

PROYECTO INTEGRADOR DE INGENIERÍA MECÁNICA

**DESARROLLO Y ENSAYOS DE ALGORITMOS PARA STAR
TRACKER DE USO SATELITAL**

Lucas Martín Muñoz

Ing. José Relloso
Director

Ing. Paula Rotondo
Co-Directora

Ing. José Argibay
Co-Director

Junio 2009
Centro Atómico Bariloche

Instituto Balseiro
Universidad Nacional de Cuyo
Comisión Nacional de Energía Atómica

*A mis padres,
por enseñarme a volar,
pero no a volar sus vuelos.*

Índice de abreviaturas

ST: Star Tracker

ARI: Algoritmo de reconocimiento de imagen

FOV: Field Of View o campo visual

DN: Digital number (intensidad de un pixel)

HDP: Herramienta de prueba

FP: Falso Positivo

VN: Verdadero negativo

N_FP: Posición de primer FP en listas.

N_VN: Posición de primer VN en listas.

Índice de contenidos

CAPÍTULO 1: INTRODUCCIÓN	- 5 -
1.1. SENSORES DE ORIENTACIÓN DE USO SATELITAL	- 5 -
1.2. SENSORES ESTELARES	- 7 -
1.2. SOFTWARE DE RECONOCIMIENTO DE ORIENTACIÓN	- 9 -
1.4. ORGANIZACIÓN DEL INFORME	- 11 -
CAPÍTULO 2: ALGORITMO DE RECONOCIMIENTO DE IMAGEN (ARI)	- 13 -
2.1. GENERALIDADES SOBRE LAS IMÁGENES ESTELARES	- 13 -
2.2. GENERALIDADES SOBRE EL ARI	- 16 -
2.3. SECUENCIA REALIZADA POR EL ARI 1	- 18 -
2.4. IMÁGENES ESTELARES DE PRUEBA	- 19 -
CAPÍTULO 3: ENSAYOS CON IMÁGENES ESTÁTICAS	- 20 -
3.1. EVALUACIÓN DEL ARI MEDIANTE INSPECCIÓN VISUAL	- 20 -
3.1.1. Descripción de las observaciones	- 20 -
3.1.2. Resultados y discusión del método	- 21 -
3.2. DESARROLLO DE HERRAMIENTA DE PRUEBA (HDP)	- 22 -
3.2.1. Concepto	- 22 -
3.2.2. Descripción de la herramienta	- 23 -
3.2.3. Validación de la HDP	- 25 -
3.3. MEJORAS Y OPTIMIZACIONES EN EL ARI	- 26 -
3.3.1. Nueva secuencia de confirmación de estrellas	- 26 -
3.3.2. Determinación de $\Delta DN1$ y $\Delta DN2$	- 27 -
3.3.3. Otros cambios realizados en el ARI	- 27 -
3.4. PERFORMANCE DEL ARI EN FOTOS REALES	- 28 -
3.4.1. Histogramas de ARI 1 y ARI 2	- 28 -
3.4.2. Visualización de FPs y VNs con el ARI 2	- 30 -
3.5. CONCLUSIONES DE ENSAYOS CON FOTOS ESTÁTICAS	- 32 -
3.5.1. Funcionamiento de la HDP	- 32 -
3.5.2. ARI 1 y ARI 2	- 32 -
CAPÍTULO 4: ENSAYOS CON IMÁGENES DE ESTRELLAS EN MOVIMIENTO	- 34 -
4.1. GENERACIÓN DE IMÁGENES SIMULADAS	- 35 -
4.1.1. Modelo de estrellas estáticas y en movimiento	- 35 -
4.1.2. Aplicación del modelo	- 38 -
4.1.3. Generación de imágenes con estrellas uniformemente distribuidas	- 39 -
4.1.4. Generación de imágenes con simpix y autosimpix	- 41 -
4.2. PRIMERAS PRUEBAS DE RECONOCIMIENTO	- 44 -
4.2.1. Problema de la detección múltiple	- 44 -
4.2.2. Solución al problema	- 46 -
4.3. VALIDACIÓN Y MEDICIÓN DE PERFORMANCE DEL ARI	- 47 -
4.3.1. Medición de la performance del ARI para una velocidad dada	- 47 -
4.3.2. Resultados y discusión de evaluación de performance	- 48 -
4.3.3. Probabilidad de reconocimiento en hemisferio sur y hemisferio norte	- 52 -
DIAGRAMA DE GANTT DEL PROYECTO INTEGRADOR	- 55 -
CONCLUSIONES	- 56 -
ANEXOS	- 57 -

APÉNDICE I: PROGRAMAS UTILIZADOS	- 57 -
APÉNDICE II: APLICACIONES DESARROLLADAS	- 62 -
APÉNDICE III: CATÁLOGOS ESTELARES	- 65 -
APÉNDICE IV: SISTEMAS DE COORDENADAS	- 66 -
REFERENCIAS	- 68 -
ÍNDICE DE FIGURAS	- 69 -
AGRADECIMIENTOS	- 70 -

Resumen

La determinación de orientación satelital en la actualidad se lleva a cabo, entre otros métodos, por sensores estelares de orientación (Star Trackers). El principio de funcionamiento de dichos sensores se basa en la determinación de puntos brillantes en una imagen estelar e identificación de estrellas a través de la comparación con un catálogo de estelar. La etapa inicial del algoritmo de reconocimiento (detección de puntos brillantes) es crítica, ya que de esto depende el éxito del reconocimiento de orientación.

En esta tesis se trabajó con el algoritmo de reconocimiento de imagen (ARI) de un proyecto de star tracker comercial. Se realizaron extensivas pruebas con imágenes reales estáticas a los fines de mejorar la estrategia de búsqueda de estrellas y de validar la rutina. En una segunda etapa se usaron imágenes simuladas en movimiento para comprobar el funcionamiento del algoritmo en condiciones más adversas. Se validó la rutina verificando el buen funcionamiento de la misma sobre una gran cantidad de fotos. Se determinó la performance del algoritmo en función de la velocidad de giro del satélite.

Durante el transcurso del proyecto fue necesaria la realización de herramientas computacionales en lenguaje C para ejecutar dichas pruebas y validaciones.

Palabras claves: Star tracker, sensor estelar, imagen estelares, control de actitud, reconocimiento de imagen.

ALGORITHMS DEVELOPMENT AND TESTING FOR SATELLITE STAR TRACKER

Abstract

Star trackers are widely used as satellite attitude sensors. These sensors take pictures and determine the bright spots, which are then compared with a star catalogue to identify the star images. Finally, these identified stars are used to obtain the attitude of the spacecraft. The initial spot-detection is critical, since the attitude determined by the sensor is highly dependant on an accurate determination of bright spots.

This thesis presents an analysis and proposes an improvement of a recognition algorithm for a commercial star tracker project. Several tests have been made with real and simulated pictures to improve the spot-detection strategy, and to validate the improvements made to the initial algorithm. Additionally, simulated images of moving stars, due to the rotation of the spacecraft, were analyzed. These pictures present more adverse conditions for the star detection algorithm.

A variety of software tools were created on C language to implement all the tasks mentioned.

Keywords: Star tracker, star image, attitude dynamics and control, image recognition.

Capítulo 1: Introducción

1.1. Sensores de orientación de uso satelital

En el campo del diseño de satélites, la disciplina de guiado y control se encarga de determinar y controlar la posición y orientación del vehículo espacial. En este marco, el control de actitud (orientación) determina y manipula la orientación de la nave en algún sistema de referencia inercial. Controlar la actitud de un vehículo requiere (a) sensores para determinar la actitud, (b) actuadores para aplicar torques que reorienten la nave y (c) algoritmos para comandar los actuadores. La matemática de los aspectos básicos en este tema está muy desarrollada, pero en lo que respecta a dispositivos (sensores y actuadores), hay mucha actividad debido a que los avances tecnológicos incentivan mejoras continuas en las prestaciones [1].

En los últimos 50 años de la historia de las misiones espaciales, hubo experimentación extensiva en hardware satelital y el desarrollo aun continúa. A continuación se presentan las características principales de una variedad de sensores [2].

Sensores solares

El sol es una fuente de radiación relativamente pequeña, intensa e inconfundible. Los sensores solares determinan la orientación del satélite mediante la medición del vector sol en el sistema de referencia del satélite. Podemos distinguir entre sensores solares gruesos y finos. Los primeros consisten en celdas solares que entregan una señal analógica en función del ángulo del sol relativo a las mismas. Así, el vector sol se determina a partir de las mediciones de muchas celdas solares que apuntan a distintitos ángulos. Los sensores solares finos toman fotos con un campo visual (FOV) grande (del orden de 120°) para seguir al sol en las imágenes. Llegan a precisiones del orden de algunos arcosegundos. Los sensores solares son confiables, pequeños y de bajo consumo eléctrico. Como desventaja, no funcionan cuando el satélite está eclipsado por la tierra, además, no determinan completamente la actitud porque solo determinan la dirección del sol en relación al satélite, y no pueden conocer la rotación del satélite al rededor del eje que une a los dos cuerpos.

Sensores de horizonte

Para satélites de órbita cercana, la tierra es el segundo objeto celeste más brillante y cubre el ~40% del cielo. Los sensores de horizonte detectan la discontinuidad térmica entre la tierra y el espacio. Consisten de un mecanismo de escaneo, un sistema óptico, un detector de radiación y una electrónica de procesamiento de señal. No son muy precisos por la mala definición del horizonte terrestre. Como ventaja, son robustos ya que detectan el horizonte de una forma directa. Una contra es que necesitan explorar el cielo, ya sea usando el movimiento del satélite o usando plataformas móviles. Por otra parte, al igual que los sensores solares, no determinan completamente la orientación porque no pueden dar el ángulo de rotación del vehículo espacial respecto del vector tierra.

Magnetómetros

Proveen la magnitud y dirección del campo magnético terrestre. Los más simples consisten de tres bobinas ubicadas a 90° (magnetómetros de inducción) y una electrónica de procesamiento de señal. Son confiables, livianos, y tienen un bajo requerimiento de potencia. Como desventaja, no son muy precisos, tienen un error del orden de 0.5°. Por otra parte, como el campo magnético terrestre disminuye con $1/r^3$, y estos sensores son vulnerables al campo magnético generado por la electrónica del satélite, el uso de los mismos se restringe a los satélites de órbita baja. Igual que los sensores ya mencionados, no determinan completamente la orientación.

Giróscopos

A diferencia de los mencionados hasta ahora, son sensores relativos (necesitan la orientación inicial para funcionar). Detectan rotación en los tres ejes. Clásicamente, se constituyen de una masa rotante, pero existen giróscopos láser y de resonancia. Los de tipo láser utilizan luz coherente reflejada en un camino cerrado. Los de resonancia utilizan un cristal con forma de copa. La dirección de la oscilación se mantiene fija al sistema de referencia inercial, así que se mide esta orientación relativa al vehículo para determinar la actitud.

Sensores estelares (*star trackers*)

Los sensores estelares miden coordenadas de estrellas en el sistema de referencia del satélite y proveen información de orientación al comparar las coordenadas medidas con las direcciones estelares obtenidas de un catálogo estelar. La medición de las coordenadas de estrellas en el sistema de referencia del satélite tiene como primera etapa la toma de una foto estelar con ángulo de campo visual (field of view) del orden de 20°. Luego se eligen las estrellas mas brillantes de la foto para identificar la imagen estelar en el catálogo. En general, los star trackers son los mas precisos, llegando a precisiones inferiores al arcosegundo en los mejores casos. Además son versátiles, por lo que se adaptan a diversas misiones espaciales. Estas fuertes ventajas tienen un costo: los star trackers, en relación a otros sensores, son pesados, caros, requieren mucho software y consumen mucha potencia. Otra desventaja es que son susceptibles a errores debido a la sensibilidad a la luz del sol reflejada por el satélite o por los gases de escape de los propulsores. También hay otras fuentes de confusión para el algoritmo de identificación: planetas, cometas, otros satélites cercanos y cualquier objeto extraño que se presente el FOV. En la siguiente sección se explora el tema de sensores estelares.

1.2. Sensores estelares

Hay varios modelos de star trackers: scanners estelares (star scanners), que usan la rotación de la nave para detectar estrellas, sensores de cabeza móvil (gimbals star trackers), los que están montados en una plataforma móvil pudiendo barrer diferentes partes de la esfera celeste, y los star trackers de cabeza fija (fixed head star trackers), con un FOV limitado y fijo respecto del satélite. A continuación nos centraremos en los sensores de cabeza fija, basados en detectores de imagen digitales de tipo CCD (charged-coupled device) o APS (active pixel sensor), siendo estos los mas utilizados en la actualidad. Las figura 1.1 y figura 1.2 ilustran un típico star tracker comercial, que posee un sensor óptico CCD de 1024x1024 píxeles, su masa es de 3kg y el consumo eléctrico es de 10W.

Los componentes básicos de un sensor estelar son (a) *protector solar* (baffle), (b) *sistema óptico*, (c) *módulo de detección*, (d) *electrónica de procesamiento* y (e) *carcasa* (ver figura 1.1 y figura 1.2). Se describen brevemente cada una de estas partes:

- a) La luz intensa es un problema para los sensores estelares, por lo que un *protector solar* (baffle) cuidadosamente diseñado es critico para la buena performance del

sensor. Incluso con protector solar, los star trackers son típicamente inoperables con el sol a menos de $\sim 45^\circ$ del FOV.

- b) El *sistema óptico* consiste de lentes que proyectan el FOV en el plano focal.
- c) El *módulo de detección* contiene al detector CCD o APS. Estos detectores (similares a los utilizados en cámaras fotográficas digitales comerciales) están compuestos por un arreglo de cientos de miles de foto-sensores (valores comunes son 512x512 o 1024x1024) dispuestos en una matriz rectangular paralela al plano focal. Cada uno de los fotosensores individuales es denominado *pixel*. La señal eléctrica de cada pixel (proporcional a la intensidad lumínica recibida) es amplificada y digitalizada (con un ADC o Analogical to Digital Converter). La información entregada por el módulo de detección es un vector con los valores de intensidad o *DNs* de cada pixel.
- d) La *electrónica de procesamiento* toma la imagen digitalizada y mediante el *software de reconocimiento de orientación* (ver sección 1.2) determina la actitud del satélite.
- e) La *carcasa* cubre y protege el sistema óptico, el módulo de detección y la electrónica de procesamiento. Se debe tener especial cuidado en el diseño del conjunto y su acople al satélite debido a las condiciones extremas que el ambiente espacial impone (gran amplitud de temperatura y radiación intensa).

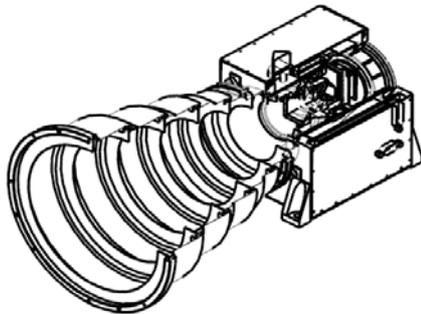


figura 1.1: Vista esquemática en corte (a) e imagen (b) de un star tracker comercial.

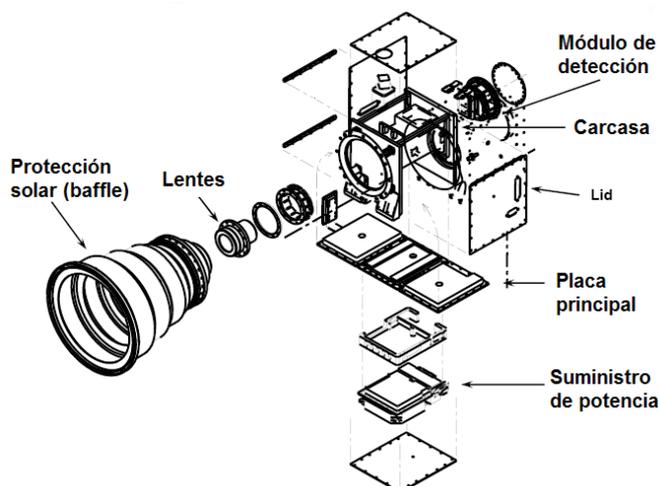


figura 1.2: Vista explosiva de un star tracker.

Para dar una idea del estado del arte, se pueden mencionar star trackers con múltiples cabezas ópticas (hasta 4) para asegurar una visión ininterrumpida del campo estelar, y star trackers con precisiones del orden de decimas de arcosegundo. Las principales ventajas que pretenden obtenerse con los últimos desarrollos son la *autonomía para el vehículo portador*, el *análisis de actitud en tiempo real* y la *eliminación de otros sensores* [3].

Un ST comercial tiene un precio muy alto porque hay pocos fabricantes en el mundo que tienen diseños calificados en vuelo. Por otra parte, un satélite usa dos de ellos normalmente. Hay una clara tendencia por parte de los diseñadores de satélites de usar sensores estelares como dispositivo principal de orientación, lo que constituye una oportunidad para los fabricantes de Star Trackers.

El desarrollo de componentes satelitales de INVAP es parte de un plan estratégico que incluye el uso de dichos dispositivos en los satélites propios así como la inserción de los mismos en el mercado internacional. El proyecto de star tracker de INVAP está pensado para ser estrenado en el primer satélite geostacionario de comunicaciones argentino.

1.2. Software de reconocimiento de orientación

La descripción detallada de los algoritmos que componen el software del ST merece por lo menos un capítulo aparte. Sin embargo, a continuación se sintetizan los

principios de funcionamiento de los algoritmos, dejando de lado muchos aspectos necesarios para una comprensión acabada del software.

En primer lugar, el ST tiene la posibilidad de funcionar en dos modos: modo de *reconocimiento inicial* y modo *seguimiento*. En el reconocimiento inicial, se parte de una imagen estelar completa y se determina la actitud sin ninguna otra información. En el modo de seguimiento, se conoce la actitud aproximada (debido a reconocimientos previos) y se usan las porciones de la imagen en las que se sabe que hay estrellas. Con dos mediciones de actitud separadas por un intervalo de tiempo pequeño, es posible calcular la velocidad angular del satélite y realizar un análisis predictivo de la posición de las estrellas en la próxima fotografía. Así las cosas, el modo de seguimiento permite un reconocimiento eficiente y de bajo costo computacional.

El reconocimiento completo de la actitud en modo de reconocimiento inicial se puede segmentar en (a) reconocimiento de imagen, (b) primera identificación y (c) verificación de tríos y selección final.

a. Reconocimiento de imagen

El algoritmo de reconocimiento de imagen es la parte del software del Star Tracker que se encarga de adquirir la foto estelar y reconocer en ella las estrellas presentes, reportando la posición en la foto y la intensidad lumínica de cada una. La posición de las estrellas en la foto se mide con un sistema de coordenadas cartesiano.

Paso seguido, el algoritmo ordena las estrellas encontradas en orden decreciente de intensidad lumínica y entrega una lista con las primeras N_f estrellas de dicha lista.

b. Primera identificación

En primer lugar, se menciona que el brillo de las estrellas no se usa en ninguna etapa de identificación de estrellas en el catálogo. Esto es así para no depender de la buena calibración del detector CCD o APS, entre otras cosas (ver [3]).

Al comenzar con la identificación de las estrellas visibles, el primer paso es tomar cada una de las estrellas (de las N_f más brillantes) y medir la distancia angular¹ respecto

¹ La distancia angular entre dos estrellas en la imagen es el ángulo entre las dos estrellas que tiene su vértice en el sensor solar. Teniendo en cuenta que el FOV tiene un ángulo relativamente pequeño, este se determina calculando la distancia en píxeles entre estas estrellas y multiplicándola por un factor.

de cada una de las estrellas restantes en la foto. A continuación se forman tríos de estrellas, se toman cada uno de los tríos y se buscan coincidencias en el catálogo estelar (ver catálogos en Apéndice III). Dado que las distancias fueron calculadas con una tolerancia, el resultado de esta etapa consiste de varios posibles sky#² asociados a cada estrella de la foto.

c. Verificación de tríos y selección final

En esta etapa se forman nuevamente tríos, esta vez es con todos los sky# candidatos de cada estrella de la imagen. Luego, las distancias en la foto de cada trío son comparadas con las distancias teóricas entre esos sky#, según catálogo. Si la comparación de cada trío coincide, los sky# del trío reciben una confirmación. Finalmente, a las estrellas se les asigna el sky# que mas confirmaciones tiene.

