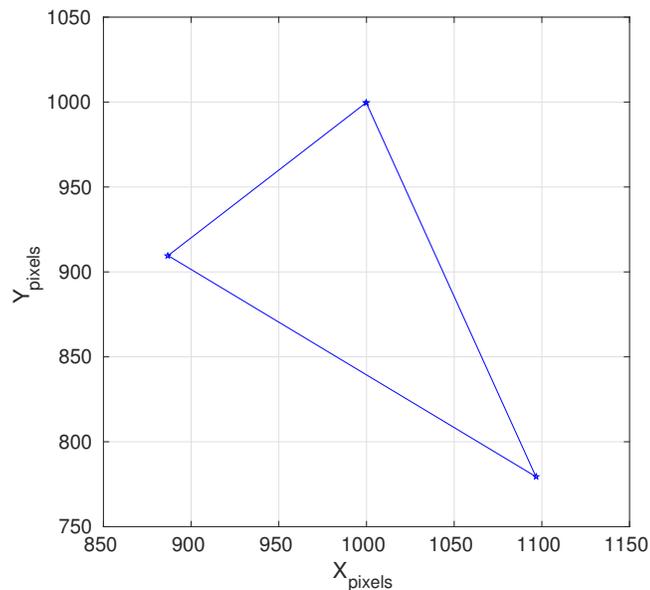


Figura 40: Polígono creado con 3 vértices.

Los vértices de un polígono en coordenadas xy en píxeles son mostrados en la Tabla 32. Usando estos vértices se calcula la invariante como es explicado en la sección 4.1 y se obtiene que $w_{\hat{s}} = -0.168496 + j0.250242$.

Tabla 32: Proyección de coordenadas esféricas a coordenadas xy en píxeles.

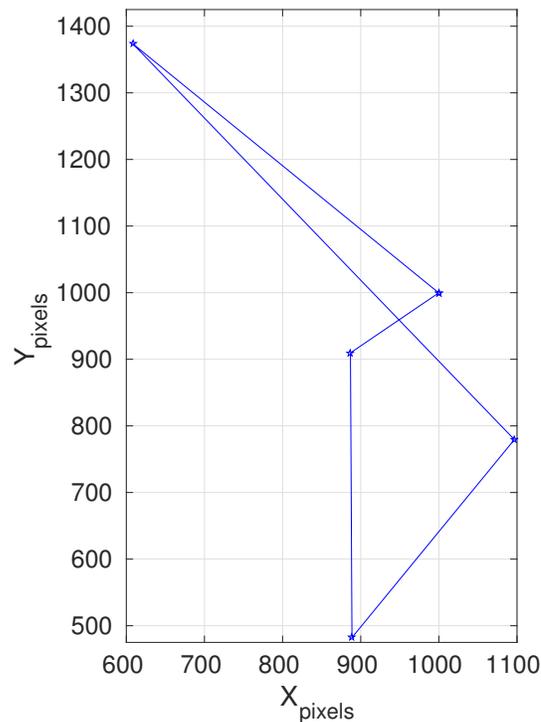
x_{pixel}	y_{pixel}
999.830000	999.670000
886.852054	909.609956
1096.619837	779.430369

La siguiente etapa usa la invariante $w_{\hat{s}}$ para buscarla en la base de datos de invariantes correspondiente a $n_1 = 3$. El procedimiento es explicado en el algoritmo 3. Para este ejemplo en particular existen 6 coincidencias en la base de datos de invariantes, esto es, 6 polígonos en colisión. Los polígonos en colisión son presentados en la Tabla 33 con sus respectivas áreas de la invariante.

Tabla 33: Polígonos en colisiones para la invariante $-0.168496 + j0.250242$.

Estrella ID	max parte Real Invariante	min parte Real Invariante	max parte Imag. Invariante	min parte Imag. Invariante
521	-0.1519695759	-0.1753270604	0.2623763668	0.2390186795
17489	3232.5195498195	-4182.9429558135	3293.2160175888	2.199E-07
17532	2347.0197945952	-2159.4012960162	3592.2100900568	9.8532E-06
17537	6646.1064850981	-3205.9792420233	4416.5257262437	2.406E-07
35132	-0.1571093692	-0.1697659855	0.2547889946	0.2421324032
80221	-0.1671269378	-0.1731991383	0.251062081	0.2449898791

Debido a que se tiene 6 polígonos en colisión se necesita realizar una segunda iteración del algoritmo, sin embargo, esta vez el número de vértices se pone como $n_2 = 5$. Tal como se explicó en la subsección 4.3.2, se crea el polígono ilustrado en la Figura 41.

**Figura 41:** Polígono creado con 5 vértices.

Los vértices del segundo polígono en las coordenadas xy en píxeles son mostrados en la tabla 34. Usando estos vértices se calcula la invariante una vez más, con lo que se obtiene que $w_{\hat{s}} = 0.102727 - j1.014748$.

Tabla 34: Proyección de las coordenadas esféricas a coordenadas xy en pixeles.

x_{pixel}	y_{pixel}
999.830000	999.670000
886.852054	909.609956
888.783867	482.232661
1096.619837	779.430369
608.320644	1373.673749

Como en la iteración anterior, la siguiente etapa consiste en usar la invariante $w_{\hat{s}}$ como una búsqueda en la base de datos correspondiente a $n_2 = 5$. Esta vez la búsqueda no se hace en toda la base de datos, solamente se hace en los polígonos que se encuentran en colisión, es decir, los polígonos asociados a la estrella con IDs 521, 17489, 17532, 17537, 35132 y 80221. Después de la segunda iteración todavía existen dos polígonos en colisión. Estos polígonos son presentados en la Tabla 35 con sus respectivas áreas de invariantes.

Tabla 35: Polígonos en colisión para la invariante $0.102727 - j1.014748$.

Estrella ID	max parte Real Invariante	min parte Real Invariante	max parte Imag. Invariante	min parte Imag. Invariante
521	0.1111637507	0.0917819035	-1.0082497425	-1.0276324445
17537	0.1327022003	-0.0168609103	-0.8812039845	-1.0313116224

Debido a que aún existen 2 polígonos en colisión una tercera iteración del algoritmo es necesario, pero esta vez se tiene que $n_3 = 6$. Con lo que se forma el polígono ilustrado en la Figura 42.