Para conocer más detalles acerca de los algoritmos de primera identificación y de verificación de tríos y selección final, dirigirse a (3).

Vale la pena mencionar el grado de avance del desarrollo de software del ST en cuestión: hasta el momento, el esfuerzo se concentró en la parte del software correspondiente al modo de *reconocimiento inicial*. Se encuentran en etapas avanzadas sus tres segmentos a, b y c. Resta validar gran parte del código. Al comenzar este trabajo, existía una versión preliminar del algoritmo de reconocimiento de imagen.

Esta tesis se enfocó en el algoritmo de reconocimiento de imagen (en adelante se usará la abreviación ARI), etapa (a) del reconocimiento de actitud.

1.4. Organización del informe

La meta general del trabajo consistió en avanzar lo máximo posible con el desarrollo del software del ST, en lo relativo al ARI. El trabajo realizado se puede seccionar en dos etapas: en la primera se realizaron pruebas con imágenes de estrellas estáticas y en la segunda se hicieron con imágenes con estrellas en movimiento.

² Número de identificación de cada estrella en el catálogo.

En la primera etapa, con imágenes estáticas, se utilizó un conjunto de fotos tomadas a cielo abierto por el sensor estelar en desarrollo de INVAP. Los objetivos de esta fase son (1) reducir al mínimo las fallas del algoritmo, (2) lograr que el ARI detecte la mayor cantidad de estrellas posibles y (3) validar el algoritmo probándolo en una gran cantidad de fotos. Para esto se realizaron primeramente verificaciones visuales del funcionamiento del ARI. Dichas verificaciones visuales consisten principalmente en observar en un grupo de fotos si los centros delatados por el ARI son realmente estrellas. Las observaciones también son útiles para comprender cómo es la distribución de píxeles altos en una estrella y para visualizar el funcionamiento del ARI en diferentes tipos y tamaños de estrella. Esto es esencial para proponer soluciones y mejoras al ARI. El proceso de observación es iterativo, ya que luego de cada modificación del ARI se lo debe evaluar nuevamente.

Para validar el ARI y así garantizar el buen funcionamiento del mismo con fotos estáticas se requiere probarlo sobre una gran cantidad de fotos (del orden de cientos). De esta forma se obtiene una buena estadística. Como esto es irrealizable con el método de revisión visual, surgió la realización de una herramienta de prueba. Esta tiene como entradas el grupo de fotos a evaluar las salidas son: fallas y cantidad de estrellas encontradas en cada foto.

Es deseable que el ST admita una velocidad de rotación del satélite lo más alta posible. La velocidad de rotación genera un desplazamiento lineal del centro de las estrellas sobre el plano de la imagen. En la segunda etapa de la validación del ARI se utilizaron imágenes de estrellas movidas simuladas (generadas con algoritmos) con el fin de validar y caracterizar el ARI en uno de los casos de operación más intensivos. Para comenzar con esta etapa, se utilizó el ARI resultante de la primera parte (pruebas con fotos estáticas) y en el transcurso fue modificado pertinentemente. Como objetivos de la segunda etapa mencionamos (1) detectar y corregir fallas particulares de las estrellas en movimiento y (2) caracterizar el funcionamiento del ST en función de la velocidad de rotación.

Capítulo 2: Algoritmo de reconocimiento de imagen (ARI)

El presente capítulo presenta el funcionamiento del ARI inicial (existente al comenzar el proyecto). Para introducir el tema y facilitar la comprensión del algoritmo se exhibe primeramente una descripción de las imágenes estelares.

2.1. Generalidades sobre las imágenes estelares

Las imágenes tienen 512 x 512 píxeles y son almacenadas en la memoria en un vector que contiene la intensidad de cada pixel en el orden que lo indica la figura 2.1. Dicha intensidad lumínica se denomina *digital number* o *DN*. Las imágenes no tienen colores, sino que el vector indica solo escala de grises. Cada pixel adopta un valor entero entre 0 y 1024. Por otra parte, en el código del ARI utiliza un sistema de coordenadas que se observa en la misma figura. De esta manera, por mencionar ejemplos, el pixel 0 corresponde a las coordenadas $\{x = 0, y = 0\}$ y el último pixel (número $512^2 - 1$) tiene coordenadas $\{x = 511, y = 511\}$.

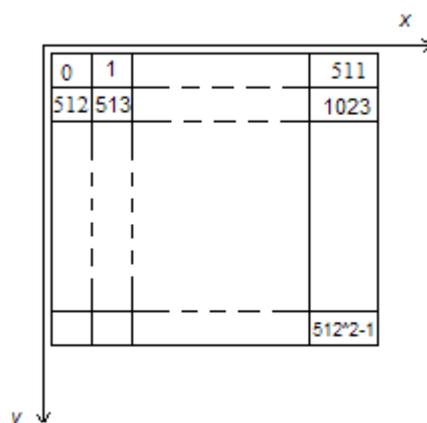


figura 2.1: Sistema de coordenadas y orden de los píxeles en la foto estelar

El nivel de DN del fondo³ varía levemente entre pixel y pixel. A esta variación la denominamos ruido y tiene generalmente un valor de $\pm 1 DN$, es decir que el valor de los píxeles oscila entre tres números enteros. Por otra parte suele variar el valor promedio del fondo entre diferentes partes de la imagen. El resultado son rampas de intensidad con un ruido montado.

Las estrellas en las fotos se presentan como pequeños grupos de píxeles de valor alto con respecto al fondo. En la figura 2.2 se muestra un gráfico de barras de una estrella. Por ser relativamente brillante, no resulta difícil detectarla con el ARI. Se observa el ruido de fondo de 3 DN de amplitud.

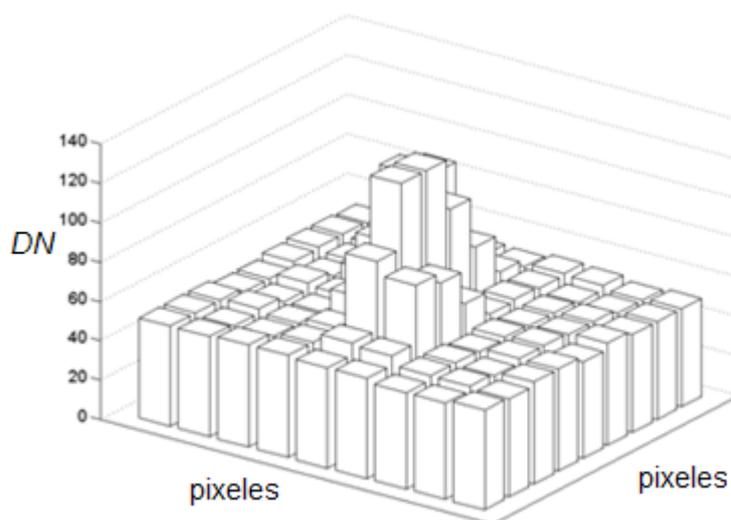


figura 2.2: Gráfico de barras de una estrella relativamente brillante.

Por otra parte, la figura 2.3 presenta el gráfico de barras de una estrella más difícil de detectar, confundible con el ruido de fondo. En lugar de ser redonda es alargada debido a una velocidad angular del satélite de 0,03 rad/s. El pixel más alto tiene 11 DN por encima del fondo.

³ Llamamos fondo a las partes de la foto donde no hay estrellas

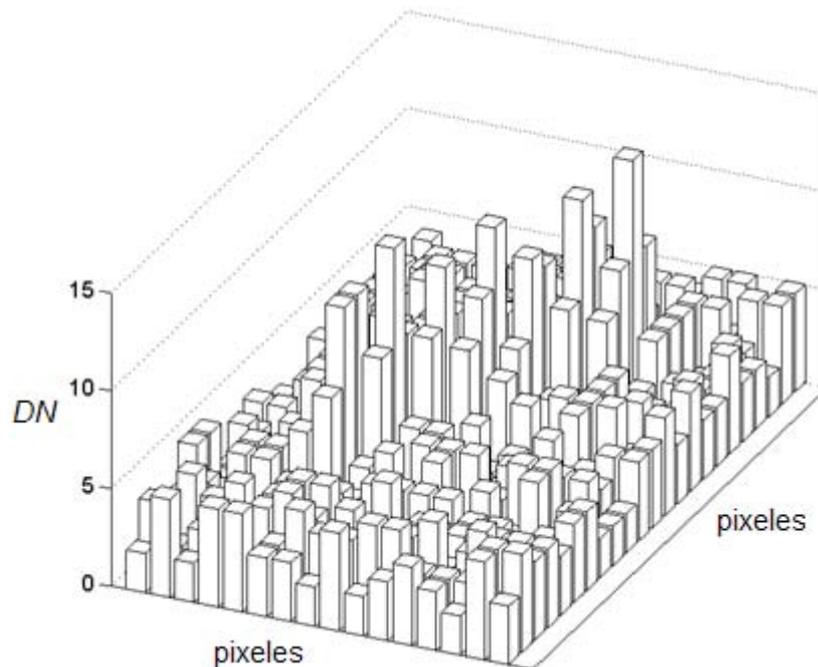


figura 2.3: Gráfico de barras de una estrella pequeña, confundible con el fondo.

La forma de medir el brillo de una estrella en la foto es mediante el cálculo de la suma de los DN s de los píxeles que la componen, restando el valor de fondo (llamamos a esta suma DN_t). Es útil conocer la relación entre ésta dimensión y la *magnitud estelar*, siendo esta última la forma común de medir el brillo de un cuerpo estelar en la astronomía. La relación entre la magnitud (m) y la intensidad lumínica (medida en DN s) de una estrella está dada por

$$m - m_0 = -2,5 \cdot \log_{10} \frac{DN_t}{DN_{t_0}}$$

Donde m_0 y DN_{t_0} son constantes. Se advierte que a mayor brillo, menor es la magnitud. Como datos de referencia mencionamos que una estrella pequeña para el ARI (la menos intensa detectable) tiene alrededor de $m = 5,5$ y $DN_t = 40$, mientras que una estrella significativamente brillante tiene $m = 2$ y $DN_t = 1000$.

Por último, se menciona que la magnitud estelar depende del espectro visual del sensor u ojo que la mide. Por lo cual, una misma estrella puede tener diferentes magnitudes para diferentes sensores. La magnitud del sensor del ST se la define como *magnitud instrumental*.

2.2. Generalidades sobre el ARI

El ARI recorre los píxeles del vector de DN uno a uno, en orden, buscando valores particularmente altos en relación al fondo, al encontrar un *pixel alto* realiza una secuencia de interrogación en los píxeles limítrofes para analizar si se trata de una estrella (la secuencia se describe en la siguiente sección). En caso de no encontrar un pixel alto, actualiza el valor del fondo teniendo en cuenta los últimos píxeles rastreados. En el comienzo de cada línea debe reiniciarse el cálculo del fondo.

Una vez recorridos todos los píxeles de la imagen, se tienen los conjuntos de píxeles altos que contienen estrellas. Con esta información el algoritmo calcula los centroides y el tamaño de cada estrella. Esto se hace tomando un pequeño mapa de píxeles rectangular que contiene a la estrella en cuestión. El tamaño se mide sumando los DN que cada pixel de la estrella tiene por encima del fondo (llamaremos DN_t a los DN totales de la estrella). A su vez, este fondo se calcula como el promedio de los píxeles frontera del mapa de píxeles (no pertenecientes a la estrella). Los parámetros DN_t , coordenada x y coordenada y de una estrella se calculan como:

$$DN_t = \sum_i (DN_i - DN_{fondo})$$

$$x_c = \frac{\sum_i DN_i \cdot x_i}{DN_t}$$

$$y_c = \frac{\sum_i DN_i \cdot y_i}{DN_t}$$

Donde i barre todos los píxeles del rectángulo que no pertenecen a la frontera.

Un último paso en el ARI ordena las estrellas encontradas según DN_t , en forma decreciente.

La información que entrega el ARI es una lista para cada foto con las columnas:

número de estrella	coordenada x (píxeles)	Coordenada y (píxeles)	DN
--------------------	--------------------------	--------------------------	------

El ARI entrega solo las primeras N_f estrellas (N_f todavía no está definido, pero adoptará un valor cercano a 12).

Hay dos tipos de fallas importantes:

- **Falsos positivos o FPs:** Centros reportados por el ARI que no son estrellas realmente.
- **Verdaderos negativos VNs:** Estrellas que no son reportadas por el ARI.

Es de interés que el ARI entregue una gran cantidad de estrellas correctamente, es decir, sin FPs ni VNs. De forma equivalente, se desea que los primeros FPs y VNs sean centros lo mas chicos posibles (pequeño DN_t).

Se destaca que resulta mucho mas grave la presencia de FPs que de VNs en el resultado: los VNs degradan la probabilidad de un posterior reconocimiento exitoso, los FPs, además de tener el mismo efecto, inducen un costo computacional no deseado.

Graficando el DN de estrella versus el número de estrella (dentro de la lista de salida del ARI) para una gran cantidad de listas (figura 2.4) se encuentra una relación entre dichos parámetros. Por lo tanto, puede ser usado el dato *número de estrella* en lugar de *DN* a la hora de buscar FPs y VNs para evaluar la calidad del algoritmo.

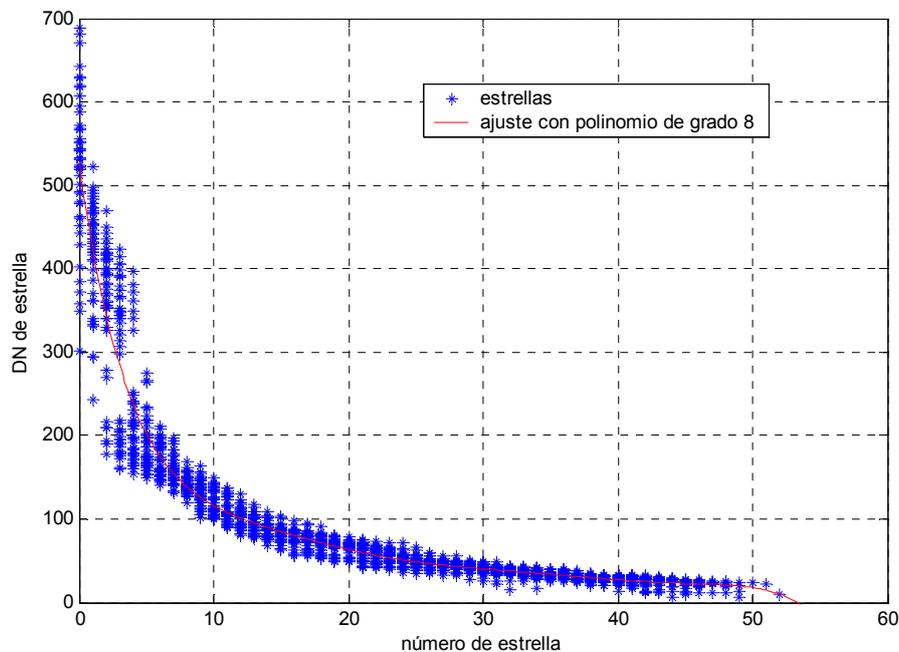


figura 2.4: Relación entre número de estrella en la lista y intensidad en DN.

2.3. Secuencia realizada por el ARI 1

Como ya se mencionó, el ARI recorre los píxeles de la imagen uno a uno. A continuación se explica la secuencia de interrogación efectuada en cada pixel por el ARI inicial (el que se usó al comenzar el trabajo). A este lo llamaremos ARI 1.

La

figura 2.5 esquematiza la rutina cuando la búsqueda se encuentra en el píxel "i". Dentro de la rutina, en caso de confirmar que el pixel actual tiene un valor mayor que el fondo (ΔDN mas alto), se examinan los píxeles adyacentes uno por uno, asignándoles el mismo *id* (número de identificación de centro estelar) a todos los que tienen un valor de mas de ΔDN por encima del fondo. En el caso de que alguno de los píxeles examinados ya pertenezca a una estrella previamente encontrada (ya tiene un *id* asignado), se les concede al resto de los píxeles el mismo *id*. Notar que solo el pixel "i" o el pixel de la derecha pueden estar identificados por el ARI al revisar el renglón de píxeles inmediatamente superior. El conjunto de píxeles con un mismo *id* constituyen un centro estelar. El valor que se adopta para ΔDN en este algoritmo es 5.

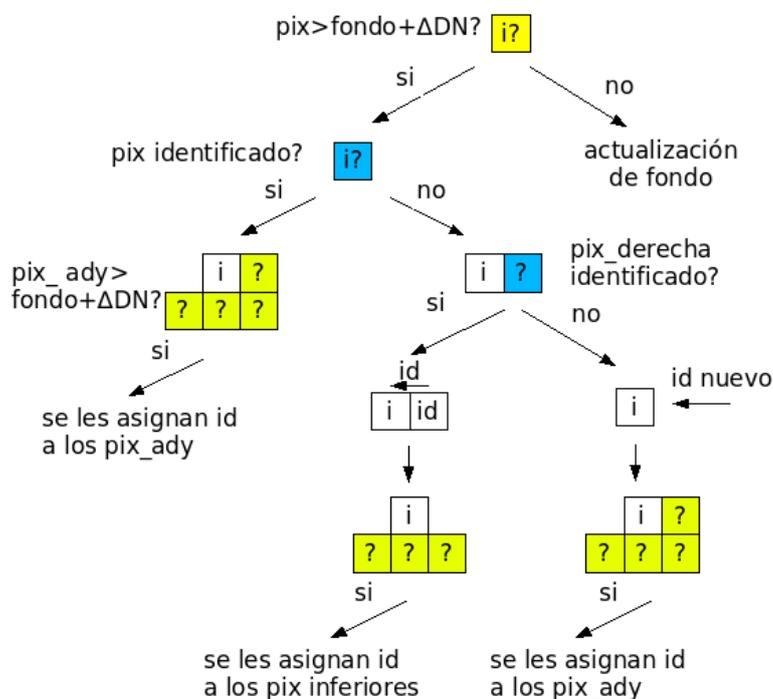


figura 2.5: Secuencia realizada por el ARI inicial al recorrer el vector de DN

En el caso de que el pixel actual no sobrepase al valor de fondo por más de ΔDN , se procede a actualizar el fondo con dicho pixel. Esto se hace calculando un promedio de los últimos 10 píxeles rastreados. Conociendo el fondo viejo (el del pixel “ $i - 1$ ”), no es necesario realizar la suma de los últimos 10 píxeles si no que el fondo nuevo se calcula como:

$$fondo_{nuevo} = \frac{fondo_{viejo} \cdot 10 - DN_{ultimo} + DN_{nuevo}}{10}$$

Donde DN_{ultimo} es la intensidad del primero de los 10 píxeles, valor que no entra en el cálculo del promedio actual.

Un proceso posterior dentro del ARI 1, selecciona únicamente a los centros que tienen 3 o mas píxeles altos.

2.4. Imágenes estelares de prueba

Para validar el ARI, se utilizaron (1) imágenes reales estáticas e (2) imágenes simuladas en movimiento.

1. Para realizar la primera etapa, con imágenes reales quietas, se dispone de un set de 747 fotos estelares tomadas a cielo abierto en una noche sin nubes. Fueron sacadas automáticamente cada 45 segundos con la cámara fija a la tierra. Las imágenes tienen solamente escala de grises y su tamaño es de 512x512 píxeles. Algunas de las fotos son defectuosas, por los que se usan solo 450 del total.
2. Las imágenes simuladas son realizadas por computadora. Un programa genera un vector imagen utilizando un modelo matemático para determinar la distribución e intensidad de los píxeles altos que forman las estrellas. Se puede incluir fácilmente el ruido de fondo aleatorio y anomalías como variaciones lineales del valor del fondo, para probar al ARI con fotos lo mas reales posibles y/o con condiciones extremas.

Capítulo 3: Ensayos con imágenes estáticas

Para utilizar las fotografías en la evaluación del algoritmo se comenzó realizando inspecciones visuales de los resultados del ARI. Luego, en búsqueda de un método más eficiente para la revisión de grandes cantidades de imágenes, se realizó una *herramienta de prueba* de fotos reales. Basándose en estas pruebas y ensayos, se realizaron mejoras al ARI. Con la herramienta de prueba se midió a performance del ARI en fotos estáticas.

3.1. Evaluación del ARI mediante Inspección visual

La forma mas simple que se puede pensar para verificar el funcionamiento exitoso del ARI es observar en la foto estelar, las posiciones en las que el ARI delata estrellas. Con el fin de que la tarea sea ágil, se utilizan herramientas de manipulación de imágenes realizadas específicamente para el desarrollo del ST.

3.1.1. Descripción de las observaciones

El método consistió en verificar de forma visual (1) la existencia de estrellas en las coordenadas en las que el ARI lo predice y (2) la detección por parte del ARI de las estrellas que se observan en la foto. La no verificación de (1) y/o (2) indica la presencia de FPs y/o VNs respectivamente. Por otra parte, es de interés conocer las distribuciones de *DN* que adoptan las estrellas según su magnitud.

En este afán resulta muy útil poder mirar zonas de la imagen con gran aumento (de forma de ver los valores de cada pixel) y poder marcar automáticamente los centros delatados por el ARI en la imagen estelar. Por eso fueron de utilidad los programas *ciclograf* y *visor*, ambos realizados por INVAP. En el Apéndice I se encuentra una descripción detallada de estos programas. En resumen, *ciclograf* despliega los resultados tanto en forma numérica (lista de coordenadas y magnitudes para cada estrella) como en forma gráfica (identificación de centros en la fotografía). Por otra parte, *visor* permite observar las fotografías con una ampliación a nivel de pixel, esto es, se despliega una

porción de la imagen de tamaño variable donde se representa la intensidad de cada pixel con escala de colores y numéricamente.

Con estas herramientas se pueden identificar todos los *FPs* y *VNs* y observarlos detenidamente para detectar el motivo de falla del ARI.

Una mejora importante del método consistió en restarle una imagen a las fotos estelares para una observación mas clara (ver apéndice I, programa *visor*). Como imagen de resta se usó en cada caso una foto consecutiva en la serie de fotografías reales. Con esto se logra eliminar muchos defectos que están repetidos en las dos imágenes. La imagen resultante tiene un valor de fondo cercano a 0 (el fondo tiene un poco de ruido) y posee las mismas estrellas que la imagen inicial. Los defectos más fastidiosos que se eliminan de esta forma son píxeles muy altos con respecto al fondo, no pertenecientes a ninguna estrella. Los mismos corresponden a un mal funcionamiento de algunos píxeles del detector óptico. Dichos píxeles altos no se pueden diferenciar de las estrellas cuando se observa la imagen completa, por lo que poder eliminarlos es una gran ventaja a la hora de observar imágenes.

El método de detección de *FP's* y *VN's* por inspección visual es lento y tedioso, y tiene como desventaja el hecho de que deben realizarse nuevas inspecciones cada vez que se modifica el ARI.

3.1.2. Resultados y discusión del método

Se observaron con el método de inspección visual algunas decenas de fotos de la serie de fotos reales.

No resultó ser un método rápido como se esperaba: en un principio se deseaba poder encontrar fallas del ARI rápidamente, recorriendo una gran serie de imágenes en tiempo razonable. El mayor inconveniente en este sentido es el hecho de que es necesario observar cada estrella detenidamente con *visor*. Así, se puede aprender mucho del funcionamiento del ARI con distintos tamaños y tipos de estrella, pero no es posible revisar cientos de fotos, lo que sería necesario para validar correctamente el ARI.

Se obtuvieron importantes resultados cualitativos, entre los cuales mencionamos que el ARI falla inevitablemente para estrellas suficientemente pequeñas, por más que sean fácilmente detectadas con la observación directa de una foto. Teniendo en mente el diagrama de la

figura 2.5, para que el ARI detecte estrellas muy pequeñas es necesario bajar el umbral ΔDN . Bajando mucho este valor se corre el riesgo de que el ruido de fondo o pequeñas imperfecciones en la imagen sean consideradas estrellas. Así las cosas, existe un compromiso entre la detección de FPs (ΔDN bajo) y de VNs (ΔDN alto).

En cuanto a estrellas grandes, como es de esperar, no hay ningún problema con la detección de las mismas. Por esto, las mejoras en el ARI apuntan a encontrar la mayor cantidad de estrellas posibles (lo que es equivalente a encontrar estrellas lo mas pequeñas posible) sin fallas. Como ya se expresó en la sección 2.2, el reconocimiento de orientación posterior al ARI sólo usa las N_f primeras estrellas que entrega el ARI, de modo que lo deseable es que no existan FPs ni VNs entre esos centros.

Otra desventaja del método de inspección visual es que en cada modificación del ARI se deben realizar nuevamente las observaciones, cosa que no ocurre con el método que se explyea en la siguiente sección.

3.2. Desarrollo de Herramienta de prueba (HDP)

3.2.1. Concepto

Se aprovecha el hecho de que la serie de 450 fotos (reales estáticas) fueron sacadas con un espacio de tiempo constante entre toma y toma. Este intervalo de tiempo se corresponde con una variación de orientación (por la rotación de la tierra) de aproximadamente $0,2^\circ$, que significa un corrimiento de aproximadamente $0,2^\circ/15^\circ \cdot 512 = 6,8$ píxeles. Dadas dos fotos consecutivas que llamaremos A y B, se realizan imágenes que resultan de calcular el módulo de la resta de los DN píxel a píxel. El resultado es una foto en la que aparecen tanto las estrellas de la foto A como las de la foto B y un fondo cercano a 0 dado que la intensidad del fondo no varía mucho entre foto y foto. Un ejemplo de estas imágenes resultantes se ilustra en la figura 3.1. Se aplica el ARI a imágenes de este tipo y solo se confirman los pares de estrellas encontrados, lo que garantiza que se encuentren todas las estrellas de la imagen. Estas estrellas se comparan con los resultados de la aplicación del ARI a las imágenes originales.

3.2.2. Descripción de la herramienta

El proceso se realiza utilizando las fotos de la serie de a dos consecutivas, es decir, tomando la primera con la segunda, la segunda con la tercera, etc. Así, se tienen pares de centros para cada estrella válida y eventualmente centros no acompañados que no corresponden a estrellas. Todos los pares de centros tienen igual distancia vectorial dentro de cierto error. Dicha distancia se determina a partir de un promedio de pares de estrellas en un grupo de fotos. Se determinan las estrellas de cada foto mediante un algoritmo que confirma como estrellas a los centros que pertenecen a un par (en realidad, uno solo de cada par, el de menor coordenada y , que es el que corresponde al número de foto). De este proceso surge una lista de estrellas para cada foto. Dichas listas son guardadas y utilizadas para ser contrastadas (mediante otro algoritmo) con las listas producidas por el ARI que se desea probar. El algoritmo que compara las listas determina los FPs y VNs de cada foto. Notar que una vez producidas las listas de estrellas para un set de fotos, las mismas se pueden usar más de una vez con distintos ARIs.



figura 3.1: Foto calculada como el módulo de la resta de dos imágenes.

En la figura 3.2 se presenta un diagrama del funcionamiento de la HDP. En él, se indican los algoritmos con cuadros. Como se puede ver en la figura, se deben utilizar tres algoritmos para generar las listas de estrellas de cada foto:

1. **Generación de fotos dobles:** Crea fotos calculadas como el módulo de la resta de las imágenes originales. Para esto abre dos archivos de imágenes, y dentro de un bucle calcula píxel por píxel el módulo de la resta y guarda en un tercer archivo.
2. **ARI (sensible):** En segundo lugar se utiliza el ARI para detectar centros de las fotos previamente creadas. El código del ARI que se usa para este paso sufre un cambio respecto del ARI del ST. Éste debe ser muy sensible para detectar hasta los centros menos intensos. La sensibilidad del ARI se acomoda variando el umbral de intensidad que tiene que superar un píxel para ser considerado parte de una estrella (ΔDN). Notar que de esta forma se encuentran muchos FPs, que no representan un problema ya que son discriminados en el paso 3.
3. **Reconocimiento de pares de centros:** Este algoritmo reconoce los pares de centros y genera listas ingresando solo uno de los centros de cada par (el que está mas arriba del par, es decir, el de menor coordenada y). Esto se hace recorriendo de a uno los centros de las listas generadas en el punto 2 y buscando para cada uno de estos, otro que esté a una distancia vectorial conocida, dentro la tolerancia predefinida.

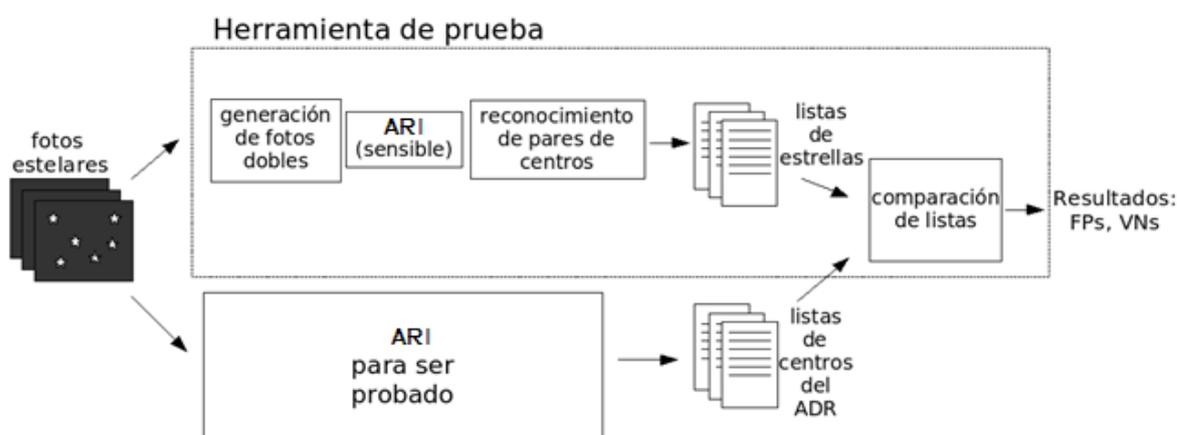


figura 3.2: Diagrama de algoritmos y archivos usados.

Como último paso de la prueba, un algoritmo (**comparación de listas**) compara las listas de estrellas (provenientes de los algoritmos recién explicados) con las listas de centros de un conjunto de fotos deseado (provenientes del ARI) y entrega como resultados:

- De cada foto del conjunto: el FP y el VN de mayor intensidad, incluyendo
→ DNs de los mismos.

- Posición de los mismos dentro de la lista de cada foto, a estos valores se los llamará N_FP y N_VN respectivamente.
- Ventana de 8x8 píxeles centrada en dichos FP y VN.
- Del conjunto de fotos: histograma de primeros FPs de cada foto, histograma de primeros VNs de cada foto, cada uno con media y desviación estándar.

En esta rutina se omiten los FPs y VNs que están cerca del borde de la foto (a una distancia menor que la de los pares de centros) ya que uno de los centros del par puede no estar dentro de la imagen, y eso daría falsos reportes de FPs o VNs. Los códigos de las partes de la HDP se nombran en el apéndice II.

Se utilizan diferentes segmentos de la información descrita arriba para (1) validar y (2) mejorar el ARI, y para (3) validar la herramienta de prueba:

Para la validación de ARI se usa la media de los histogramas de FPs y VNs, que es equivalente al promedio de los N_FP y N_VN del conjunto de fotos.

Para analizar y clasificar los FPs y VNs se usan las ventanas de píxeles de los FPs y VNs. Esto tiene como finalidad por un lado detectar fallas en el ARI y plantear estrategias para la mejora del mismo y por otro lado validar la HDP al verificar que las ventanas reportadas realmente sean FPs y VNs.

3.2.3. Validación de la HDP

Se utilizó una serie de fotos simuladas con un pequeño corrimiento entre foto y foto para validar la HDP. En la sección 4.1 se expone el método de generación de estas fotos.

Dado que el algoritmo de reconocimiento dentro de la HDP (el que reconoce centro en las fotos dobles) es más sensible que el ARI que se desea probar, las listas de estrellas producidas por la herramienta resultan más largas que las listas de centros creadas por el ARI. Por esto, es de esperar que se detecten VNs de pequeño tamaño y no FPs.

Se probaron 14 fotos. La HDP no encontró ningún FP y el promedio de la posición de los primeros VN fue de 52.8, siendo la menor de 49. Con este resultado queda demostrado el buen funcionamiento del método de la herramienta de prueba.

3.3. Mejoras y optimizaciones en el ARI

3.3.1. Nueva secuencia de confirmación de estrellas

Basándose en los resultados cualitativos de las inspecciones visuales y en los resultados de la HDP se modificó la secuencia de confirmación de estrellas del ARI. Llamaremos ARI 2 al algoritmo modificado. En este algoritmo la secuencia efectuada al recorrer el vector de DN es la ilustrada en la figura 3.3. El primer paso al indagar un pixel para ver si éste forma parte de una estrella es analizar si tiene un valor por encima de $\Delta DN1$ respecto del fondo. En caso de confirmarse, se verifica que el promedio de los cuatro píxeles adyacentes sea mayor que el fondo, esta vez con un umbral de $\Delta DN2$. Por último se examina si alguno de los píxeles que ha sido previamente identificado forma parte de otro centro. Si esto es positivo, se le asignan los nuevos píxeles al centro ya existente, si es negativo, se crea un nuevo *id* para los cinco píxeles que forman una nueva estrella.

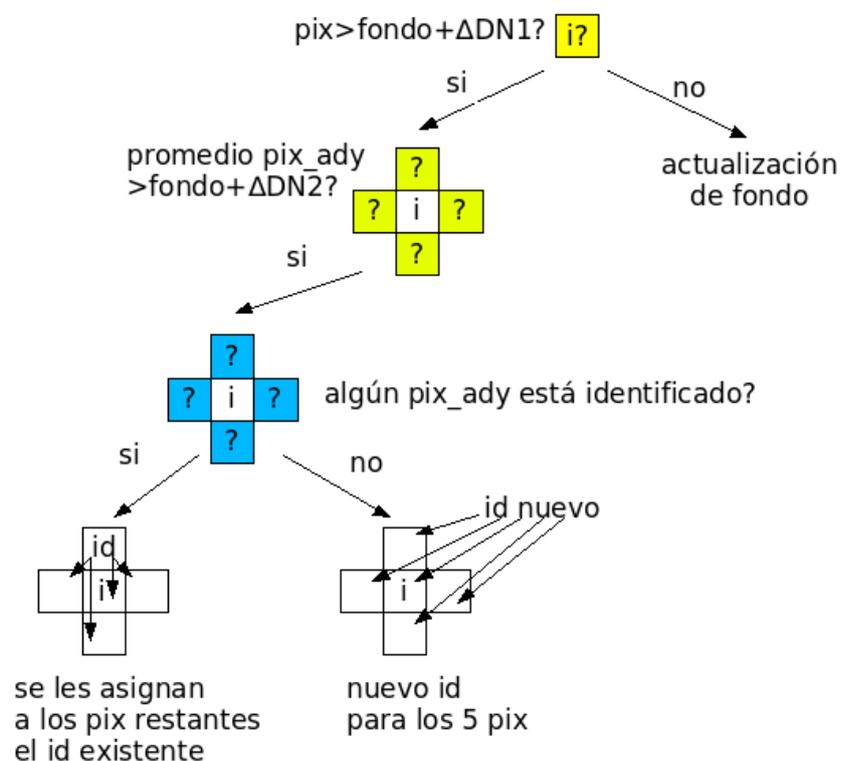


figura 3.3: Esquema de la secuencia realizada por el ARI modificado.

Con este cambio, al promediar los valores de cuatro píxeles se busca reducir el efecto del ruido de fondo, de forma de que dicho ruido no sea confundido con estrellas.

3.3.2. Determinación de $\Delta DN1$ y $\Delta DN2$

Una vez obtenidas las estrellas de las 450 fotos reales, se realizaron sucesivas pruebas con el algoritmo de reconocimiento. Cada prueba consistió en la ejecución del algoritmo sobre una gran cantidad de fotos (del orden de 100) y entre prueba y prueba se modificaron los parámetros del algoritmo $\Delta DN1$ y $\Delta DN2$. Se estableció $\Delta DN1 = 6$ y $\Delta DN2 = 3$.

3.3.3. Otros cambios realizados en el ARI

Además de modificarse el método de detección de estrellas del ARI que da lugar al ARI nuevo, también se realizó un cambio en la forma de obtener el valor del fondo en el comienzo de cada línea. En lugar de tomar como fondo al primer píxel de la línea (ARI viejo) se utiliza como fondo al promedio de los primeros cinco píxeles de la línea inmediatamente superior (ARI nuevo).

Por otra parte, se modificó la actualización del valor de fondo en cada píxel. En un principio se calculaba el fondo como el promedio de los últimos diez píxeles. Con la modificación, se calcula el fondo a partir de los DN del nuevo píxel y del valor del fondo previo. Esto reduce la cantidad de operaciones de punto flotante de 4 a 3:

$$\text{Promedio: } fondo_{nuevo} = \frac{fondo_{viejo} \cdot 10 - DN_{ultimo} + DN_{nuevo}}{10} \quad 4 \text{ operaciones}$$

$$\text{Actualización simple: } fondo_{nuevo} = \frac{fondo_{viejo} \cdot 7 + DN_{nuevo}}{8} \quad 3 \text{ operaciones}$$

3.4. Performance del ARI en fotos reales

3.4.1. Histogramas de ARI 1 y ARI 2

Se probaron las 450 imágenes reales en reposo. La información más tangible y representativa de la calidad del ARI son los promedios de los N_FP y N_VN, extraídos de los histogramas. La figura 3.4 y

figura 3.5 muestran los histogramas para el ARI 2. Se obtuvo 21.16 para el promedio de los N_FP y 21,01 para el promedio de los N_VN. Los histogramas de la figura 3.6 y figura 3.7 corresponden a las pruebas con el ARI 1. Las medias de los mismos son de 19.21 y de 13.05.

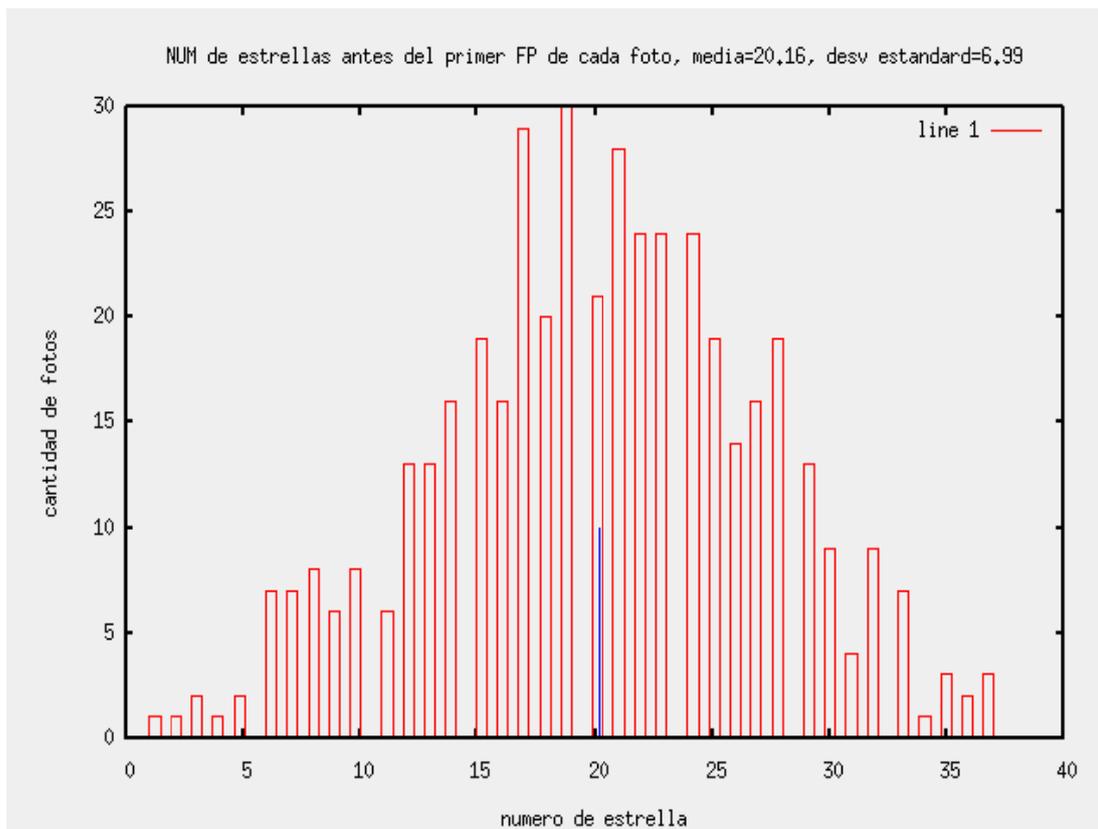


figura 3.4: Histograma de primeros FPs, 450 fotos estáticas, ARI 2.