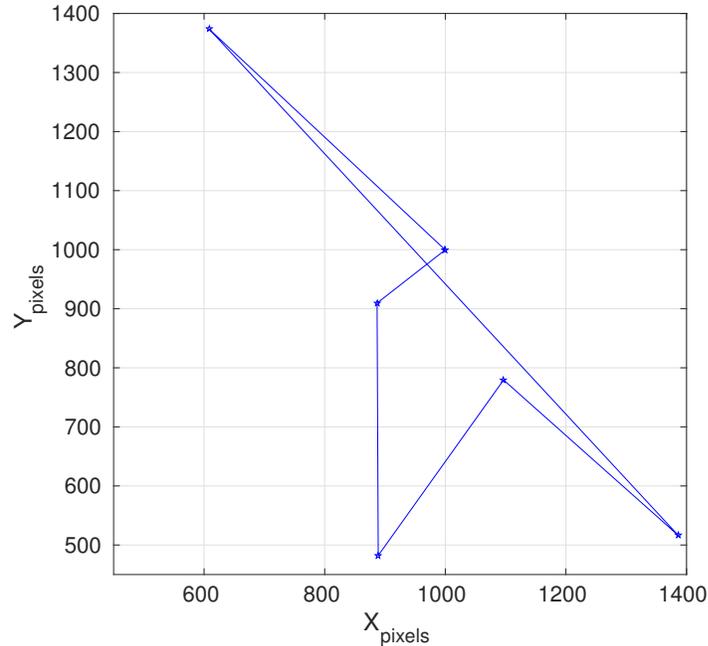


Figura 42: Polígono formado con 6 vértices.

Los vértices del tercer polígono en coordenadas xy en píxeles se muestra en la Tabla 36. Usando estos vértices se calcula una vez más la invariante y se obtiene que $w_{\hat{s}} = 0.389193 - j0.286978$.

Tabla 36: Proyección de las coordenadas esféricas a coordenadas xy en píxeles.

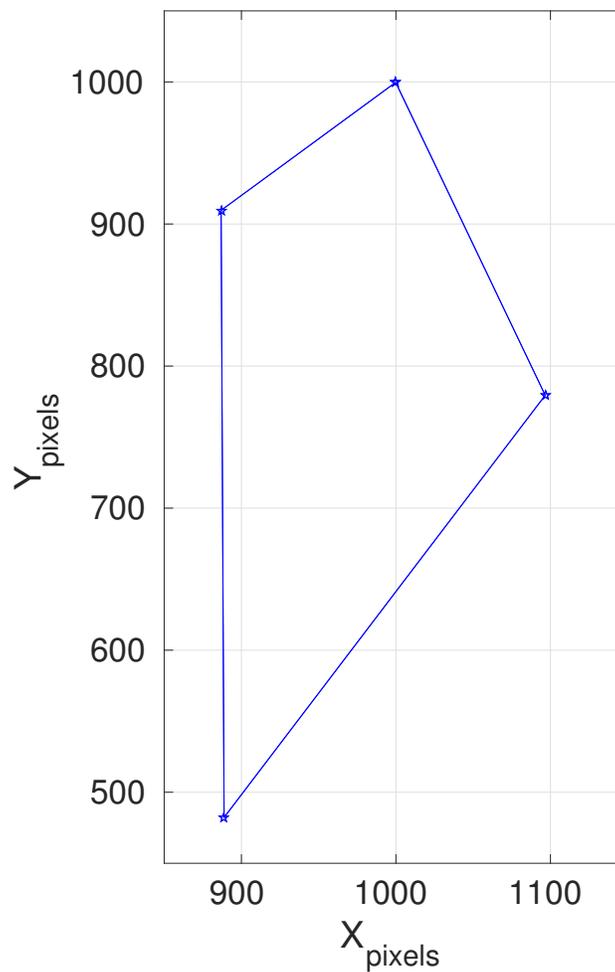
x_{pixel}	y_{pixel}
999.830000	999.670000
886.852054	909.609956
888.783867	482.232661
1096.619837	779.430369
1386.269946	516.282789
608.320644	1373.673749

Como se hizo anteriormente, se usa la invariante calculada $w_{\hat{s}}$ para buscarla en la base datos de la invariante correspondiente a $n_3 = 6$ y solamente se busca en los polígonos que están en colisión de la iteración anterior, esto es, los polígonos asociados a la estrella con IDs 521 y 42415. Después de esta tercera iteración solo se encuentra una sola coincidencia al polígono asociado a la estrella IDs 521 como se puede ver en la tabla 37.

Tabla 37: Polígono coincidente para la invariante $0.389193 - j0.286978$.

Estrella ID	max parte Real Invariante	min parte Real Invariante	max parte Imag. Invariante	min parte Imag. Invariante
521	0.3942746582	0.3830342737	-0.2818117814	-0.2930521864

Para verificar que el polígono identificado efectivamente corresponde a la estrella con IDs 521, se crea un nuevo polígono usando la misma estrella CS y además se usa $n_4 = 4$ como podemos observar en la Figura 43.

**Figura 43:** Polígono creado con 4 vértices.

Los vértices del cuarto polígono en las coordenadas xy píxeles son mostrados en la Tabla 38.

Tabla 38: Proyección de las coordenadas esféricas al plano xy en pixeles.

x_{pixel}	y_{pixel}
999.830000	999.670000
886.852054	909.609956
888.783867	482.232661
1096.619837	779.430369

Usando estos nuevos vértices se vuelve a calcular una vez más la invariante, con lo que se obtiene que $w_{\hat{s}_{n_4}} = 0.373283 + j0.150135$. Entonces, usando la estrella encontrada anteriormente con IDs 521 como búsqueda en la base de datos de la invariante correspondiente se obtiene el área de la invariante del plano complejo como se puede ver en la Tabla 39, y podemos verificar que $w_{\hat{s}_{n_4}}$ en efecto cae dentro de esta área, por lo que se puede asegurar que el polígono encontrado realmente corresponde a la estrella CS con IDs 521.