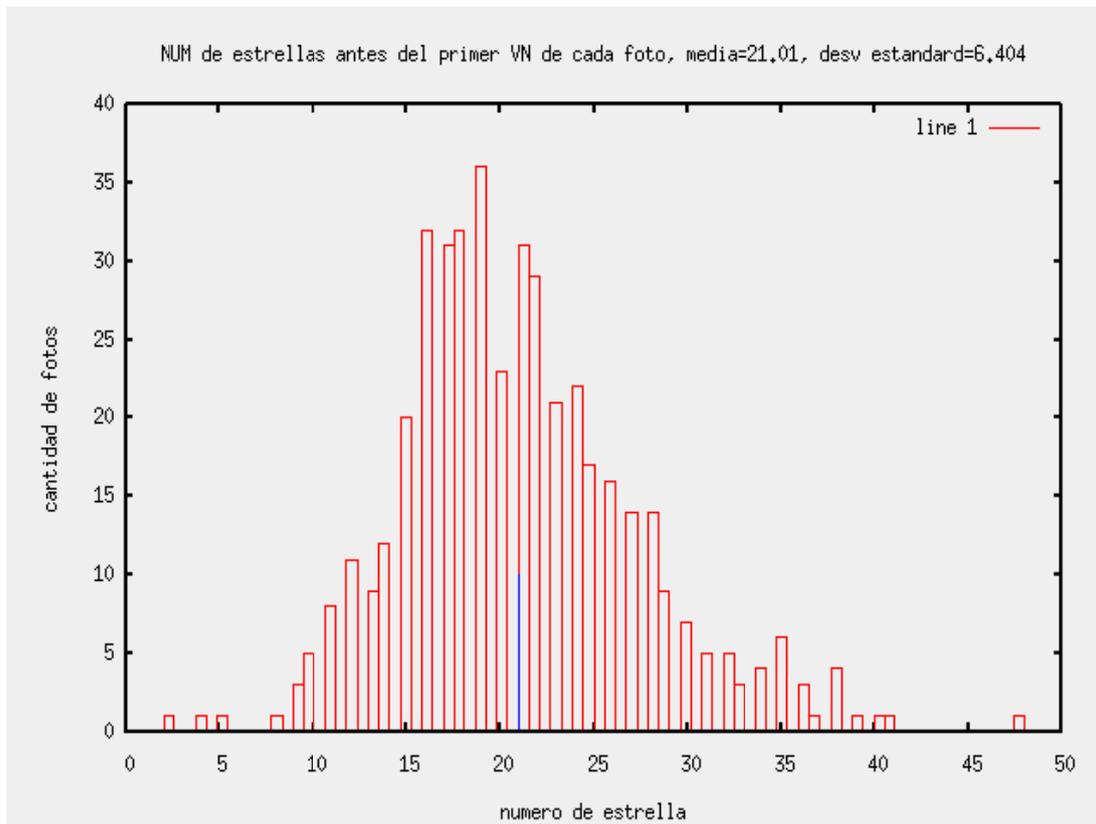


figura 3.5: Histograma de primeros VNs, 450 fotos estáticas, ARI 2.

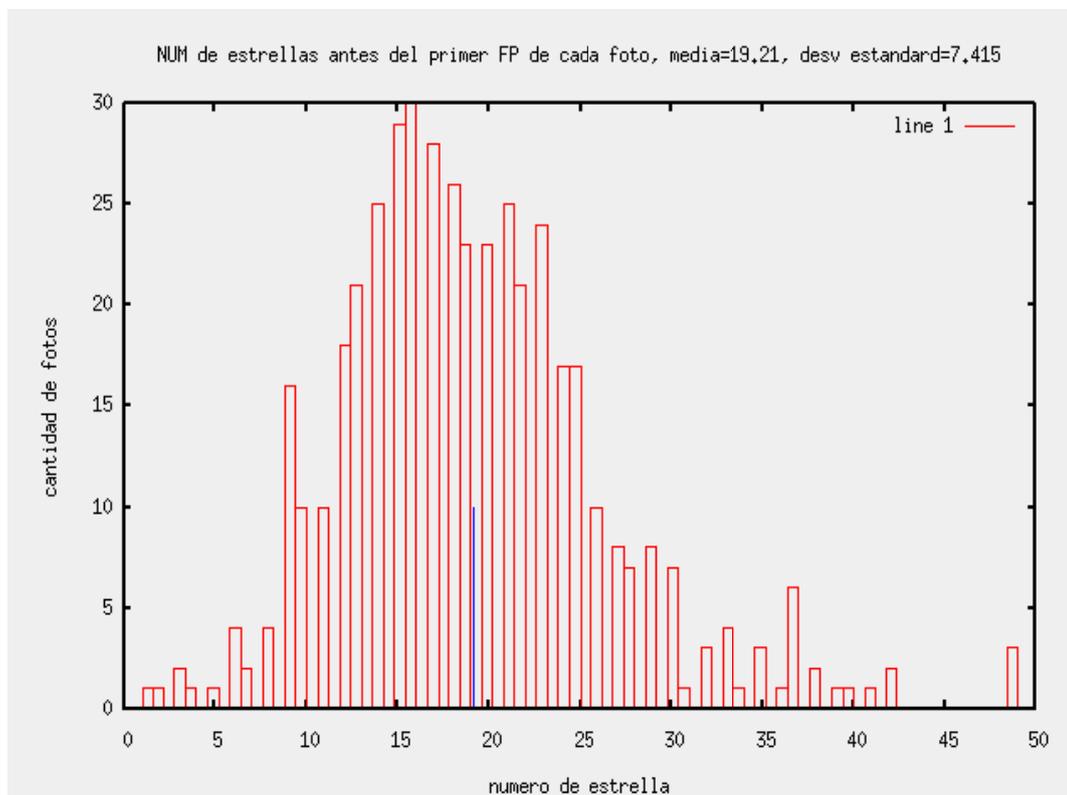


figura 3.6: Histograma de primeros FPs, 450 fotos estáticas, ARI 1.

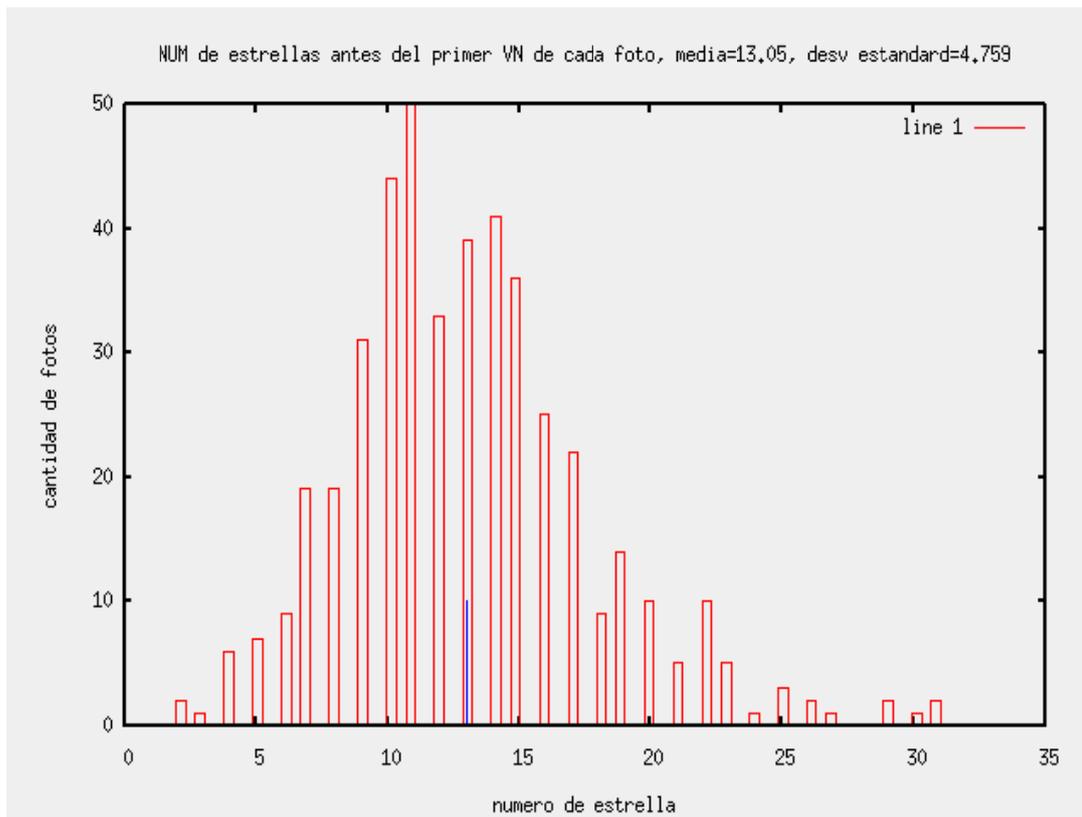


figura 3.7: Histograma de primeros VNs, 450 fotos estáticas, ARI 1.

3.4.2. Visualización de FPs y VNs con el ARI 2

Se observaron las primeras 102 ventanas de FPs y 68 ventanas de VNs a los fines de conocer las fallas del ARI. Se utilizó para esto el ARI 2.

Las posiciones en los que la HDP delató FPs resultaron ser en su mayoría pequeños conjuntos de píxeles levemente más intensos que el fondo, y por otra parte se encontraron conjuntos grandes de píxeles intensos. Los segundos son claramente estrellas, por lo tanto son falsos reportes de la HDP y se encontraron en el 33% de los casos. La figura 3.8 muestra tres ejemplos de el primer tipo de FPs (pequeños conjuntos de píxeles levemente más intensos que el fondo).

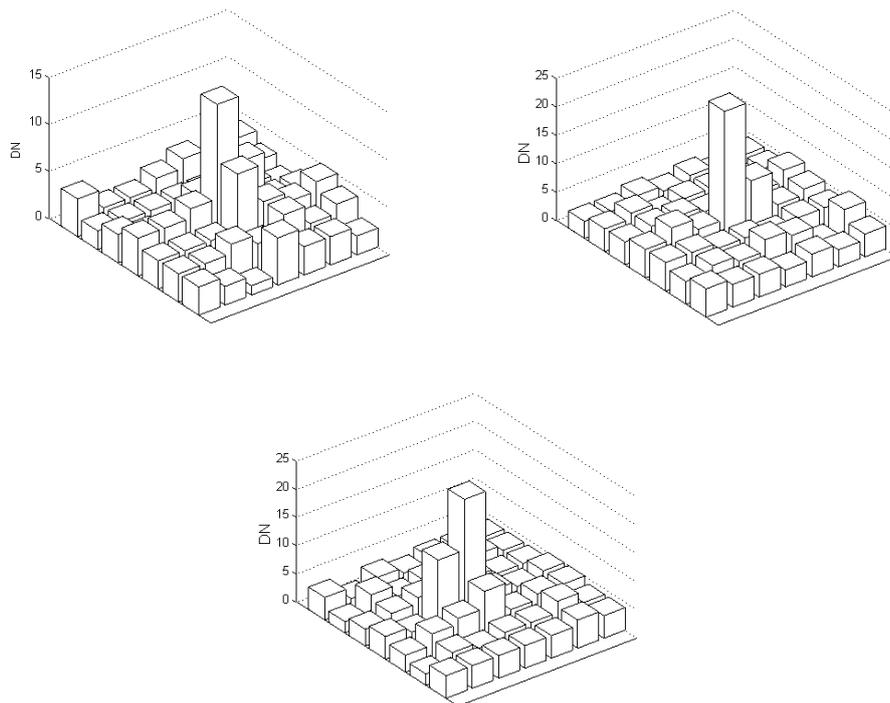


figura 3.8: FPs delatados por HDP, conjuntos de pocos píxeles levemente altos.

Las posiciones en los que la herramienta de prueba delató VNs resultaron pequeñas estrellas que no fueron detectadas por el ARI, o bien falsas detecciones de VNs, es decir, conjuntos de píxeles que no están mas calientes que el fondo, mas allá del ruido. Igual que con los FPs, los segundos tienen como causa a errores de la HDP, constituyen el 32% de las 68 ventanas observadas. La figura 3.9 ilustra 2 ejemplos de reportes de VNs del primer caso (pequeñas estrellas que no fueron detectadas por el ARI).

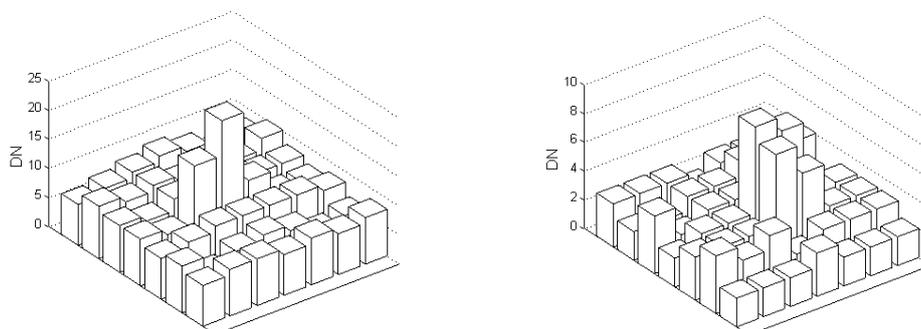


figura 3.9: VNs delatados por la HDP, conjuntos de pocos píxeles levemente altos.

Cabe destacar que los FPs y VNs que revela correctamente el ARI son centros o estrellas muy débiles. Un dato cuantitativo de este hecho es que el 89% de las fallas visualizadas están por encima de la posición 20 en las listas.

3.5. Conclusiones de ensayos con fotos estáticas

3.5.1. Funcionamiento de la HDP

Las fallas de la HDP mencionadas en resultados son:

- El 33% de los FPs delatados no son FPs realmente, según las 51 observaciones.
- El 32% de los VNs delatados no son VNs realmente, según las 34 observaciones.

Este mal funcionamiento de la herramienta de prueba hace que los resultados de los histogramas sean conservativos, en el sentido de que la calidad del ARI es mejor que la reportada. Es decir que el promedio de la cantidad de estrellas sin FPs y sin VNs en cada foto, es mayor que 20.16 y 21.01 respectivamente para el ARI 2, valores mencionados en la sección resultados.

3.5.2. ARI 1 y ARI 2

Debido a los resultados de N_{FP} y N_{VN} para el ARI 1 y el ARI 2 (19.21 y 13.05 para el ARI 0 y 20.16 y 21.01 para el ARI 1), se concluye que hay una mejora en el segundo algoritmo respecto al primero.

Vale destacar que el ARI 2 es de más fácil comprensión y tiene menor número de líneas de código que el ARI 1, siendo esta una ventaja desde el punto de vista de la simplicidad y confiabilidad.

Un típico FP encontrado con el ARI 1 es el que resulta luego de un píxel muy bajo con respecto al fondo (píxel erróneo), en este caso el valor del fondo tomado por el ARI decrece significativamente y el siguiente píxel es considerado alto, en este punto el ARI 1 delata una estrella mientras que el ARI 2 es robusto.

Como ya se mencionó en la sección 3.1.2, el ARI falla inevitablemente para estrellas suficientemente pequeñas, por más que sean divisables a ojo. Tanto con el ARI 1 como con el ARI 2, aumentando la sensibilidad del algoritmo (ΔDN , $\Delta DN1$ y $\Delta DN2$ bajos) se reconocen muchas estrellas, a costa de mayor cantidad de FPs. En el otro extremo, con baja sensibilidad (ΔDN , $\Delta DN1$, $\Delta DN2$ altos), se encuentran muchos VNs. Se optimizaron los parámetros $\Delta DN1$ y $\Delta DN2$ en el ARI 2 para maximizar la performance (mismo número de fallas de cada tipo).

Capítulo 4: Ensayos con imágenes de estrellas en movimiento

El movimiento del centro de la estrella en la imagen se debe al tiempo de exposición del detector óptico y a la velocidad de giro del satélite. En las imágenes con movimiento, las estrellas aparecen alargadas en lugar de redondas y el área que ocupan es mayor al de las estrellas estáticas. Como inconveniente desde el punto de vista del reconocimiento, la intensidad máxima de cada estrella queda disminuida, ya que la intensidad total debe distribuirse en una gran cantidad de píxeles. Si se tiene el star tracker girando a una velocidad ω , siendo el ángulo que le corresponde al ancho de un píxel igual a α y con un tiempo de exposición del detector de t , la cantidad de píxeles que barre el centro de una estrella es:

$$n = \frac{\omega \cdot t}{\alpha}$$

Se puede tomar como referencia una velocidad angular de diseño de $2^\circ/\text{seg}$, por otra parte, se tiene un ángulo de píxel de $\alpha = 0.033^\circ$ y un tiempo de exposición de $0,2\text{s}$. Con estos valores la cantidad de píxeles que barre el centro de la estrella es $n \cong 12$.

Así las cosas, a mayor velocidad angular, menor será la cantidad de estrellas detectadas en cada imagen. Un dato importante es que el post-procesamiento (etapas b y c del star tracker) que determina la actitud necesita un mínimo de 4 estrellas para un reconocimiento exitoso (ver [3]).

Es de gran interés conocer la velocidad límite a la cual el star tracker puede reconocer la actitud. En principio, esta velocidad máxima no está bien definida, si no que depende de la cantidad de estrellas en el FOV y de la magnitud de las mismas. A su vez, la cantidad de estrellas en el FOV varía según la posición estelar de la foto. En el análisis de reconocimiento en velocidad se obtuvo, para distintas velocidades, la probabilidad de reconocimiento exitoso de actitud, calculada como la frecuencia con la que se obtienen mas de 4 estrellas en las fotos reconocidas. Recopilando datos de varias velocidades, es posible resumir la información en gráficas de *cantidad de estrellas encontradas versus velocidad* y de *probabilidad de reconocimiento exitoso versus velocidad*.

4.1. Generación de imágenes simuladas

Se realizó un modelo de estrellas reales estáticas para generar imágenes simuladas en movimiento. Para esto se tomaron una gran cantidad de estrellas reales, se ajustó el modelo con estas estrellas y se verificó su validez en forma cualitativa. Dicho ajuste se realizó con ayuda de MATLAB. Paso seguido se realizó en C una rutina para generar imágenes. Esta realiza un archivo con el vector de intensidades de la foto simulada, con estrellas distribuidas en la imagen de forma conveniente. Uno de los parámetros de entrada es la velocidad de rotación.

Por otra parte se utilizaron los programas *simpix* y *autosimpix* realizados por INVAP. El primero genera imágenes estelares simuladas tomando como entradas a la posición de la esfera celeste de la cual se desea la imagen y la velocidad de rotación, entre otras cosas. *Autosimpix* es una rutina que utiliza *simpix* reiteradas veces con el fin de realizar un grupo de imágenes, cada una con sus parámetros de posición y velocidad.

La generación de imágenes con el primer método (estrellas uniformemente distribuidas) fue utilizada para detectar fallas en el ARI en la detección de estrellas en movimiento. Una vez corregidas las fallas, los programas *simpix* y *autosimpix* se usaron en la generación de una gran cantidad de imágenes para validar y caracterizar el ARI.

4.1.1. Modelo de estrellas estáticas y en movimiento

A la hora de simular estrellas, es importante poder reproducir la distribución de DN's en los píxeles que conforman dicha estrella. Tomando como coordenadas cartesianas en el plano de la imagen a x e y , un modelo matemático que modela con muy buena aproximación la distribución para estrellas estáticas es

$$f(x, y) = A \cdot e^{-\frac{(x-x_0)^2 + (y-y_0)^2}{2\sigma^2}} \quad (1)$$

donde x_0 e y_0 son las coordenadas del centro de la estrella, A y σ son constantes que determinan la intensidad y superficie de la estrella. La figura 4.1 muestra la forma de la función de dos variables dada por la formula (1).

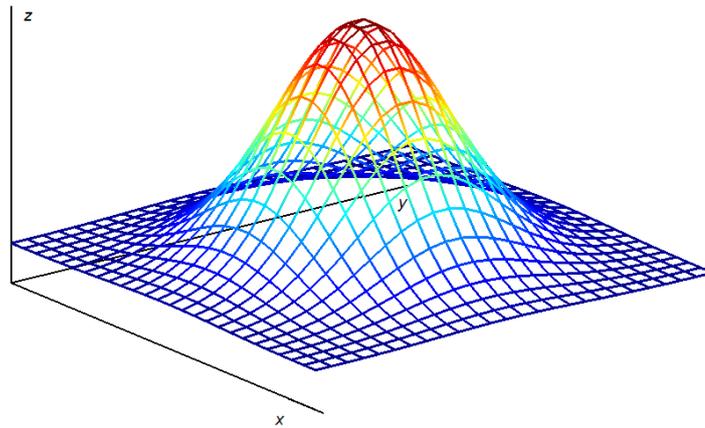


figura 4.1: Esquema de la función (1)

En el caso de una foto digital ideal⁴, el valor del DN en un pixel es la integral de la función (1) en el área del pixel, dividido por dicha área. Sin embargo las fotos reales presentan una desviación de este valor de DN ya que los detectores de cada pixel no toman luz de todo el área en forma uniforme (pixel non-uniformity). En forma aproximada, podemos considerar que los DN de un pixel son el valor de la función (1) evaluada en el centro del pixel.

Se puede pensar que una estrella movida en una imagen es la superposición de muchas estrellas estáticas de poca intensidad con sus centros distribuidos a lo largo de un segmento de la imagen. En forma continua, se puede generar una estrella en movimiento con la ecuación

$$f_2(x, y) = \int_0^1 A \cdot e^{-\frac{(x-x_c(t))^2 + (y-y_c(t))^2}{2\sigma^2}} dt \quad (2)$$

en este caso $x_c(t)$ e $y_c(t)$ son funciones lineales en t , y dan los puntos que barre la estrella en el FOV. En otras palabras, t parametriza la recta formada por las ecuaciones $x_c(t)$ e $y_c(t)$. Dado que la ecuación (2) no se puede integrar simbólicamente, se recurre a realizar una integración numérica, resulta

$$f_2(x, y) \approx \sum_{i=1}^N \frac{A}{N} \cdot e^{-\frac{(x-x_c(t))^2 + (y-y_c(t))^2}{2\sigma^2}} \quad (3)$$

⁴ En una foto ideal, cada pixel da el valor de la luz recibida en ese área del detector. Esto no se cumple estrictamente en un detector óptico por no existir uniformidad en cada pixel (“pixel non-uniformity”).

Donde ahora t toma solo valores enteros, $x_c(t)$ e $y_c(t)$ son puntos equiespaciados a lo largo de la recta que barre la estrella y N es la cantidad de intervalos de integración.

Para aplicar el modelo en la generación de fotos simuladas se deben ajustar las estrellas estáticas reales con la ecuación (1), de forma de obtener los parámetros A , x_0 , y_0 y σ para estrellas reales. A continuación se explica el procedimiento de ajuste. Como primer paso se extrae la porción de la foto que contiene a la estrella, se tiene una matriz de 8×8 cuyos elementos son los DN de cada pixel. Llamaremos $dn(x,y)$ a la función de intensidad de la estrella real (función discreta), esto es el elemento (x,y) de la matriz. El centro (x_0, y_0) es el baricentro de la de la matriz y se calcula como

$$x_0 = \frac{\sum_{x=1}^8 \sum_{y=1}^8 x \cdot dn(x,y)}{\sum_{x=1}^8 \sum_{y=1}^8 dn(x,y)} \quad y_0 = \frac{\sum_{x=1}^8 \sum_{y=1}^8 y \cdot dn(x,y)}{\sum_{x=1}^8 \sum_{y=1}^8 dn(x,y)} \quad (4)$$

Por otra parte se calcula DN_t , que es la suma de los $dn(x,y)$ para todos los x e y , que equivale a la integral de volumen sobre la estrella, si la función fuera continua.

$$DN_t = \sum_{x=1}^8 \sum_{y=1}^8 dn(x,y) \quad (5)$$

Calculando la integral de volumen de la superficie de la formula (1) se obtiene

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y) dx dy = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} A \cdot e^{-\frac{(x-x_0)^2 + (y-y_0)^2}{2\sigma^2}} dx dy = 2\pi A \sigma \quad (6)$$

Como la integral de la estrella real es igual a DN_t , se debe cumplir la igualdad

$$2\pi A \sigma = DN_t \quad (7)$$

con lo que

$$A = \frac{DN_t}{2\pi\sigma} \quad (8)$$

No es posible calcular A porque no se conoce el valor de la incógnita σ . Ingresando la ecuación (8) en la ecuación (1) tenemos

$$f(x, y) = \frac{DN_t}{2\pi\sigma} \cdot e^{-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma^2}} \quad (9)$$

Así las cosas, no es necesario el cálculo de A ya que se tiene a la fórmula (1) en función de DN_t , un parámetro más directo. Solo queda determinar el parámetro σ , para lo cual se puede recurrir al método de cuadrados mínimos. Según este método, se busca el valor de σ que minimiza la suma de los cuadrados de las distancias entre la fórmula (9) y la estrella real. Matemáticamente, se debe minimizar el funcional

$$F(\sigma) \approx \sum_{x=1}^8 \sum_{y=1}^8 \left(\frac{DN_t}{2\pi\sigma} \cdot e^{-\frac{(x-x_0)^2+(y-y_0)^2}{2\sigma^2}} - dn(x, y) \right)^2 \quad (10)$$

Debido a la dificultad de resolver el mínimo de $F(\sigma)$ en forma analítica, se recurre a evaluar la función en varios valores de σ y buscar el menor de todos.

4.1.2. Aplicación del modelo

Se extrajeron 50 estrellas de una foto real estática con una subrutina dentro del algoritmo del ARI. Las estrellas de diferentes magnitudes se almacenaron como matrices de 8x8 en un archivo. Se realizó una rutina en MATLAB que calcula el parámetro σ para cada estrella. Con este fin se generó un vector $\bar{\sigma}$ con valores crecientes entre 0,1 y 4 equiespaciados por 0,05 (78 elementos). Se evaluó el funcional $F(\sigma)$ dado por la fórmula (10) en los valores del vector $\bar{\sigma}$ y se buscó el mínimo de los valores resultantes. Como resultado para el valor de σ se obtuvo 0,55 píxeles independientemente del tamaño de la estrella.

Paso seguido se realizaron en MATLAB gráficas de superficies que dan valores de los DN s en cada pixel de la estrella. Comparando gráficos de estrellas simuladas y reales se realiza una verificación visual cualitativa. La figura 4.2 y figura 4.3 muestran ejemplos para estrellas medianamente brillantes y muy poco brillantes respectivamente. En el primer caso la estrella simulada y real tienen 128 y 166 DN s respectivamente. En el segundo caso tienen 11 y 14 DN 's respectivamente. Notar que en el caso de estrellas pequeñas el ruido de fondo se evidencia más fuertemente que con estrellas grandes.

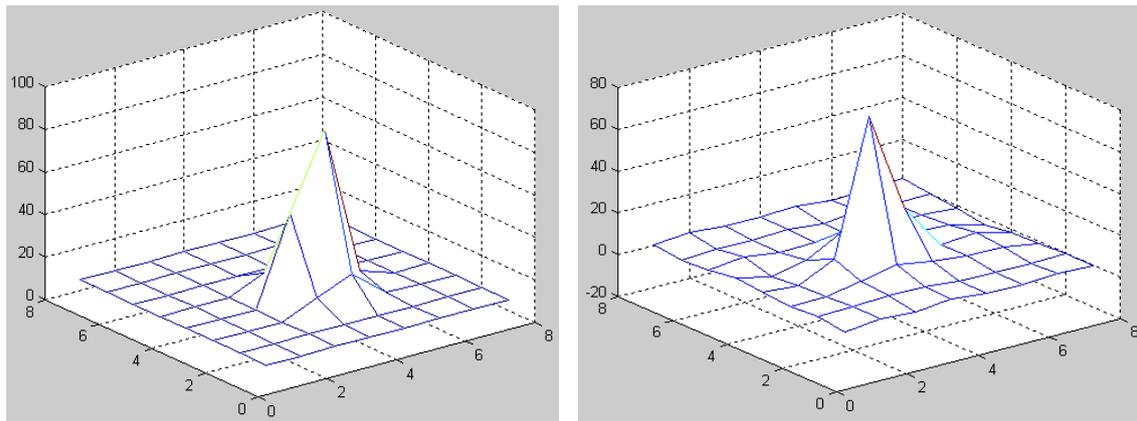
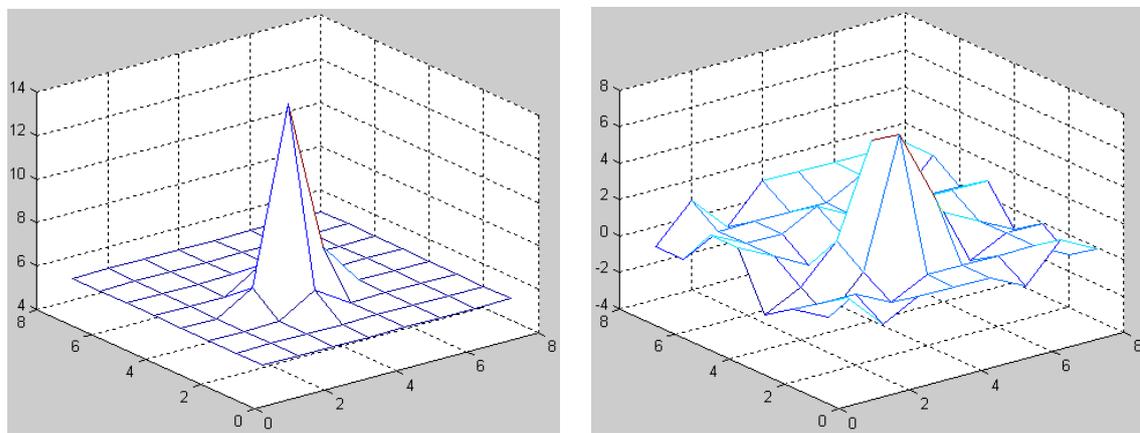


figura 4.2: Estrella simulada (a) y una estrella real (b).



(a)

(b)

figura 4.3: Estrella simulada (a) y estrella real (b), magnitudes comparables.

4.1.3. Generación de imágenes con estrellas uniformemente distribuidas

Se realizó una rutina en C para generar los archivos con los vectores de las imágenes. El funcionamiento se basa en la evaluación de la formula (9) en cada pixel de la imagen y para cada una de las estrellas presentes.

En primer lugar se realizan dos vectores con las componentes x_0 e y_0 de los centros de las estrellas y se definen las funciones lineales. Los centros de cada estrella deben barrer una recta dada por

$$x_c(t) = a \cdot t + x_0$$

$$y_c(t) = b \cdot t + y_0$$

El parámetro t toma valores entre 1 y $N = 15$, por lo que el centro de la estrella movida resultante estará levemente desplazado de x_0 . Una vez fijados los parámetros a y b , que son iguales para todas las estrellas de la foto, se puede escribir la función de una estrella como

$$f(x, y, x_0, y_0, t) = \frac{DN_t}{2\pi\sigma N} \cdot e^{-\frac{(x-x_c(t))^2+(y-y_c(t))^2}{2\sigma^2}}$$

Recordar que x e y tienen unidades de píxeles. Finalmente, el valor de cada pixel de la imagen es la suma de las evaluaciones de todas las funciones $f(x, y, x_0, y_0, t)$.

Se implementó la rutina de generación de imágenes (ver en Apéndice II los nombres de las rutinas) en la realización de imágenes con arreglos de 64 estrellas en movimiento. Las posiciones (x_0, y_0) componen un arreglo de 8 por 8 con intensidades decrecientes de izquierda a derecha y de arriba a abajo. La figura 4.4 muestra un ejemplo de dichas fotos. La estrella mas brillante (superior izquierda) tiene 480 DN s mientras que la menos brillante (inferior derecha) tiene 1 DN . Esta distribución de estrellas permite evaluar al ARI según la cantidad de estrellas detectadas en función de la velocidad.

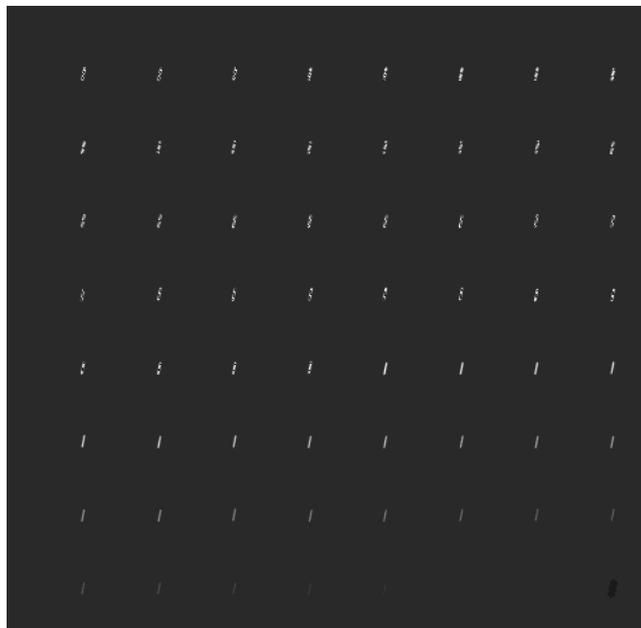


figura 4.4: Foto simulada con el programa generador de estrellas.

Fueron generadas 63 imágenes de este tipo, variando en cada foto el ángulo de la estela de las estrellas. Así se logran ángulos entre 0 y $6,3rad \approx 2\pi rad$ con una imagen cada $0,1rad$. Notar que la segunda mitad de fotos tiene los mismos ángulos que la primera, ya que la suma de πrad a una estela da una estrella equivalente. Las 63 fotos con arreglos de estrellas en distintos ángulos proveen $63 \cdot 64 = 4032$ estrellas para probar el ARI.

4.1.4. Generación de imágenes con *simpix* y *autosimpix*

Para una etapa de validación y caracterización del ARI se simularon las fotos con los programas *simpix* y *autosimpix* realizados por INVAP (ver descripción de los programas en Apéndice I). Se hicieron 14 series de 1382 fotos cada una, para velocidades entre 0 y $0,065 rad/s$ ($3,72^\circ/s$).

La cantidad de veces que entra el FOV (cuadrado de 15° de lado) en la esfera celeste se puede calcular como el ángulo sólido de la esfera dividido por el ángulo sólido del FOV. Esto es aproximadamente igual a:

$$\frac{A_{esfera}}{A_{FOV}} \cong \frac{4\pi rad^2}{0,2618^2 rad^2} \cong 183$$

Para realizar las pruebas con el algoritmo de reconocimiento, se consideran necesarias entre 5 y 10 veces esa cantidad, de forma de barrer varias veces la esfera celeste. El valor utilizado es de 1382 y está dentro de esta condición.

Se realizaron las imágenes con un nivel de fondo constante de 50 DNs sumado a un ruido de 2 DN de amplitud. De esta forma, un pixel de fondo (donde no hay estrellas) toma el valor de 49, 50 o 51 DN con igual probabilidad cada uno.

Para probar el algoritmo en condiciones diversas, se utilizaron velocidades de rotación con dirección aleatoria en las componentes x e y de la foto. En otras palabras, la dirección de las estelas de las estrellas en una foto es aleatoria.

En un sistema de ejes fijos a la imagen, como muestra la figura 4.5, las componentes ω_1 , ω_2 y ω_3 corresponden a rotaciones en los ejes x , y e z respectivamente.

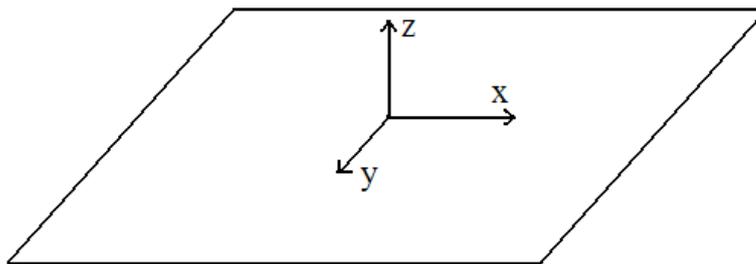


figura 4.5: Imagen y sus ejes de coordenadas

Como podemos ver, una rotación ω_3 genera una imagen con estrellas formando pequeños arcos de circunferencia de longitudes crecientes hacia radios mayores y estrellas prácticamente quietas en la parte central de la foto. La longitud del arco mas grande (ubicado en una esquina de la imagen, a una distancia de $512 \cdot \sqrt{2}/2 \cong 362$ píxeles del centro) toma un valor de $\omega_3 \cdot t \cdot 362$, donde ω_3 está expresado en radianes. Tomando los valores $\omega_3 = 2^\circ/s = 0.0349rad$ y $t = 0,2s$ se obtiene una longitud de estrella de 2.5 píxeles. Como esta longitud es menor que los 12 píxeles mencionados anteriormente para rotaciones con componentes en x e y , se concluye que las rotaciones en z pueden dejarse de lado en las simulaciones: estos no son los casos más severos para el ARI.

Otro detalle a mencionar es la ubicación de las fotos en la esfera celeste. Es importante elegir los puntos en los que estarán centradas las fotos de forma que sean lo mas equidistantes posibles. Esto no se puede hacer distribuyéndolos uniformemente sobre paralelos y meridianos porque de esta forma se acumularían muchos puntos en los polos. A continuación usaremos el sistema de coordenadas ecuatorial, cuyas coordenadas son ascensión recta (α), declinación (δ) y rotación (rot). Este sistema se explica en el Apéndice IV. La solución elegida fue dividir el eje δ en 19 intervalos iguales, o sea $\delta = -90^\circ; -80^\circ; \dots; -10^\circ; 0^\circ; 10^\circ; \dots; 80^\circ; 90^\circ$ y para cada valor de δ se correspondieron una cantidad de valores de α proporcional al perímetro de la circunferencia formada por ese δ (circunferencia punteada de la figura 4.6).

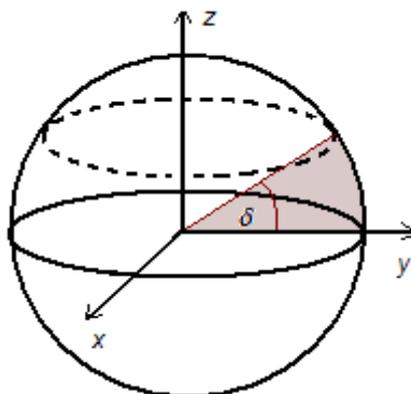


figura 4.6: Esfera celeste de radio unitario, se indica el ángulo de declinación δ .

En el ecuador ($\delta = 0$) se distribuyeron 120 puntos, y en cada declinación δ_i se ubicaron tantos puntos como el entero mas cercano a $120 \cdot \cos(\delta_i)$. En total resultaron 1382 fotos. La figura 4.7 muestra la distribución de las fotos en la esfera celeste.