Tabla 39: Área de la invariante de la base de datos de polígonos con 4 vértices asociado a la estrella ID 521.

Star ID	max. parte Real Invariante	min. parte Real Invariante	max. parte Imag. Invariante	min. parte Imag. Invariante
521	0.3821802366	0.3692368861	0.1565191106	0.1435757601

Con este procedimiento podemos observar que la estrella encontrada coincide con la estrella central con la que comenzó este ejemplo.

Es importante recalcar que el método desarrollado en esta tesis es muy innovador y sus resultados son muy buenos comparados a otros métodos que se encuentran en artículos científicos. En el siguiente capítulo se presentan las conclusiones a este trabajo.

Capítulo 5. Conclusiones y trabajo a futuro

5.1. Conclusiones

En esta tesis se ha desarrollado un nuevo método muy innovador para la identificación de estrellas basado en generar polígonos para cada estrella dentro del catálogo de estrellas y asociando a cada polígono con un identificador único, el cual es invariante a la transformación de semejanza, que se refiere a que su valor es invariante a las operaciones de escalamiento, rotación y desplazamiento. Los vértices de cada polígono son las posiciones de las estrellas que se encuentran en la vecindad dentro del FOV.

Se demostró que este algoritmo es altamente confiable aún a niveles de $\pm 424 \mu\text{rad}$, con una tasa de reconocimiento muy alta y puede ser usado como una parte de un sensor de estrellas real. También se demostró que el algoritmo puede identificar estrellas en tiempo mas rápido comparados a todos algoritmos de identificación de estrellas que tienen reportados sus tiempos de identificación, además el algoritmo desarrollado puede ejecutarse en procesadores ARM de 400MHz usados en algunos cubesats.

Se crearon dos catálogos de estrellas los cuales contienen 5,024 estrellas para intensidades ≤ 6.0 y de 2,841 estrellas para intensidades ≤ 5.5 , los cuales son parte del catálogo de HYG. Se presentó el concepto de *invariante a la transformación de la semejanza* (Chávez *et al.*, 2013) que mapea un polígono a un número complejo, lo cual es una ventaja ya que solo se busca un dato comparado a los otros algoritmos que buscan más información en la base de datos.

La invariante a la transformación a la semejanza es analítica en el sentido que si se introduce ruido uniformemente distribuido en la posición de cada uno de los vértices del polígono, esto provoca un cambio controlado del número complejo correspondiente, si a un polígono se le introduce ruido posicional de forma repetida con niveles distintos se produce un área del cambio de la invariante. Sin embargo, un problema asociado al introducir ruido posicional en los vértices formados con estrellas es que si el nivel de ruido es lo suficiente grande para hacer que el orden de los vértices cambien o las estrellas usadas para los vértices puedan cambiar, hace que el polígono cambie su forma de forma radical por lo que los valores de las invariantes no se encuentren dentro de la misma área, esto provoca que se tengan varias áreas de invariantes distintas.

En orden a indexar el catálogo se construye una base de datos de invariantes los cuales asocian cada estrella a un número de áreas de invariantes en el plano complejo. Estas áreas de

invariantes son el resultado de calcular un millón de veces a la invariante por cada estrella del catálogo.

Para evaluar el rendimiento del reconocimiento de estrellas, se crean imágenes sintéticas para cada uno de los cuerpos celestes dentro del catálogo de estrellas. Para cada imagen sintética, se introduce ruido uniformemente distribuido a la posición de las estrellas y se aplica una transformación de semejanza. Después de esto se forma un polígono y se calcula su invariante compleja. Cada invariante compleja se usa para buscar dentro de la base de datos de la invariante correspondiente para encontrar una coincidencia. Si por algún motivo se encuentra una colisión (en esta tesis se le llama colisión cuando el valor de una invariante calculada existe en más de un área de invariante para estrellas distintas), se modifican al número de vértices para crear otro polígono que pertenece a la misma estrella central, se vuelve a calcular la invariante y se hace su búsqueda en su base de datos correspondiente, para comprobar si el polígono reconocido es válido se realiza por lo menos una iteración extra para verificar que la nueva invariante corresponde al polígono reconocido.

El método propuesto es capaz de identificar correctamente en dos iteraciones el 99.999 % de 50.24 millones de polígonos creados, utilizando una magnitud aparente ≤ 6.0 y con un nivel de ruido introducido de ± 0.5 pixel. Si el nivel de ruido se incrementa a ± 1.0 pixeles, este método aún es capaz de identificar el 99.999 % de los polígonos pero se necesitan 3 iteraciones. Si el nivel de ruido se incrementa a 2 o 3 pixels, el algoritmo es capaz de identificar al 99.9 % de polígonos, con cuatro y cinco iteraciones respectivamente. Incrementando el nivel de ruido a 4 pixeles es necesario usar ocho iteraciones para identificar el 99.68 % de los polígonos. En el caso de utilizar magnitud aparente ≤ 5.5 , y niveles de ruido de 1, 2, 3 or 4 pixels, el algoritmo es capaz de identificar 99.999 %, 99.85 %, 99.7 %, 99.44 % de los polígonos después de 3, 3, 5 and 7 iteraciones respectivamente después de evaluar 28.41 millones de polígonos. Para el caso de imágenes sintéticas ideales y sin ruido se pudieron identificar todos los polígonos.

El algoritmo propuesto comprueba que los polígonos reconocidos corresponden al polígono de las estrellas bajo análisis, el resultado final es que no existe ningún caso de reconocimiento de un polígono que no corresponda a la estrella en estudio.

Algunas actividades sugeridas para continuar avanzando con este trabajo se enuncian en la siguiente sección.

Basado en este trabajo de investigación fue posible escribir un artículo científico para su publi-

cación:

Hernández, E.A., Alonso, M.A., Chávez, E., Covarrubias, D.H. and Conte, R., Feb 2017. Robust polygon recognition method with similarity invariants applied to star identification. *Advances in Space Research*, Volume 59, Issue 4, Pages 1095–1111.

5.2. Trabajo a Futuro

En su forma actual el algoritmo propuesto no puede reconocer polígonos para el caso cuando algunas de las estrellas de los polígonos creados desaparecen o cuando aparece una estrella falsa en la imagen y que no forma parte del catálogo de estrellas. En este caso el algoritmo solo indica que no encontró ningún polígono. Como un trabajo a futuro es necesario extender este trabajo para que el método sea capaz de identificar a los polígonos aún en la ausencia de estrellas válidas o en la presencia de estrellas falsas. Esto se puede lograr agregando alguna técnica tal como la combinación inteligente de índices de estrellas desarrollada por Mortari *et al.* (2004).

Al realizar esta tesis se pudo observar que un área importante que no existen muchos estudios desarrollados se encuentra en el campo del procesamiento de la imagen capturada en el espacio, teniendo en cuenta que uno de los retos a enfrentar es la iluminación no homogénea debido a rayos del Sol o de la Luna reflejándose a un cierto ángulo del sensor de estrellas y pudiera suceder que éstas se presentan en áreas en donde la brillantez de las estrellas sean de baja intensidad.

Literatura citada

- Abdullah, S., PirahanSiah, F., Abidin, N. Z., y Sahran, S. (2010). Multi-threshold approach for license plate recognition system. *World Acad. Sci. Eng. Technol*, **48**: 804–808.
- Abdullah, S. L. S., Jamil, N., *et al.* (2012). Segmentation of natural images using an improved thresholding-based technique. *Procedia Engineering*, **41**: 938–944.
- Abubaker, A., Qahwaji, R., Ipson, S., y Saleh, M. (2007). One scan connected component labeling technique. En: *Signal Processing and Communications, 2007. ICSPC 2007. IEEE International Conference on*. IEEE, pp. 1283–1286.
- Agarwal, T. (2015). Types of charge-coupled devices with their working principles. <https://www.elprocus.com/wp-content/uploads/2014/07/Frame-transfer.jpg>.
- Ali, J., Zhang, C., y Fang, J. (2006). An algorithm for astro-inertial navigation using ccd star sensors. *Aerospace science and technology*, **10**(5): 449–454.
- Alibaba (2017). Fastest rotation angular gyroscope sensor with measuring speed of 300 degrees per second. https://www.alibaba.com/product-detail/Fastest-Rotation-Angular-Gyroscope-Sensor-With_1873220489.html.
- Anantrasirichai, N., Nicholson, L., Morgan, J. E., Erchova, I., Mortlock, K., North, R. V., Albon, J., y Achim, A. (2014). Adaptive-weighted bilateral filtering and other pre-processing techniques for optical coherence tomography. *Computerized Medical Imaging and Graphics*, **38**(6): 526–539.
- Anthony, S. M. y Granick, S. (2009). Image analysis with rapid and accurate two-dimensional gaussian fitting. *Langmuir*, **25**(14): 8152–8160.
- Arbabmir, M. V., Mohammadi, S. M., Salahshour, S., y Somayehee, F. (2014). Improving night sky star image processing algorithm for star sensors. *J. Opt. Soc. Am. A*, **31**(4): 794–801.
- Arora, S., Acharya, J., Verma, A., y Panigrahi, P. K. (2008). Multilevel thresholding for image segmentation through a fast statistical recursive algorithm. *Pattern Recognition Letters*, **29**(2): 119–125.
- Auden, W. H. (2011). Coordinate systems. <https://dept.astro.lsa.umich.edu/ugactivities/Labs/coords/>.
- Aurich, V. y Weule, J. (1995). Non-linear gaussian filters performing edge preserving diffusion. En: *Mustererkennung 1995*. Springer, pp. 538–545.
- Azizabadi, M., Behrad, A., y Ghaznavi-Ghouschi, M. (2014). Vlsi implementation of star detection and centroid calculation algorithms for star tracking applications. *Journal of real-time image processing*, **9**(1): 127–140.
- Balancer (2010). Trait-bryan angles. <http://forums.airbase.ru/2010/12/t76164--trait-bryan-angles.2416.html>.
- Baldini, D.; Barni, M. y Foggi, A. (1993). A new star constellation matching algorithm for satellite attitude determination. *ESA Journal*, **17**: 185–198.
- Ball, R. S. (1915). *A treatise on spherical astronomy*. Cambridge University Press.
- Bessell, M. S. (2005). Standard photometric systems. *Annu. Rev. Astron. Astrophys.*, **43**: 293–336.
- Bohlin, R. y Gilliland, R. (2004). Hubble space telescope absolute spectrophotometry of vega from the far-ultraviolet to the infrared. *The Astronomical Journal*, **127**(6): 3508.