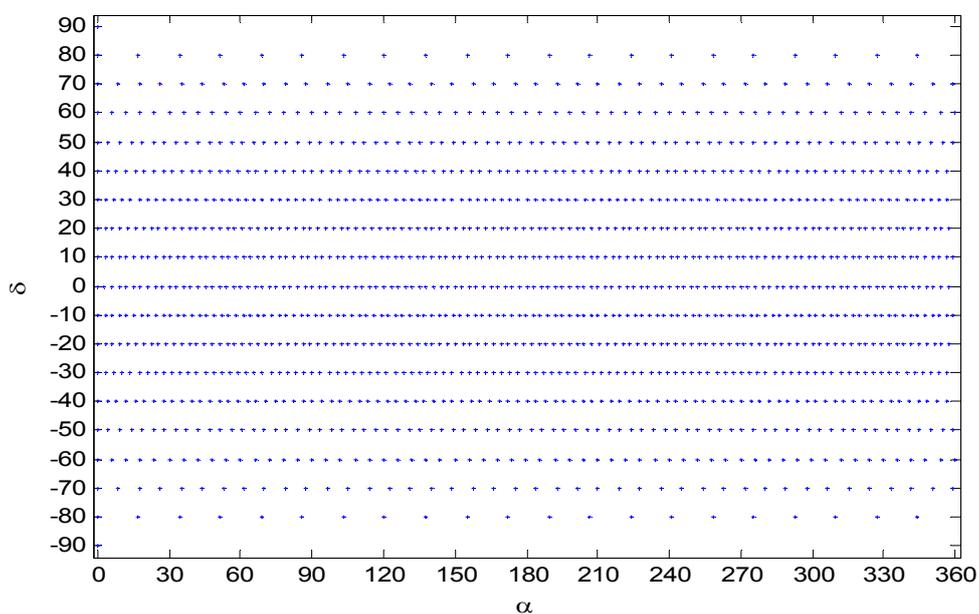


figura 4.7: Distribución de las 1382 fotos simuladas.

Otro dato de entrada de *simpix* es la coordenada de rotación. En estas pruebas, la rotación fue fijada en 0 para todas las imágenes realizadas.

4.2. Primeras pruebas de reconocimiento

4.2.1. Problema de la detección múltiple

Se probó el ARI con las imágenes con estrellas uniformemente distribuidas (explicadas en la sección 4.1.3). Como se puede apreciar en el ejemplo de la figura 4.8, se encontró un mal funcionamiento del ARI que corresponde a una detección múltiple de estrellas. La estrella mas pequeña reconocida tiene 107 *DNs*.

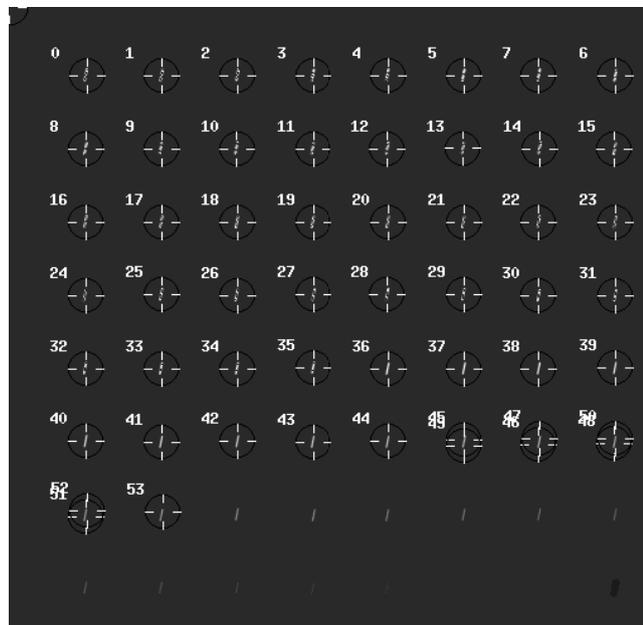


figura 4.8: Reconocimiento de centros en la foto de la figura 4.4.

Se puede ver que esta falla sucede con estrellas pequeñas (las mas pequeñas que el ARI puede detectar). Ocurre tanto para el ARI 1 como para el ARI 2 y se debe a que la secuencia del ARI (secciones 2.3 y 3.3.1) pasa muchas veces por la misma estrella en diferentes partes y no llega a unir todos los sectores en los que encuentra píxeles de alto valor. Cuando el ARI encuentra un grupo de píxeles intensos, indaga si dentro de ese grupo alguno de los píxeles pertenece a alguna estrella ya detectada. Las detecciones múltiples ocurren más frecuentemente cuando la estela de la estrella tiene un ángulo entre $\sim 20^\circ$ y $\sim 80^\circ$ respecto del eje x , en la dirección que ilustra la figura 4.8.

Para visualizar las estrellas múltiples detectadas sobre las fotos simuladas se realizó un programa que, tomando como entrada una serie de listas del ARI, entrega en la consola de Linux un arreglo de símbolos por cada foto, como se muestra en la figura 4.9 (mención de la rutina en Apéndice II).

```

---ángulo=4.2 radianes---
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * * * 2 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
---
---ángulo=4.3 radianes---
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
* * * 2 2 2 * 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
---
---ángulo=4.4 radianes---
* * * * *
* * * * *
* * * * *
* * * * *
* * * * *
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0

```

figura 4.9: Salida del programa de detección de reconocimiento múltiple.

En cada arreglo se indica el ángulo que tienen las estrellas respecto al eje x en la foto. De los elementos del arreglo, el símbolo “*” indica que una estrella fue detectada correctamente, el “o” indica que ninguna estrella se detectó en ese lugar y un número mayor a 1 delata la cantidad de estrellas múltiples detectadas en esa posición.

Se observaron varios casos de detecciones múltiples a nivel de pixel y todos presentaron pares o tríos de grupos de píxeles a pequeña distancia. Las figuras figura 4.10, figura 4.11, y figura 4.12 muestran ejemplos de esto. Las estrellas son débiles y debido al ruido la parte central de la misma no es lo suficientemente alta como para ser detectada. Se encontraron 59 detecciones múltiples en 50 imágenes.

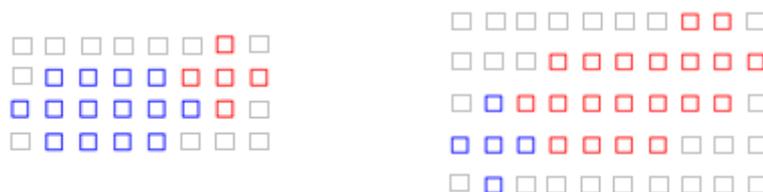


figura 4.10: Falla del ARI, dos estrellas reconocidas como dos centros cada una.

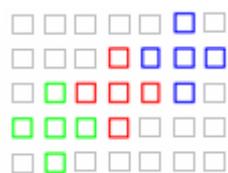


figura 4.11: Estrella reconocida como tres centros.

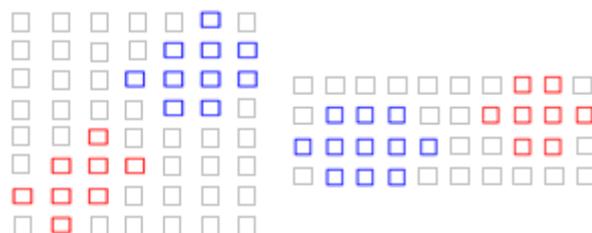


figura 4.12: En estos casos, los dos centros de la estrella se encuentran separados.

4.2.2. Solución al problema

A continuación se explica la corrección realizada en el código del ARI para solucionar la falla de detecciones múltiples. El ARI original, luego de encontrar un grupo de cinco píxeles intensos (ver sección 3.3.1), indaga si en dicho grupo de píxeles hay alguno que pertenece a una estrella ya detectada. La corrección consiste en verificar además los píxeles que se muestran en la figura 4.13 (píxeles indicados con "?"). Es decir, una vez encontrados cinco píxeles altos (los que se indican con "*" en la figura), se interroga si hay una estrella o porción de estrella cercana ya identificada.

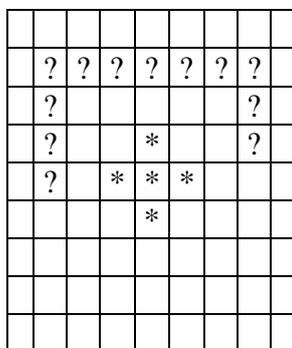


figura 4.13: Píxeles interrogados por el ARI al encontrar una estrella.

Sólo se interroga la parte superior de la estrella: en la parte inferior no puede haber centros porque el ARI no exploró todavía ese sector de la imagen. Notar que de esta manera, un par de estrellas a una distancia menor que cinco píxeles serán reconocidas como una sola estrella.

4.3. Validación y medición de performance del ARI

La forma de evaluar el ARI con fotos movidas es determinando la probabilidad de reconocimiento exitoso (P) en función de la velocidad de rotación (ω) del satélite. La información que usaremos para obtener P es la *cantidad de estrellas identificadas* (N) en la imagen en función de ω . Aclaremos que N solo tiene en cuenta las estrellas que están en el catálogo de vuelo (apéndice III). Ambos conjuntos de datos (P vs. ω y N vs. ω) son relevantes en la ficha de información técnica de cualquier star tracker comercial.

El reconocimiento de actitud posterior al ARI funciona si se detectan cuatro o más estrellas que figuran en el catálogo de vuelo. Entonces, la frecuencia con la que se detectan más de cuatro estrellas da P (una estrella detectada en este sentido implica que la estrella además de ser detectada por el ARI, se encuentre en el catálogo de vuelo).

Esto se realizó reconociendo estrellas en varios conjuntos de fotos, cada conjunto con una velocidad constante. Se utilizaron imágenes generadas por *simpix* definidas en la sección 4.1.4. A continuación se explica el procedimiento realizado con cada velocidad.

4.3.1. Medición de la performance del ARI para una velocidad dada

Tomando un grupo de 1382 imágenes generadas como los que se describen en la sección 4.1.4, se realiza el procedimiento que se explica a continuación para determinar la performance del ARI a esa velocidad.

En primer lugar se confecciona un histograma que da la dispersión de la cantidad de estrellas identificadas en cada foto. Esto es, los ejes del histograma representan el *porcentaje de fotos* (eje y) y N (eje x). El subsiguiente procesamiento de datos consiste en ajustar las distribuciones obtenidas en el histograma con una distribución normal, (se usó

la bibliografía [4]). Finalmente, P se calcula como el área debajo de la curva para los valores de abscisa mayores que cuatro.

Para este análisis se usaron rutinas ya existentes y debieron realizarse nuevas. El Apéndice II describe los nuevos códigos.

4.3.2. Resultados y discusión de evaluación de performance

Se usaron los 14 grupos de 1382 fotos con velocidades comprendidas entre 0 y $0,065 \text{ rad/s}$ ($3,72^\circ/\text{s}$). Fueron confeccionados con éxito los 14 histogramas. La figura 4.14 muestra a modo de ejemplo uno de ellos, en particular, en este caso la velocidad de rotación es de $0,04 \text{ rad/s}$. También se observa el ajuste con una gaussiana, cuya concordancia con los datos demuestra que la distribución es normal. El área amarilla bajo la curva para valores de abscisa mayores que 4 representa a P . La desviación estándar en este caso es de $\sigma = 3.3$. Notar que P se incrementa cuando crece el número medio de estrellas identificadas y cuando decrece la desviación estándar de dicho número. En el resto de los histogramas, la desviación estándar no varía significativamente, se mantiene entre 3,0 y 3,4. La figura 4.15 muestra 10 de las 14 distribuciones normales para velocidades de hasta 0.055 rad/s (cada 0.005 rad/s). Se observa que tienen formas prácticamente similares (σ varía muy poco) y se desplazan hacia abscisas menores al aumentar la velocidad. Las distribuciones para velocidades pequeñas resultan tener desplazamientos pequeños.

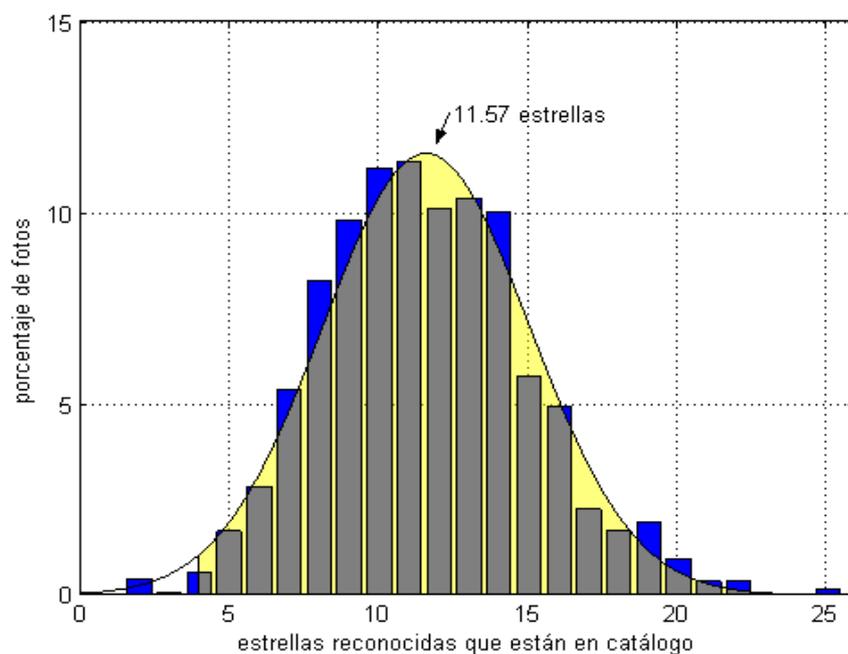


figura 4.14: Histograma para $\omega = 0,04 \text{ rad/s}$.

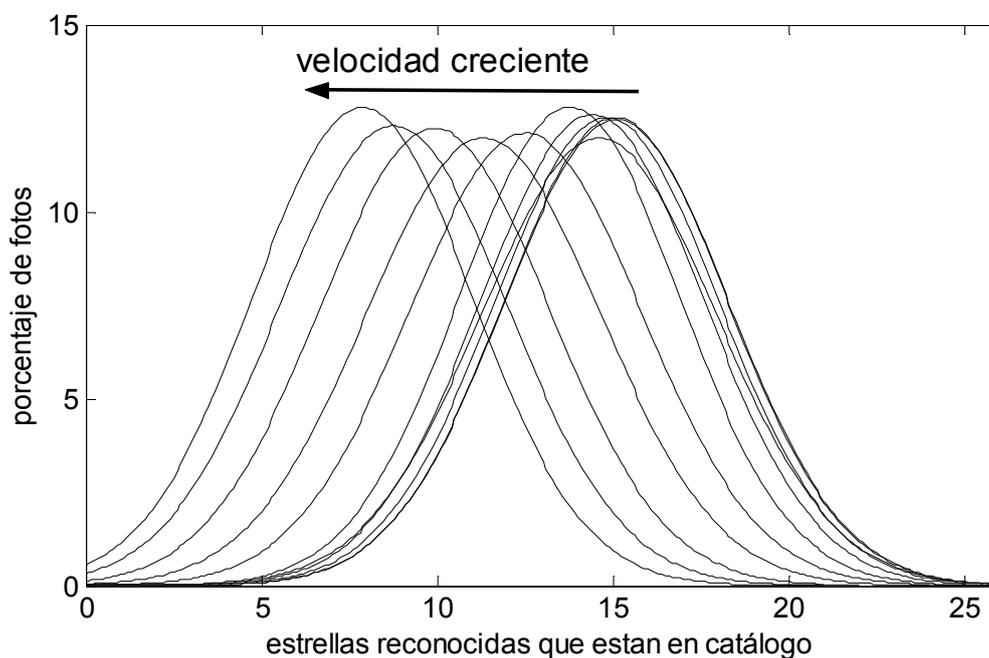


figura 4.15: Conjunto de histogramas para $\omega \leq 0.055 \text{ rad/s}$

Una vez obtenida la información de los 14 histogramas se confecciona el gráfico de cantidad de estrellas promedio (\bar{N}) versus ω (figura 4.16) con los valores medios de cada histograma.

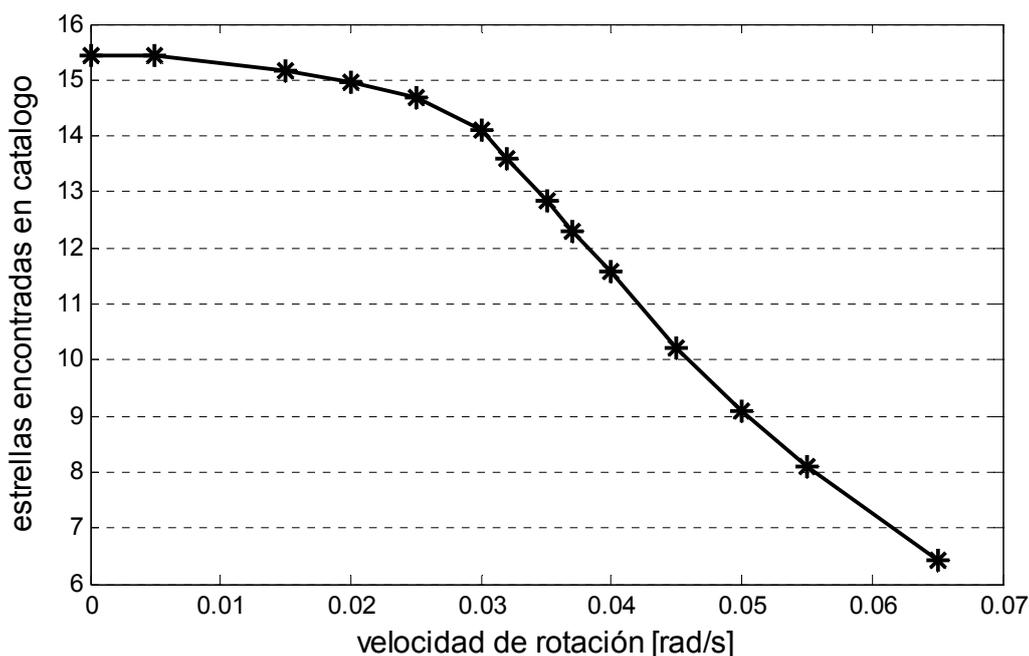


figura 4.16: \bar{N} versus ω .

Se distinguen una zona de poca variación del número de estrellas encontrado (hasta velocidad de rotación de $0,03 \text{ rad/s}$) y una zona de mayor pendiente para velocidades mayores.

El FOV entra 183 veces en la esfera celeste y el catálogo de vuelo tiene 3231 estrellas. Con estos datos, se calcula la cantidad promedio de estrellas en cada FOV como $3231/183 = 17,7$. De estas 17,7, se reconocen entre 14 y 15,5 para $\omega < 0,03 \text{ rad/s}$ (representan entre 79% y 87% respectivamente).

El número medio de estrellas encontradas está siempre muy por arriba de 4, sin embargo esto no garantiza una buena probabilidad de reconocimiento en todos los casos.

La probabilidad de reconocimiento exitoso versus velocidad se ilustra en la figura 4.17 y la figura 4.18 muestra una ampliación de la zona de baja velocidad.