- Brekke, P., Chaloupy, M., Fleck, B., Haugan, S., van Overbeek, T., y Schweitzer, H. (2004). Space weather effects on soho and its space weather warning capabilities. En: *Effects of Space Weather on Technology Infrastructure*. Springer, pp. 109–122.
- Chávez, E., Chávez-Cáliz, A. C., y López-López, J. L. (2013). Polygon matching and indexing under affine transformations. *CoRR*, **abs/1304.4994**.
- Cochran, A. (1981). Spectrophotometry with a self-scanned silicon photodiode array. ii-secondary standard stars. *The Astrophysical Journal Supplement Series*, **45**: 83–96.
- Cooperk (2014). Hpu ncs2200 earth moon-sun relationships lecture. <http://pt.slideshare.net/cooperk2/hpu-ncs2200-earth-moonsun-relationships-lecture>.
- Dass, R. y Devi, S. (2012). Image segmentation techniques 1.
- Davidson, M. W. (2015). Understanding digital imaging. <http://zeiss-campus.magnet.fsu.edu/print/basics/digitalimaging-print.html>.
- Delac, K., Grgic, M., y Kos, T. (2006). Sub-image homomorphic filtering technique for improving facial identification under difficult illumination conditions. En: *International Conference on Systems, Signals and Image Processing*. Vol. 1, pp. 21–23.
- ESA, E. D. (2008). Space weather effects and applications. <http://sidc.be/esww5/presentations/tutorial/daly.ppt>.
- ESRL, E. S. R. L. (2017). ecliptic plane. <https://www.esrl.noaa.gov/gmd/grad/solcalc/ecliptic.gif>.
- FACG (2013). The international celestial reference system (icrs). <http://www.iers.org/IERS/EN/Science/ICRS/ICRS.html>.
- Feng-gen, C., Bin, L., Tao, Z., y Jun, Y. (2011). Improved triad for star sensor. En: *Computer Science and Network Technology (ICCSNT), 2011 International Conference on*. IEEE, Vol. 1, pp. 282–286.
- Fruchter, A. *et al.* (2009). Hst multidrizzle handbook. *HST MultiDrizzle, HST Data Handbooks*, **1**.
- Goodrich (2004). Hd-1003 star tracker attitude sensor. http://www.isr.goodrich.com/datasheets/oss/Goodrich_HD1003_StarTracker.pdf.
- Groth, E. J. (1986). A pattern-matching algorithm for two-dimensional coordinate lists. *The astronomical journal*, **91**: 1244–1248.
- Guide, E. (2015). Lens basics. <http://www.exposureguide.com/lens-basics.htm>.
- Gupta, G. (2011). Algorithm for image processing using improved median filter and comparison of mean, median and improved median filter. *International Journal of Soft Computing and Engineering (IJSCE) ISSN*, pp. 2231–2307.
- Hazard, C., Sutton, J., Argue, A., Kenworthy, C. M., Morrison, L., y Murray, C. (1971). Accurate radio and optical positions of 3g273b. *Nature*, **233**(40): 89–91.
- He, L., Chao, Y., Suzuki, K., y Wu, K. (2009). Fast connected-component labeling. *Pattern Recognition*, **42**(9): 1977–1987.
- Ho, K. (2012). A survey of algorithms for star identification with low-cost star trackers. *Acta Astronautica*, **73**: 156–163.

- Inc, E. O. (2016a). Understanding optical specifications. <http://www.edmundoptics.com/resources/application-notes/optics/understanding-optical-specifications/>.
- Inc, E. O. (2016b). Introduction to modulation transfer function. <http://www.edmundoptics.com/resources/application-notes/optics/introduction-to-modulation-transfer-function>.
- Janson, S., Hardy, B., Chin, A., Rumsey, D., Ehrlich, D., y Hinkley, D. (2012). Attitude control on the pico satellite solar cell testbed-2.
- Jia, H., Yang, J., y Li, X. (2010). Minimum variance unbiased subpixel centroid estimation of point image limited by photon shot noise. *JOSA A*, **27**(9): 2038–2045.
- Jiancheng, F., Saffih, F., Wei, Q., *et al.* (2009). Automatic star pattern recognition in tracking mode. En: *Applied Sciences and Technology (IBCAST), 2009 6th International Bhurban Conference on*. IEEE, pp. 249–252.
- Jiancheng F., L. N. y T., Y. (2006). *Autonomous Celestial Navigation Elements and Method for Spacecraft*. Beijing University of Aeronautics and Astronautics. National Defense Industry press.
- Jing, Y. y Liang, W. (2012). An improved star identification method based on neural network. En: *Industrial Informatics (INDIN), 2012 10th IEEE International Conference on*, July. pp. 118–123.
- Jørgensen, J. L. (2000). In-orbit performance of a fully autonomous star tracker. En: *Spacecraft Guidance, Navigation and Control Systems*. Vol. 425, p. 103.
- Junkins, J. L., Strikwerda, T. E., y Kraige, L. G. (1978). Star pattern recognition and spacecraft attitude determination. Reporte técnico, DTIC Document.
- Kaplan, G., Josties, F., Angerhofer, P., Johnston, K., y Spencer, J. (1982). Precise radio source positions from interferometric observations. *The Astronomical Journal*, **87**: 570–576.
- Katake, A. B. (2006). *Modeling, image processing and attitude estimation of high speed star sensors*. Tesis de doctorado, Texas A&M University.
- Kawano, H., Shimoji, H., Yoshikawa, S., Miyatake, K., Hama, K., y Nakamura, S. (2008). Optical testing of star sensor (ii): Alignment of star sensor from off-line pictures taken in orbital test. *Optical review*, **15**(2): 118–124.
- Khan, W. (2013). Image segmentation techniques: A survey. *Journal of Image and Graphics*, **1**(4): 166–170.
- Kolomenkin, M., Pollak, S., Shimshoni, I., y Lindenbaum, M. (2008). Geometric voting algorithm for star trackers. *Aerospace and Electronic Systems, IEEE Transactions on*, **44**(2): 441–456.
- Kosik, J. C. (1991). Star pattern identification: Application to the precise attitude determination of the auroral spacecraft. *Journal of Guidance, Control and Dynamics*, **14**(2): 230–235.
- Lee, H. y Bang, H. (2007). Star pattern identification technique by modified grid algorithm. *IEEE Transactions on Aerospace and Electronic Systems*, **43**(3): 1112–1116.
- Li, Q., Xu, W., Xu, Z., y Feng, H. (2013). Influence of threshold segmentation on star pair angular separation measurement accuracy. En: *Measuring Technology and Mechatronics Automation (ICMTMA), 2013 Fifth International Conference on*. IEEE, pp. 635–638.
- Liebe, C. (1995). Star trackers for attitude determination. *Aerospace and Electronic Systems Magazine, IEEE*, **10**(6): 10–16.

- Liebe, C. C. (1992). Pattern recognition of star constellations for spacecraft applications. *IEEE Aerospace and Electronic Systems Magazine*, **7**(6): 34–41.
- Liebe, C. C. (2002). Accuracy performance of star trackers - a tutorial. *IEEE Transactions on Aerospace and Electronic Systems*, **38**(2): 587–599.
- Lieske, J. H., Lederle, T., Fricke, W., y Morando, B. (1977). Expressions for the precession quantities based upon the iau/1976/system of astronomical constants. *Astronomy and Astrophysics*, **58**: 1–16.
- Lindsey, C. S., Lindblad, T., y Eide, A. J. (1997). Method for star identification using neural networks. *Proc. SPIE*, **3077**: 471–478.
- LIU, C.-s., HUANG, X., y LIU, G.-b. (2004). A convex polygon star pattern identification algorithm for star sensor [j]. *Opto-electronic Engineering*, **9**: 001.
- Luo, L., Xu, L., y Zhang, H. (2015). An autonomous star identification algorithm based on one-dimensional vector pattern for star sensors. *Sensors*, **15**(7): 16412–16429.
- M., M.-U.-R., Jiancheng, F., Saffih, F., y Wei, Q. (2009). Automatic star pattern recognition in tracking mode. En: *Applied Sciences and Technology (IBCAST), 2009 6th International Bhurban Conference on*, Jan. pp. 249–252.
- Maini, R. y Aggarwal, H. (2010). A comprehensive review of image enhancement techniques. *arXiv preprint arXiv:1003.4053*.
- Markley, F. L. y Mortari, D. (2000). Quaternion attitude estimation using vector observations. *Journal of the Astronautical Sciences*, **48**(2): 359–380.
- MathWorks (2014). *Aerospace Toolbox User's Guide*. MathWorks Inc.
- McBryde, C. R. y Lightsey, E. G. (2012). A star tracker design for cubesats. En: *Aerospace Conference, 2012 IEEE*. IEEE, pp. 1–14.
- Mohammadi, S. M., Helfroush, M. S., y Kazemi, K. (2012). Novel shape-texture feature extraction for medical x-ray image classification. *Int J Innov Comput Inf Control*, **8**: 659–76.
- Mohammadnejad, S., Rostami, A., y Sarvi, M. N. (2012). A practical view to database generation of star identification algorithm for space applications. En: *Communication Systems, Networks & Digital Signal Processing (CSNDSP), 2012 8th International Symposium on*. IEEE, pp. 1–4.
- Mortari, D., Junkins, J. L., y Samaan, M. (2001). Lost-in-space pyramid algorithm for robust star pattern recognition. En: *Guidance and control Conference*. pp. 49–68.
- Mortari, D., Samaan, M. A., Bruccoleri, C., y Junkins, J. L. (2004). The pyramid star identification technique. *Navigation*, **51**(3): 171–183.
- NASA (1965). Operation of the horizon sensor for gemini spacecraft. <http://history.nasa.gov/SP-4002/images/fig22.jpg>.
- Nash, D. (2014). The hyg database. <http://www.astronexus.com/hyg>.
- Nelli, F. (2014). Arduino: measuring the earth's magnetic field with the magnetometer hmc5883l. <http://www.meccanismocomplesso.org/en/arduino-magnetic-magnetic-magnetometer-hmc5883l/>.

- Nightingale, A. M. y Gordeyev, S. (2013). Shack-hartmann wavefront sensor image analysis: a comparison of centroiding methods and image-processing techniques. *Optical Engineering*, **52**(7): 071413–071413.
- Osborne, J. (2010). Mapping the celestial sphere locating stars and deep space objects. <http://slideplayer.com/slide/10115091/>.
- Padgett, C. y Kreutz-Delgado, K. (1997). A grid algorithm for autonomous star identification. *Aerospace and Electronic Systems, IEEE Transactions on*, **33**(1): 202–213.
- Paladugu, L., Williams, B., y Schoen, M. (2003). Star pattern recognition for attitude determination using genetic algorithms. En: *17th Annual AIAA/USU Conference on Small Satellites*. pp. 1–9.
- Paris, S., Kornprobst, P., Tumblin, J., y Durand, F. (2009). *Bilateral filtering: Theory and applications*. Now Publishers Inc.
- Patel, I. (2015). Astronomical coordinate system. <http://www.slideshare.net/ishanpatel19/astronomical-coordinate-system>.
- Perryman, M. A., Lindegren, L., Kovalevsky, J., Hoeg, E., Bastian, U., Bernacca, P., Crézé, M., Donati, F., Grenon, M., Grewing, M., *et al.* (1997). The hipparcos catalogue. *Astronomy and Astrophysics*, **323**: L49–L52.
- Pham, M. D., Low, K.-S., y Chen, S. (2013). An autonomous star recognition algorithm with optimized database. *IEEE Transactions on Aerospace and Electronic Systems*, **49**(3): 1467–1475.
- PirahanSiah, F., Abdullah, S. N. H. S., y Sahran, S. (2014). Adaptive image thresholding based on the peak signal-to-noise ratio. *Research Journal of Applied Sciences, Engineering and Technology*, pp. 1104–1116.
- Pogosian, D. (2017). Lecture 2: Angles. https://sites.ualberta.ca/~pogosyan/teaching/ASTRO_122/lect2/lecture2.html.
- Quine, B. y Durrant-Whyte, H. F. (1996). A fast autonomous star-acquisition algorithm for spacecraft. *Control Engineering Practice*, **4**(12): 1735–1740.
- Race, B. W. (2014). Sistema de coordenadas geográficas. <http://www.barcelonaworldrace.org/es/educacion/programa-educativo/explora/navegacion/tecnologia-del-imoca-60/posicionamiento/sistema-de-coordenadas-geograficas>.
- Rad, A. M., Nobari, J. H., y Nikkhah, A. A. (2014). Optimal attitude and position determination by integration of ins, star tracker, and horizon sensor. *Aerospace and Electronic Systems Magazine, IEEE*, **29**(4): 20–33.
- Rohác, J., Rerábek, M., y Hudec, R. (2011). Multi-functional star tracker-future perspectives. *Acta Polytechnica*, **51**(6): 61.
- Rozelle, D. M. (2009). The hemispherical resonator gyro: From wineglass to the planets. En: *Proc. 19th AAS/AIAA Space Flight Mechanics Meeting*. pp. 1157–1178.
- Samaan, M. y Theil, S. (2012). Development of a low cost star tracker for the shefex mission. *Aerospace science and technology*, **23**(1): 469–478.
- Samaan, M. A. (2003). *Toward faster and more accurate star sensors using recursive centroiding and star identification*. Tesis de doctorado, Texas A&M University.