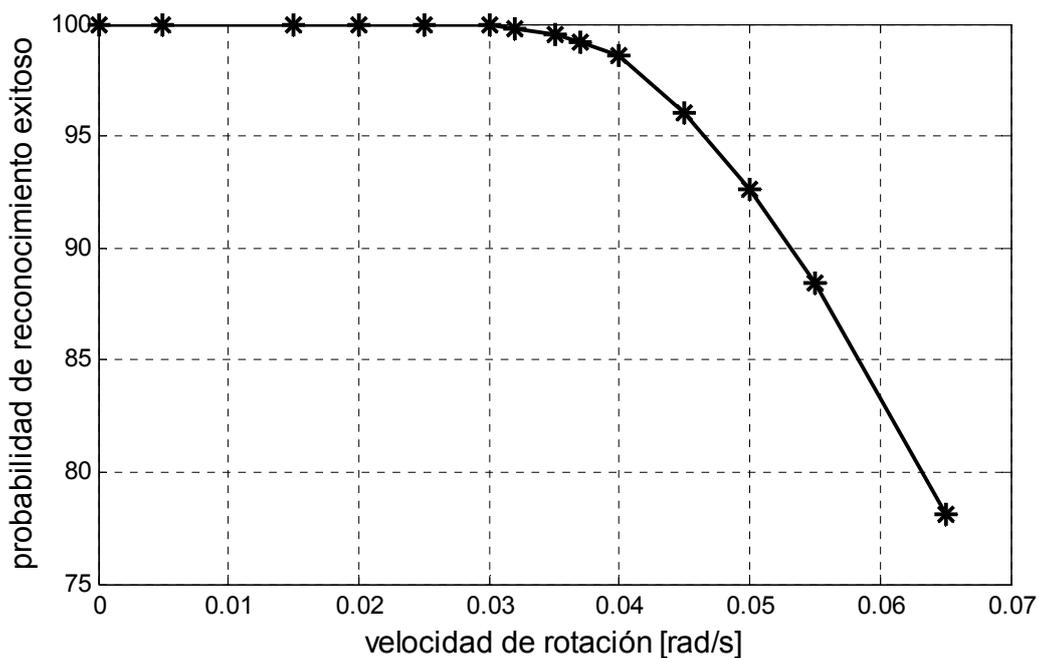


figura 4.17: P versus ω .

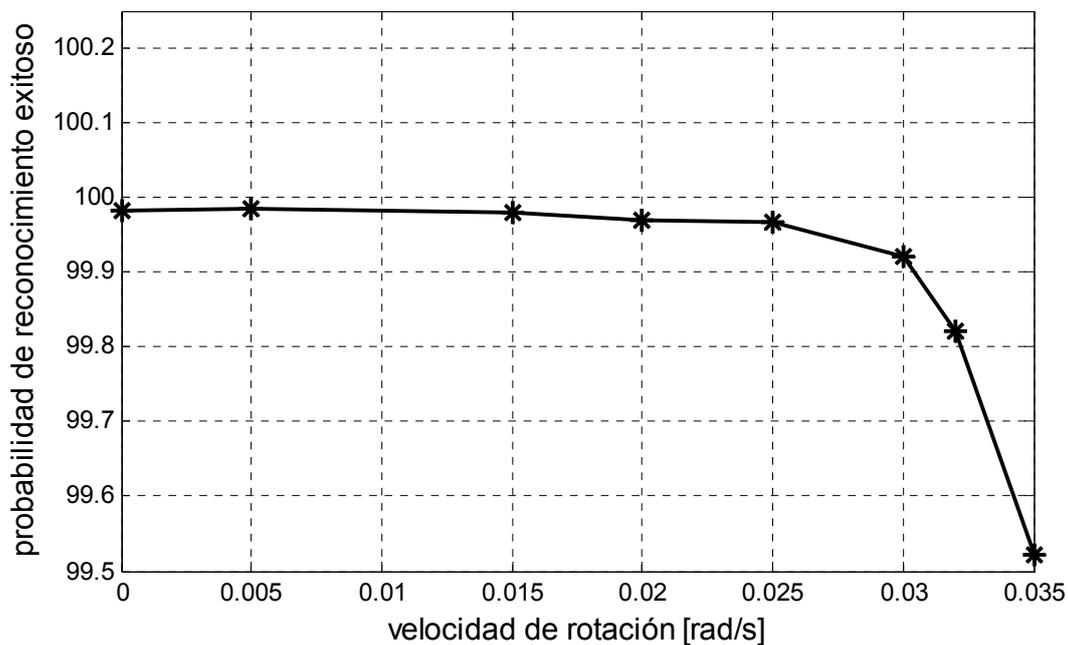


figura 4.18: P versus ω (ampliación).

En este caso también se diferencian dos zonas con distintos comportamientos. Para $\omega < 0.03 \text{ rad/s}$, P está comprendido entre 99,97 y 99,99 mientras que para $\omega > 0.03 \text{ rad/s}$, hay una gran pendiente, prácticamente constante. Queda claro que estos dos comportamientos se relacionan con las dos partes del gráfico $\bar{N}-\omega$, figura 4.16.

Si bien no es posible reportar una velocidad máxima de funcionamiento, podemos adoptar como criterio que el quiebre de $0,03 \text{ rad/s}$ de la curva P versus ω da una velocidad límite para una performance muy buena. Por otra parte, es la velocidad límite para la cual la tasa de falla es menor a 1 en 1000 ($P = 99,9\%$).

4.3.3. Probabilidad de reconocimiento en hemisferio sur y hemisferio norte

En algunos satélites particulares, existe la posibilidad de elegir la dirección en la que apunta el FOV de los ST, como es el caso de un satélite geoestacionario⁵. Para tomar esta decisión, es de interés conocer la performance en el hemisferio sur y el hemisferio norte separadamente, así se puede determinar si hay alguna dirección preferencial.

Se realizó el análisis anterior (gráficas $\bar{N}-\omega$ y $P-\omega$) para los dos hemisferios por separado, aprovechando el hecho de que son necesarios muy pocos cambios en los códigos de los algoritmos. La figura 4.19 muestra \bar{N} versus ω para los dos hemisferios, además incluye la curva de la esfera celeste completa para contrastar los datos.

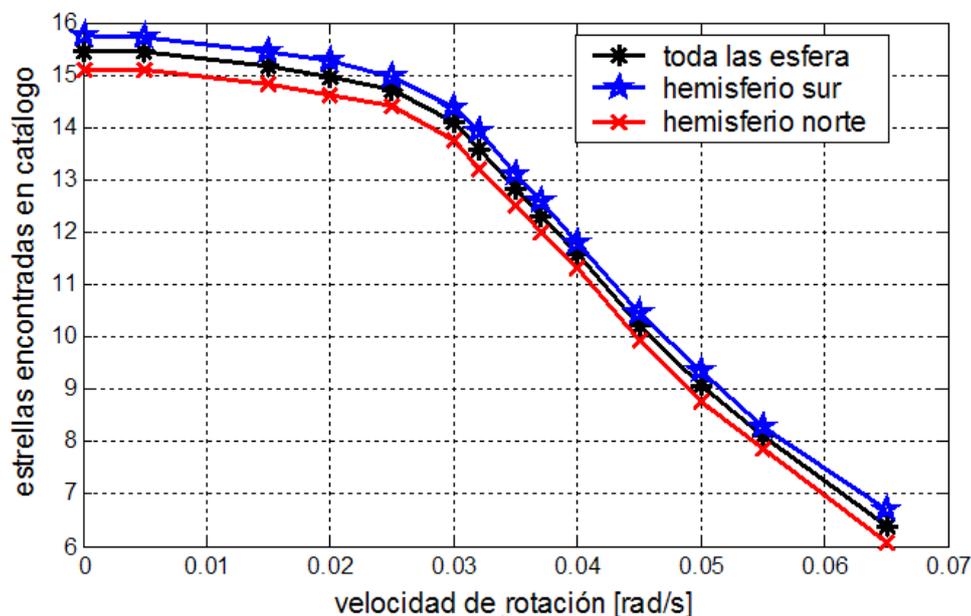


figura 4.19: \bar{N} versus ω para los dos hemisferios y para la esfera celeste completa.

⁵ Este tipo de satélites orbita en el plano del ecuador y su posición es constante respecto al sistema de referencia de la tierra. En cuanto a su orientación, solo gira respecto del eje de la tierra, por lo que hay dos caras que apuntan siempre hacia el norte y hacia el sur respectivamente. Es conveniente ubicar el ST en una de estas dos caras para evitar la presencia del sol y planetas en el FOV.

Se advierte que en hemisferio sur la cantidad de estrellas encontradas está por encima de las encontradas en el norte por un valor comprendido entre 0,45 y 0,75 dependiendo de la velocidad de giro.

Las figura 4.20 y figura 4.21 muestran P en función de ω .

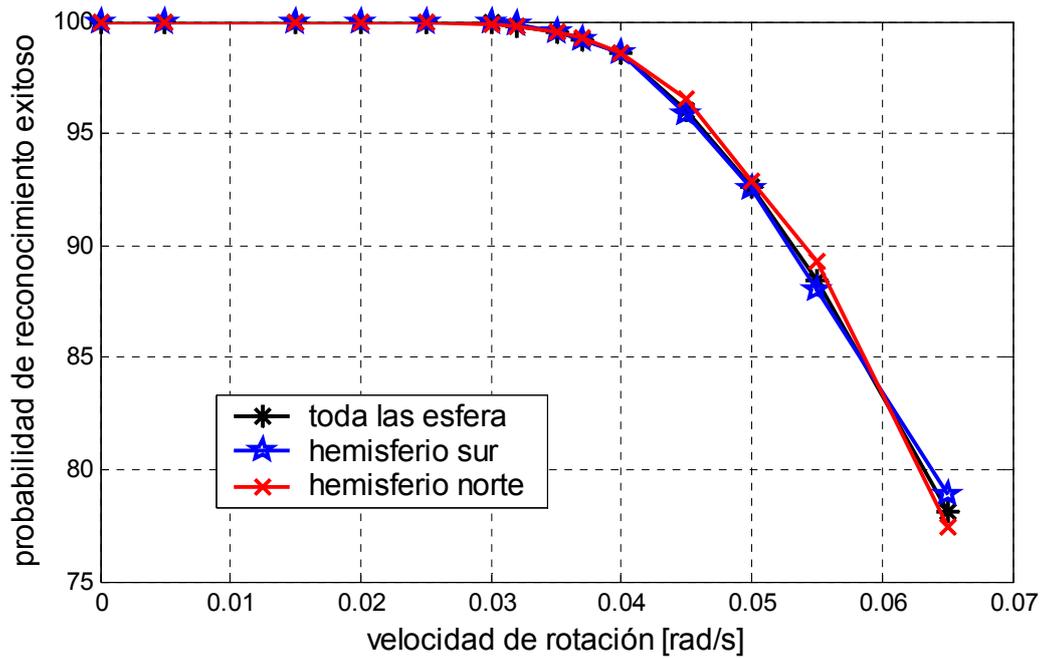


figura 4.20: P versus ω para los dos hemisferios y para la esfera celeste completa.

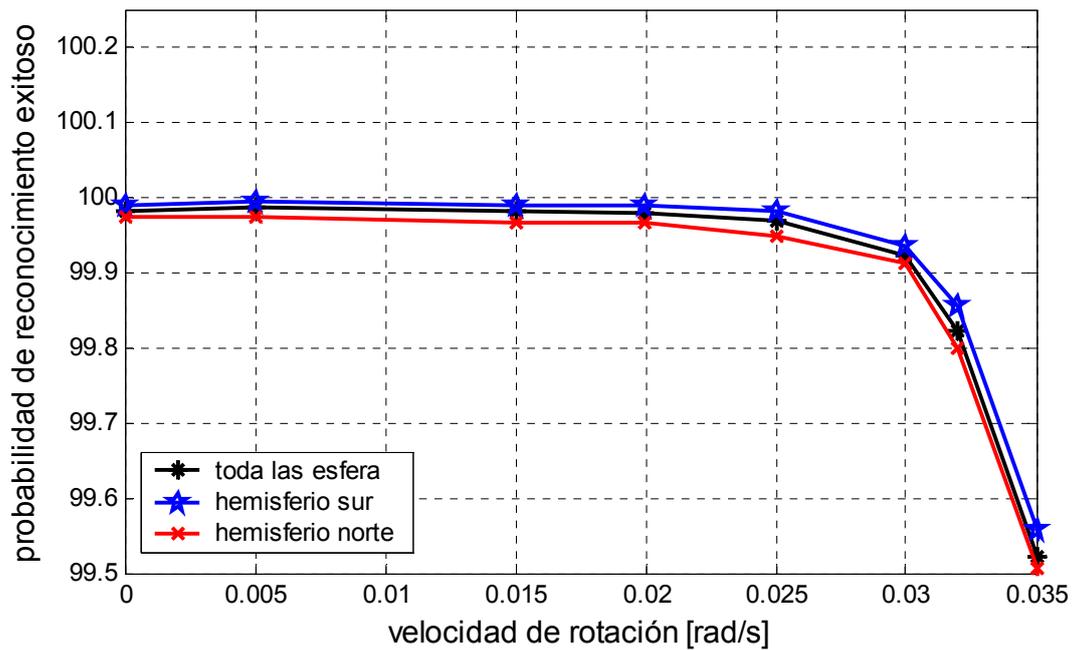
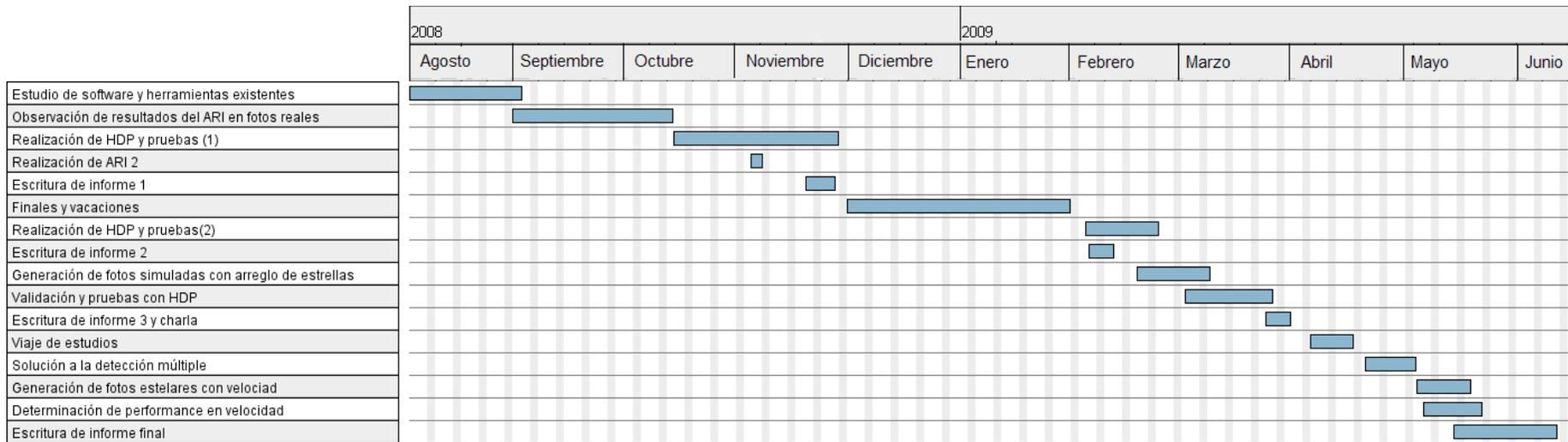


figura 4.21: P versus ω para los dos hemisferios y para la esfera celeste (ampliación).

Se ve que P es mayor en el hemisferio sur para bajas velocidades. Para $\omega = 0$, la probabilidad de reconocimiento exitoso para el hemisferio norte y sur es de 99,975% y 99,989% respectivamente. Esto es, la tasa de falla en el hemisferio norte es de 25 por millón mientras que en el hemisferio sur es de 11 por millón. La relación de tasa de falla norte-sur en este caso es de 2,3, valor que se mantiene entre 2 y 4 para $\omega < 0,025$.

Diagrama de Gantt del proyecto integrador



Conclusiones

Como conclusión general, se logró avanzar en el software del proyecto de star tracker, siendo éste el objetivo inicial del proyecto. A continuación sintetizamos las tareas realizadas a lo largo del proyecto integrador, y los resultados de las mismas.

En primer lugar, la observación de imágenes estelares reales y la visualización del funcionamiento del ARI en dichas imágenes permitió comprender varios aspectos del software. Paralelamente a esto se estudiaron y modificaron las herramientas de desarrollo ya existentes.

El desarrollo de la herramienta de prueba (HDP) fue de utilidad para probar el ARI en tiempos reducidos con un gran número de imágenes, de forma de tener suficiente estadística. A partir de los resultados obtenidos hasta este punto, se realizaron modificaciones en el ARI. Las pruebas realizadas por la HDP pusieron en evidencia una mejora en el reconocimiento de estrellas.

En una segunda etapa, se utilizaron imágenes con estrellas en movimiento. En este caso, las fotografías fueron simuladas mediante dos métodos. Se realizaron nuevamente modificaciones en el código del ARI para corregir fallas del mismo propias a las estrellas en movimiento.

Por último, se generaron 14 series de 1382 fotos cada una para la determinación de performance en función de la velocidad de rotación del satélite. Los resultados fueron gráficos de *cantidad de estrellas reconocidas* en función de *velocidad de rotación* y *probabilidad de reconocimiento de actitud* en función de *velocidad de rotación*. Se definió una velocidad máxima de operación de $0,03 \text{ rad/s}$, que corresponde a una probabilidad de reconocimiento de $99,9\%$. Se verificó que un star tracker tiene mayor probabilidad de reconocimiento de actitud cuando apunta hacia el hemisferio sur.

Anexos

Apéndice I: Programas utilizados

Los programas que se presentan a continuación fueron realizados por el grupo de guiado y control de INVAP a lo largo del desarrollo del software del ST. Son herramientas para la visualización de imágenes y resultados, validación de software, generación de imágenes y otras tareas relacionadas al desarrollo de los algoritmos de vuelo. Fueron ampliamente utilizados a lo largo de este proyecto integrador, por lo que se hará una descripción de cada uno de ellos, indicando en que parte del proyecto fueron de utilidad. Los programas se presentan en el orden en el que fueron usados durante el proyecto.

Ciclograf

Dado que el software de vuelo no entrega más información de salida que la orientación correspondiente a la imagen de entrada, es menester disponer de aplicaciones que permitan seguir la ejecución de las diferentes partes del código de vuelo. Con este fin, *ciclograf* aplica el algoritmo completo de determinación de actitud del ST a fotos con una interfaz gráfica que permite visualizar e interpretar los resultados de todas las etapas que componen el software. Luego de cada etapa se presenta información relativa al desenvolvimiento de esa fracción de software.

La consola principal de la interfaz gráfica se divide en un *área de presentación de información* y un *menú de botones* para abrir y reconocer una imagen o una serie de imágenes numeradas. La información presentada incluye: lista con centros encontrados (número de centro en orden decreciente de brillo, coordenadas x e y en la imagen y DNs totales del centro). La imagen se despliega en una ventana separada en la que a medida que se ejecuta el software se marcan automáticamente los centros encontrados por el ARI y los tríos encontrados en catálogo. El código fuente está escrito en C y se hace uso del programa *glade* para la interfaz gráfica.

En la imagen siguiente se muestra la interfaz gráfica del programa y una foto con estrellas identificadas a modo de ejemplo.

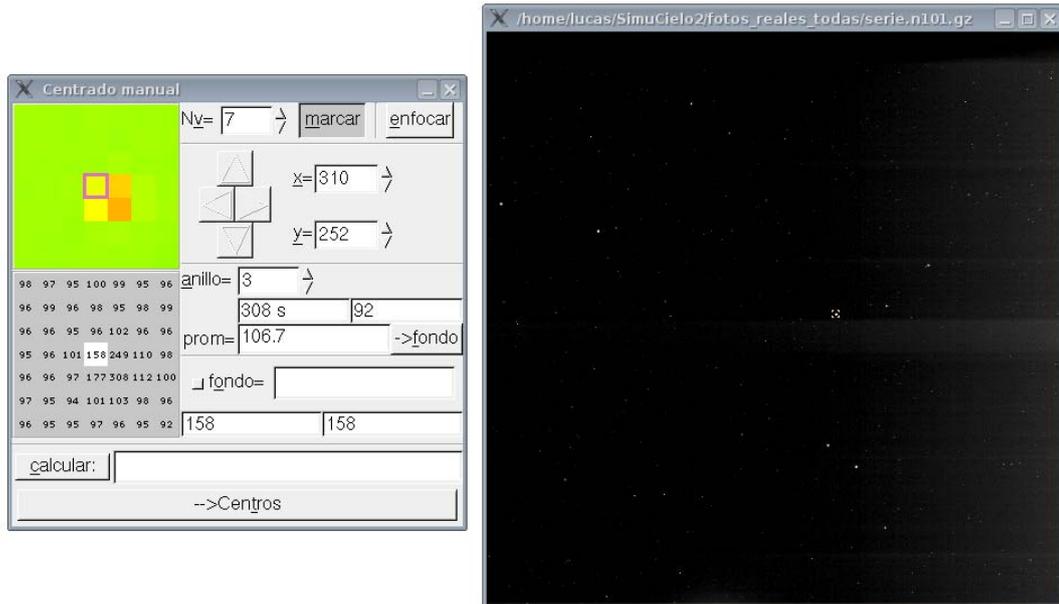


Interfaz gráfica de ciclograf.