- Scholl M. S. (1995). Star-Field Identification for Autonomous Attitude Determination. *Journal of Guidance, Control, and Dynamics*, **18**(1): 61–65. doi: 10.2514/3.56657.
- Seidelmann, P. (1982). 1980 iau theory of nutation: The final report of the iau working group on nutation. *Celestial Mechanics*, **27**(1): 79–106.
- Shuster, M. (2006). The quest for better attitudes. *The Journal of the Astronautical Sciences*, **54**(3-4): 657–683.
- Smart, W. y Green, R. (1977). *Textbook on Spherical Astronomy*. Cambridge University Press.
- Smith, S. M. y Brady, J. M. (1997). Susan—a new approach to low level image processing. *International journal of computer vision*, **23**(1): 45–78.
- Spratling, B. B. y Mortari, D. (2009). A survey on star identification algorithms. *Algorithms*, **2**(1): 93–107.
- Steyn, W., Jacobs, M., y Oosthuizen, P. (1997). A high performance star sensor system for full attitude determination on a microsatellite. En: *Proceedings of the Workshop on Control of Small Spacecraft at the 1997 Annual AAS Guidance and Control Conference, Breckenridge, CO, USA*. Vol. 5.
- Stone, R. C. (1989). A comparison of digital centering algorithms. *The Astronomical Journal*, **97**: 1227–1237.
- Sun, T., Xing, F., y You, Z. (2013). Optical system error analysis and calibration method of high-accuracy star trackers. *Sensors*, **13**(4): 4598–4623.
- Tomasi, C. y Manduchi, R. (1998). Bilateral filtering for gray and color images. En: *Computer Vision, 1998. Sixth International Conference on*. IEEE, pp. 839–846.
- Von Fintel, R. (2015). White paper: Modern cmos cameras as replacements for ccd cameras.
- Waltham, N. (2013). Ccd and cmos sensors. En: *Observing photons in space*. Springer, pp. 423–442.
- Wang, Z., Wu, B., y Sergienko, T. (2012). A non-linearity correction method for calibration of optical sensor at low level light. En: *AsiaSim 2012*. Springer, pp. 126–134.
- Wei, X., Zhang, G., y Jiang, J. (2009). Star identification algorithm based on log-polar transform. *Journal of Aerospace Computing, Information, and Communication*, **6**(8): 483–490.
- Wen-Cheng, W. y Xiao-Jun, C. (2013). A segmentation method for uneven illumination particle images. *Research Journal of Applied Sciences, Engineering and Technology*, **5**(4): 1284–1289.
- Wertz, J. R. (2012). *Spacecraft attitude determination and control*, Vol. 73. Springer Science & Business Media.
- Westlake, A. (2012). Digital camera lens buying guide. <http://www.dpreview.com/articles/9162056837/digital-camera-lens-buying-guide>.
- Woebcke, C. (2016). The precession of the equinoxes. <http://www.myastronomybook.com/precession-of-the-equinoxes.htm>.
- Xiaojuan, W. y Xinlong, W. (2011). Multiple blur of star image and the restoration under dynamic conditions. *Acta Astronautica*, **68**(11): 1903–1913.

- Xie, J. y Wang, X. (2012). A robust autonomous star identification algorithm for zy3 satellite. En: *Agro-Geoinformatics (Agro-Geoinformatics), 2012 First International Conference on*. IEEE, pp. 1–4.
- Xie, J., Tang, X., Jiang, W., y Fu, X. (2012). An autonomous star identification algorithm based on the directed circularity pattern. *star*, **4**: 5.
- Xu, W., Li, Q., Feng, H.-j., Xu, Z.-h., y Chen, Y.-t. (2013). A novel star image thresholding method for effective segmentation and centroid statistics. *Optik-International Journal for Light and Electron Optics*, **124**(20): 4673–4677.
- Yang, J., Liang, B., Zhang, T., y Song, J. (2011). A novel systematic error compensation algorithm based on least squares support vector regression for star sensor image centroid estimation. *Sensors*, **11**(8): 7341–7363.
- Yang, X., Shen, X., Long, J., y Chen, H. (2012). An improved median-based otsu image thresholding algorithm. *AASRI Procedia*, **3**: 468–473.
- Yaroslavsky, L. P. y Yaroslavskij, L. (1985). Digital picture processing. an introduction. *Digital picture processing. An introduction.. LP Yaroslavsky (LP Yaroslavskij). Springer Series in Information Sciences, Vol. 9. Springer-Verlag, Berlin-Heidelberg-New York-Tokyo. 12+ 276 pp. Price DM 112.00 (1985). ISBN 3-540-11934-5 (FR Germany), ISBN 0-387-11934-5 (USA).*, **1**.
- Yoon, H., Paek, S. W., Lim, Y., Lee, B.-H., y Lee, H. (2013). New star pattern identification with vector pattern matching for attitude determination. *Aerospace and Electronic Systems, IEEE Transactions on*, **49**(2): 1108–1118.
- Zhang, W., Quan, W., y Guo, L. (2012a). Blurred star image processing for star sensors under dynamic conditions. *Sensors*, **12**(5): 6712–6726.
- Zhang, W., Quan, W., y Guo, L. (2012b). Blurred star image processing for star sensors under dynamic conditions. *Sensors*, **12**(5): 6712–6726.
- Zhang G., W. X. y J., J. (2008). Full-sky autonomous star identification based on radial and cyclic features of star pattern. *Image and Vision Computing*, **26**(2): 891–897.
- Zhao, Y., Wei, X., Li, J., y Wang, G. (2016). Star identification algorithm based on k-l transformation and star walk formation. *IEEE Sensors Journal*, **16**(13): 5202–5210.
- Zurich, E. (2017). Experimental precession. <http://www.epm.ethz.ch/research/experimental/experimental-precession.html>.

Apéndice

Código del agrupamiento de dos escaneos por mínimos y máximos

```

for cont2_obj=1:cont_obj
    band=0;
    for k=1:cont_obj
        if (num_obj(k)==0)
            num_obj(k)=cont2_obj;
            min=minx(k);
            max=maxx(k);
            y=y_obj(k);
            num_estrellas=num_estrellas+1;
            band=1;
            break;
        end
    end
    if (band==1)
        for kk=1:cont_obj
            if (num_obj(kk)==0)
                min2=minx(kk);
                max2=maxx(kk);
                y2=y_obj(kk);
                if ((min >= min2 && max <= max2)
                    || (min2 >= min && max2 <= max)
                    || (min2 >= min && max2 >= max && min2 <= max)
                    || (min >= min2 && max2 <= max && min <= max2))
                    && abs(y-y2)==1)
                    min=min2;
                    max=max2;
                    y=y2;
                    num_obj(kk)=cont2_obj;
                else
                    if (((min2-max2)==0 && abs(min-min2)<=1
                        && abs(y-y2)==1)||((min2-max2)==0
                        && abs(max-max2)<=1 && abs(y-y2)==1))
                        min=min2;
                        max=max2;
                        y=y2;
                        num_obj(kk)=cont2_obj;
                    end
                end
            end
        end
    end
end
end
end
end
end

```

Información del catálogo HYG

La información dentro del catálogo de HYG está ordenada de la siguiente manera (Nash, 2014):

- ID: La identificación principal de las estrellas en el catálogo.
- Hip: La ID de las estrellas en el catálogo de Hipparcos, solo si existe esta información.
- HD: La ID de las estrellas en el catálogo de Henry Draper, solo si existe esta información.
- HR: La ID de las estrellas en el catálogo de Harvard, que tiene el mismo número del catálogo de estrellas brillantes de Yale, solo si existe esta información.
- Gliese: La ID de las estrellas cercanas del catálogo de Gliese tercera edición, si existe ésta información.
- Bayer Flamsteed: Designación de Bayer / Flamsteed, del catálogo de estrellas brillantes de Yale quinta edición. Ésta es una combinación de las 2 designaciones. El número de Flamsteed, si está presente se pone primero; después la abreviación de las tres letras griegas de Bayer; el número superscript de Bayer, si es que existe; y finalmente, tres letras de la abreviación de la constelación. Por ejemplo para Alfa Andrómeda tiene el valor en el campo de "21AlpAnd", y para Kappa1 Sculptoris (sin número de Flamsteed) tiene "Kap1Sci".
- RA, Dec: La ascensión a la derecha y la declinación, para la época 2000.0. Las estrellas en el catálogo Gliese han sido trasladadas a la época y equinoccio 2000.0, ya que este catálogo utiliza las coordenadas del año 1950.
- ProperName: Es el nombre común para las estrellas, tal como "Estrella de Barnard" o "Sirius". Los cuales se encuentran en el proyecto de Hipparcos, el cual enlista los nombres representativos para las 150 estrellas más brillantes y de las 150 estrellas más cercanas.
- Distancia: La distancia de las estrellas en parsecs, es la unidad más conocida en astronomía. Para convertir parsecs a años luz, multiplicar por 3.26156. Un valor mayor o igual a 10,000,000 indica que no se encuentra el dato de paralaje o se tienen dudas.
- pmra, pmdec: El movimiento de las estrellas en su ascensión a la derecha y su declinación, en milisegundos por año.

- *rv*: La velocidad radial de las estrellas en km/sec, solo si existe esta información.
- *Mag*: La **magnitud visual** aparente de las estrellas.
- *AbsMag*: La **magnitud visual** absoluta (La **magnitud aparente** a una distancia de 10 parsecs).
- *Spectrum*: El tipo espectral de la estrella, solo si existe esta información.
- *ColorIndex*: El índice de color de la estrella (magnitud azul - **magnitud visual**), solo si existe esta información.
- *X,Y,Z*: Las coordenadas cartesianas de la estrella, en un sistema basado en las coordenadas ecuatoriales como son vistas en Tierra. *+X* es la dirección del equinoccio vernal (a la época 2000), *+Z* hacia el polo celeste norte y en *+Y* en la dirección 6 Horas de la ascensión a la derecha, declinación 0°.
- *VX,VY,VZ*: Son los componentes de la velocidad cartesiana de las estrellas, en el mismo sistema de coordenadas descritas anteriormente. Estos determinan al movimiento y a la velocidad radial (cuando son conocidos). La unidad de velocidad es parsecs por año; de ahí los valores muy pequeños entre 10^{-5} y 10^{-6} , usando parsecs como unidad de base para el mapeo celeste simplifica los cálculos enormemente.
- *rarad, dehrad, pmrarad, prdehrad*: La posición está en radianes, y sus movimientos son en radianes por año.
- *Bayer*: la designación Bayer.
- *Flam*: El número de Flamsteed.
- *Comp, Comp_primary, Base*: Identifica una estrella en un sistema múltiple de estrellas. *Comp* = ID de la estrella acompañante, *Comp_primary* = ID de la estrella primaria, y *base* = catalogo ID or nombre de este sistema de múltiples estrellas. Solo usada para estrellas Gliese.
- *lum*: La luminosidad de las estrellas como un múltiplo de la luminosidad solar.
- *var*: Designación de la estrella variable estándar para las estrellas, cuando ésta es conocida.
- *var_min, var_max*: El intervalo de la magnitud aproximada de las estrellas variables. Este valor está basado en las magnitudes promedio H_P del catalogo de Hipparcos original, ajustadas a la escala de magnitud V que coincide con el campo "Mag".

Algoritmos útiles

Algoritmo 5 Txy() function - Transformation of star coordinates to xy plane

1: INPUT: The δ_i, α_i , the declination and right ascension of star coordinate

2: $x_{s_i} \leftarrow \tan(\alpha_i - \alpha_1)$

3: $y_{s_i} \leftarrow \frac{\tan(\delta_i - \delta_1)}{\cos(\alpha_i - \alpha_1)}$
 { } x_{s_i}, y_{s_i}

Algoritmo 6 TPxy() function - Transformation of xy coordinates to pixels

1: INPUT: The x_{s_i}, y_{s_i} , the x and y

2: $\hat{x}_{s_i} \leftarrow \frac{N_x}{2} \cdot \frac{1}{\tan(\frac{FOV}{2.0})} \cdot x_{s_i} + \frac{N_x}{2} + rand()_{(i*2)}$

3: $\hat{y}_{s_i} \leftarrow \frac{N_y}{2} \cdot \frac{1}{\tan(\frac{FOV}{2.0})} \cdot y_{s_i} + \frac{N_y}{2} + rand()_{(i*2+1)}$
 { the xy plane transformed to pixels }