Durante el trabajo se usó *ciclograf* intensivamente en la etapa de evaluación del ARI mediante inspección visual. Para esto se usó solo la parte relativa al reconocimiento de estrellas en la imagen, por lo que fue menester realizar modificaciones en el código fuente: se eliminaron los algoritmos de determinación de orientación posteriores al ARI.

Visor

Esta herramienta gráfica sirve para manipular y analizar imágenes estelares, realizando diferentes funciones sobre las mismas. La función mas importante de este programa para la realización de este proyecto es la visualización de partes de una imagen con aumento a nivel de píxeles. Una ventana cuadrada muestra un área de la foto de n píxeles de lado, con n variable entre 4 y 50. Las intensidades se presentan con escalas de colores y en forma numérica. En la imagen de abajo se ilustra la interfaz gráfica.

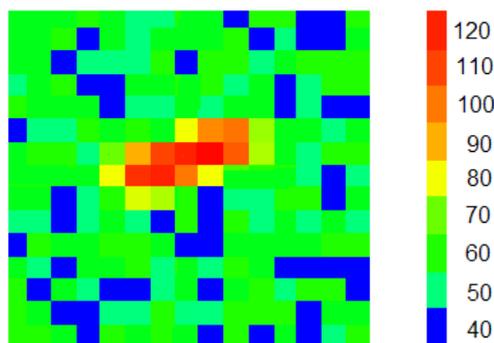


Interfaz gráfica de visor

Entre otras funciones y facilidades del programa se pueden mencionar:

- Variación del contraste de la imagen.
- Cálculo de *DN*s de una estrella en la ventana de visualización.
- Histograma de *DN*s de la foto completa.

La siguiente imagen presenta un ejemplo de la ampliación del mapa de pixeles.



Ventana de visualización de centros de visor, referencias en *DN*'s.

Por último, se menciona que *visor* otorga la opción de restar una imagen de fondo a la foto que se desea ver. El programa realiza una resta pixel a pixel entre la imagen que se desea analizar y la imagen de fondo que se quiere usar. Esto es, teniendo dos

fotografías estelares, se calcula, pixel por pixel, la resta de los *DN*. En el caso de que los *DN* calculados resulten negativos se le otorga el valor 0 a ese pixel.

Ciclo

Ciclo efectúa el reconocimiento de actitud con los algoritmos del ST sobre una serie de imágenes estelares. Es una versión de *ciclograf* sin interfaz gráfica. A grandes rasgos, el programa está compuesto por un código principal, el cual abre los archivos de las imágenes estelares de a uno, y llama a los algoritmos de reconocimiento de actitud del ST.

Para hacerlo funcionar se debe especificar la ruta de una carpeta donde se encuentran las imágenes estelares numeradas. La salida del programa son archivos con información acerca del reconocimiento. Se requirió modificar el código fuente para obtener como salida solamente la información relativa al ARI, esto es, las listas con las coordenadas y el brillo de las estrellas encontradas.

Simpix

Simpix genera imágenes estelares simuladas. Para ubicar las estrellas en las imágenes simulando un cielo real, hace uso del catálogo estelar XTE (X-ray Timing Explorer) de 32511 estrellas (ver Apéndice III: Catálogos estelares). *Simpix* genera tanto imágenes estáticas como en movimiento.

Toma como entrada los siguientes datos:

- Ruta y nombre del archivo imagen a crear.
- Posición estelar de la foto que se desea crear, medido en ascensión recta, declinación y rotación.
- Velocidad de giro de la imagen en cada uno de los tres ejes de la misma, en °/s.
- Nivel promedio de fondo, ruido y rampa de intensidades en el fondo.

Como salida, además del archivo imagen se obtiene un archivo con una lista de las estrellas presentes en la foto. Las columnas de las listas son:

Sky#	coordenada x (píxeles)	coordenada y (píxeles)	DN
------	------------------------	------------------------	----

Autosimpix

Esta rutina sistematiza el funcionamiento de *simpix* para generar una serie de fotos. En el código se ingresan la cantidad de fotos deseada, especificando la posición estelar y el vector de velocidad angular de cada una. El funcionamiento de este programa se basa en un bucle *for* que llama sucesivas veces a *simpix*. Se obtiene una serie de imágenes numeradas. Este programa fue usado en la sección 5.1.4 para realizar series de 1382 imágenes a distintas velocidades. Con este fin, se incluyó en el código una rutina que distribuye las imágenes uniformemente en la esfera celeste.

Apéndice II: Aplicaciones desarrolladas

En este apéndice se nombran las rutinas más importantes realizadas durante este trabajo, describiendo la función de cada una de ellas. Se dejan de lado pequeñas rutinas que no merecen una mención especial, así como las modificaciones en los códigos del software del ST y de las herramientas ya existentes al comenzar el proyecto. Los programas se presentan en el orden en el que fueron usados durante el proyecto.

La mayoría de los códigos fueron escritos en lenguaje C, usando el compilador *gcc* bajo Linux en una PC con microprocesador *Pentium 4*. Excepciones a esto son las rutinas que requieren herramientas gráficas, en estos casos se programó en código MATLAB, y los programas fueron corridos con MATLAB u OCTAVE (admiten el mismo código). Para las rutinas programadas en C con muchos archivos, se usó la herramienta de compilación *make* que corre en Linux.

Presentamos las aplicaciones y en el caso de tener muchos códigos, se menciona cada uno por separado:

Herramienta de prueba

Evalúa el ARI con estrellas reales, entregando los *FPs* y *VNs*. El principio de funcionamiento se expone en la sección 3.2. Fue realizado en C y se utilizó *make*. Los códigos que los componen son:

- ***ADR_test***: Es el código principal (main), para cada imagen llama a las siguientes rutinas.
- ***resta_fotos***: Toma pares de fotos consecutivas y genera una foto calculada como el módulo de resta pixel a pixel.
- ***picos***: Es el algoritmo de reconocimiento de imagen, reconoce estrellas en la imagen. Se lo modificó con respecto al del software del ST, para tener mucha sensibilidad (ΔDN bajo).
- ***recon_estre_dobles***: Reconoce pares de centros a la distancia vectorial conocida en las listas de centros que entrega *picos*. Entrega nuevas listas con estrellas que están en un par.
- ***compara_listas***: Compara las listas que entrega *recon_estre_dobles* con las listas del ARI que se desea probar.

Procesa datos ventanas

Esta rutina realiza el ajuste a estrellas reales con un modelo de distribución normal en dos dimensiones. Determina el parámetro σ para un conjunto de estrellas. Como dato de entrada, recibe matrices (mapas de píxeles) de las estrellas extraídas de fotos reales. Fue programado en MATLAB.

Genera estrellas

La rutina **genera_estrellas** programada en C crea archivos imagen con estrellas uniformemente distribuidas en el FOV.

Estrellas dobles

Esta rutina delata detecciones múltiples de estrellas (falla del ARI con estrellas en movimiento) en las imágenes producidas por *genera_estrellas*. Imprime en la terminal de *Linux* un arreglo de símbolos que indican en que estrellas de la imagen ocurrieron detecciones dobles.

Compara estrellas-centros

Esta herramienta mide la performance del ARI con una serie de fotos simuladas. En resumen, toma cada foto del conjunto, y cuenta la cantidad de estrellas que el ARI detecta en ella, siendo estrellas que figuran en el catálogo de vuelo. La salida de la herramienta es un archivo con datos numéricos para graficar un histograma. Los algoritmos fueron escritos en lenguaje C.

Los códigos que la componen son:

- **perf_vel:** Es el código principal de la herramienta. En primer lugar llama a *picos* (el código del ARI), el cual reconoce las estrellas en las imágenes. Luego llama a *est_cen* y por último a *confirma_en_cat*.
- **picos:** Es el algoritmo de reconocimiento de imagen, reconoce estrellas en las imágenes simuladas por *simpix*. Como siempre, entrega listas con las estrellas encontradas.

- **est_cen:** Compara las listas de salida de *picos* (estrellas reconocidas por el ARI) con las listas que entrega *simpix*. Recordar que las últimas tienen como dato el sky# de las estrellas de la foto. Este algoritmo entrega los sky# de las estrellas reconocidas por el ARI.
- **confirma_en_cat:** Para cada foto, toma la lista correspondiente entregada por *est_cen*. Luego se fija cuantos sky# de cada lista se encuentra en el catálogo de vuelo.

Histogramas

Esta aplicación realiza histogramas de cantidad de estrellas detectadas y gráficos de performance en velocidad (con los datos que entrega *Compara estrellas-centros*).

Se compone de dos rutinas realizadas en MATLAB:

- **procesa_histo:** Efectúa ajustes a las distribuciones normales de los histogramas, calcula P (área debajo de la curva normal para los valores de abscisa mayores que cuatro), y grafica los histogramas.
- **performance_vel:** Reúne los datos de todas las velocidades y grafica la N versus ω y P versus ω .

Apéndice III: Catálogos estelares

Los catálogos estelares base se consiguen de observatorios importantes, por ejemplo [4] o [5]. Un catálogo estelar de vuelo típico se construye a partir de un catálogo base seleccionando una cantidad reducida de estrellas que cubran toda la esfera celeste. El catálogo de vuelo almacena las estrellas en una memoria dentro del hardware del star tracker.

En nuestro ST, el catálogo base es el XTE (X-ray Timing Explorer, de 32511 estrellas). Dicho catálogo está compuesto por una lista en la que cada renglón le corresponde a una estrella. Entre otras cosas, la información dada para cada una es:

- Número de estrella, llamado número de SKYMAP (sky#).
- Posición en la esfera celeste (coordenadas x , y , z de un vector unitario⁶).
- Magnitud instrumental.
- Información sobre el movimiento de la estrella.

Las coordenadas cartesianas de las estrellas se miden en el sistema de coordenadas convencional inercial (CIS), ver Apéndice IV. El catálogo de vuelo resultante posee 3231 estrellas. En [3] se encuentra información detallada acerca de la preparación de éste catálogo.

⁶ Se utilizan coordenadas cartesianas en lugar de coordenadas celestes ya que si bien se requieren tres valores en lugar de dos para identificar un punto, se presenta la ventaja de requerir menor número de operaciones para cálculos como el de distancia entre dos estrellas.

Apéndice IV: Sistemas de coordenadas

Así como hay referencias adecuadas para ubicar puntos sobre la superficie terrestre, hay otros más apropiados para puntos en el espacio. Es importante conocer los sistemas de coordenadas asociados.

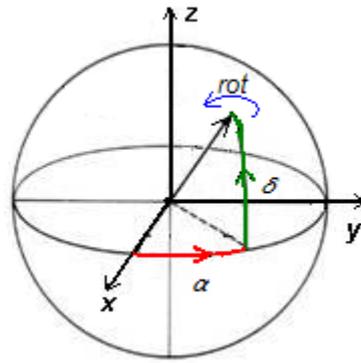
Sistema de coordenadas convencional inercial

En primer lugar, definiremos los ejes cartesianos en el sistema de referencia inercial de las estrellas. El sistema de coordenadas convencional inercial (Conventional Inertial System o CIS [6]) tiene como origen al centro de masa de la tierra. En el sentido estricto, el sistema CIS no es un sistema inercial porque el centro de masas de la Tierra sufre pequeñas aceleraciones cuando nuestro planeta se mueve alrededor del sol, razón por la cual a menudo es llamado cuasi-inercial. Para comprender la ubicación de los ejes x , y , z , consideramos las siguientes características del sistema:

- El plano $x - y$ es el plano del ecuador.
- El eje x tiene la dirección de la recta que se genera en la intersección entre el plano del ecuador y el plano del sistema solar (los planos forman un ángulo de $\sim 23^\circ$).
- El eje z , perpendicular a x e y , es el eje de rotación de la tierra y tiene el sentido tal, que forma una terna derecha.

Sistema de coordenadas ecuatorial

Para indicar actitud en el sistema inercial de coordenadas se utiliza ampliamente el sistema de coordenadas ecuatorial. En vez de definir una dirección mediante un vector en un sistema rectangular cartesiano, utiliza dos ángulos para ubicar a un punto en la esfera celeste, y un tercero para indicar rotación, como lo indica la siguiente figura:



Las coordenadas son:

- **Ascensión recta (α):** Ángulo entre el eje x y la proyección del vector sobre el plano x - y . Varía entre 0° y 360° .
- **Declinación (δ):** Ángulo entre el vector y el plano $x - y$. Varía entre -90° y 90° .
- **Rotación (*rot*):** El último ángulo indica la rotación en el eje del vector.

Las coordenadas de un vector expresadas en ascensión recta y declinación son datos útiles a la hora de ser interpretados por una persona, ya que tienen un sentido geométrico claro. No resulta ser un sistema muy adaptado a los cálculos, ya que se precisan muchas funciones trigonométricas.

Referencias

1. *Attitude dynamics and control*. s.l. : Wikipedia.
2. **Wertz, James R.** *Spacecraft Attitude Determination and Control*. s.l. : D. Reidel, 1978.
3. **Sferco, Juan Esteban.** *Determinación de orientación satelital en base a imagenes estelares*. 1999.
4. **R.Taylor, John.** *Error analysis*. 1997.
5. <http://fdd.gsfc.nasa.gov/attitude/skymap.html>.
6. <http://tdc-www.harvard.edu/catalogs/sao.html>.
7. <http://nacc.upc.es/tierra/node12.html>.

Índice de figuras

FIGURA 1.1: VISTA ESQUEMÁTICA EN CORTE (A) E IMAGEN (B) DE UN STAR TRACKER COMERCIAL.	- 8 -
FIGURA 1.2: VISTA EXPLOSIVA DE UN STAR TRACKER.	- 9 -
FIGURA 2.1: SISTEMA DE COORDENADAS Y ORDEN DE LOS PÍXELES EN LA FOTO ESTELAR	- 13 -
FIGURA 2.2: GRÁFICO DE BARRAS DE UNA ESTRELLA RELATIVAMENTE BRILLANTE.....	- 14 -
FIGURA 2.3: GRÁFICO DE BARRAS DE UNA ESTRELLA PEQUEÑA, CONFUNDIBLE CON EL FONDO.	- 15 -
FIGURA 2.4: RELACIÓN ENTRE NÚMERO DE ESTRELLA EN LA LISTA Y INTENSIDAD EN DN.	- 17 -
FIGURA 2.5: SECUENCIA REALIZADA POR EL ARI INICIAL AL RECORRER EL VECTOR DE DN	- 18 -
FIGURA 3.1: FOTO CALCULADA COMO EL MÓDULO DE LA RESTA DE DOS IMÁGENES.	- 23 -
FIGURA 3.2: DIAGRAMA DE ALGORITMOS Y ARCHIVOS USADOS.	- 24 -
FIGURA 3.3: ESQUEMA DE LA SECUENCIA REALIZADA POR EL ARI MODIFICADO.....	- 26 -
FIGURA 3.4: HISTOGRAMA DE PRIMEROS FPS, 450 FOTOS ESTÁTICAS, ARI 2.	- 28 -
FIGURA 3.5: HISTOGRAMA DE PRIMEROS VNs, 450 FOTOS ESTÁTICAS, ARI 2.	- 29 -
FIGURA 3.6: HISTOGRAMA DE PRIMEROS FPS, 450 FOTOS ESTÁTICAS, ARI 1.	- 29 -
FIGURA 3.7: HISTOGRAMA DE PRIMEROS VNs, 450 FOTOS ESTÁTICAS, ARI 1.	- 30 -
FIGURA 3.8: FPS DELATADOS POR HDP, CONJUNTOS DE POCOS PÍXELES LEVEMENTE ALTOS.	- 31 -
FIGURA 3.9: VNs DELATADOS POR LA HDP, CONJUNTOS DE POCOS PÍXELES LEVEMENTE ALTOS.	- 31 -
FIGURA 4.1: ESQUEMA DE LA FUNCIÓN (1)	- 36 -
FIGURA 4.2: ESTRELLA SIMULADA (A) Y UNA ESTRELLA REAL (B).....	- 39 -
FIGURA 4.3: ESTRELLA SIMULADA (A) Y ESTRELLA REAL (B), MAGNITUDES COMPARABLES.	- 39 -
FIGURA 4.4: FOTO SIMULADA CON EL PROGRAMA GENERADOR DE ESTRELLAS.	- 40 -
FIGURA 4.5: IMAGEN Y SUS EJES DE COORDENADAS	- 42 -
FIGURA 4.6: ESFERA CELESTE DE RADIO UNITARIO, SE INDICA EL ÁNGULO DE DECLINACIÓN δ	- 43 -
FIGURA 4.7: DISTRIBUCIÓN DE LAS 1382 FOTOS SIMULADAS.	- 43 -
FIGURA 4.8: RECONOCIMIENTO DE CENTROS EN LA FOTO DE LA FIGURA 4.4.....	- 44 -
FIGURA 4.9: SALIDA DEL PROGRAMA DE DETECCIÓN DE RECONOCIMIENTO MÚLTIPLE.....	- 45 -
FIGURA 4.10: FALLA DEL ARI, DOS ESTRELLAS RECONOCIDAS COMO DOS CENTROS CADA UNA.	- 45 -
FIGURA 4.11: ESTRELLA RECONOCIDA COMO TRES CENTROS.	- 46 -
FIGURA 4.12: EN ESTOS CASOS, LOS DOS CENTROS DE LA ESTRELLA SE ENCUENTRAN SEPARADOS.	- 46 -
FIGURA 4.13: PÍXELES INTERROGADOS POR EL ARI AL ENCONTRAR UNA ESTRELLA.....	- 46 -
FIGURA 4.14: HISTOGRAMA PARA $\omega = 0,04rads$	- 49 -
FIGURA 4.15: CONJUNTO DE HISTOGRAMAS PARA $\omega \leq 0.055rads$	- 49 -
FIGURA 4.16: N VERSUS ω	- 50 -
FIGURA 4.17: P VERSUS ω	- 51 -
FIGURA 4.18: P VERSUS ω (AMPLIACIÓN).....	- 51 -
FIGURA 4.19: N VERSUS ω PARA LOS DOS HEMISFERIOS Y PARA LA ESFERA CELESTE COMPLETA.	- 52 -
FIGURA 4.20: P VERSUS ω PARA LOS DOS HEMISFERIOS Y PARA LA ESFERA CELESTE COMPLETA.	- 53 -
FIGURA 4.21: P VERSUS ω PARA LOS DOS HEMISFERIOS Y PARA LA ESFERA CELESTE (AMPLIACIÓN).	- 54 -

Agradecimientos

Gracias a todos los que tienen algo que ver con el hecho de que estoy terminando este trabajo:

A mis padres, por darme, al crecer, la libertad para hacer muchas cosas, siempre con su consciente guía. Por dejarme solo en montones de situaciones y tareas, por la educación de primera, por la forma clara y madura de explicar las cosas, por “enseñarme a volar, pero no a volar sus vuelos”, por dejarme decidir que es bueno y que es malo, y por alentarme a escribir esta parte de la tesis.

A la gente de INVAP: Bepi, Paula, José, el otro José, Hernan, Esteban, Gabriel, Pabo, Andrés x2, Charly y mas. Gracias por los momentos agradables durante el trabajo, y tantos almuerzos llenos de risas.

Hermanos cuatro, gracias por ser tan buenos amigos y consejeros. De las opiniones tan diversas que tenemos se aprende mucho.

Gente del Balseiro, profesores y docentes en general, que saben transmitir sus conocimientos en un ambiente muy agradable.

Compañeros de estudio, gracias por ser mi familia durante los últimos tres años.

Gracias a mis amigos más fuertes de Bs As, Toto, Diego R., Marcos R., Erick M., por el aliento, la admiración mutua y el aprecio que no se degrada ni con los años ni con los kilómetros.