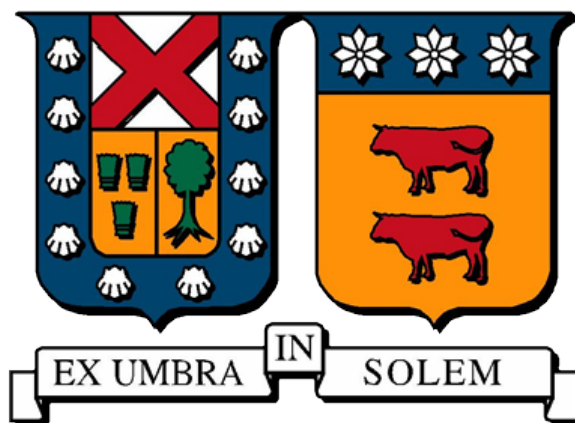


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
SANTIAGO – CHILE



“INDEXACIÓN DE OBJETOS ASTRONÓMICOS”

GABRIEL ANDRÉS CANDIA GALLARDO

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO CIVIL
INFORMÁTICO**

PROFESOR GUÍA:
PROFESOR CORREFERENTE:

MARCELO MENDOZA
MAURICIO SOLAR

ENERO – 2014

Agradecimientos

Termina un ciclo importante de mi vida, un ciclo que no estuvo exento de complicaciones. Muchas veces pensé que no sería capaz de lograrlo, pero en esos momentos siempre tuve una palabra de apoyo de mi círculo cercano.

Parto por agradecer a mi familia que siempre estuvo ahí cuando necesitaba ayuda o simplemente para darme una palabra de aliento. A mi hermana Macarena que siempre se preocupó de que estuviera bien y me ofreció ayuda a pesar de no entender mucho mis materias. A mi hermano Marcelo que me ayudó en algunas tareas o a estudiar ciertas materias y que también trasnocho en más de una ocasión junto a mi. A mis padres que nunca dudaron de mi y me apoyaron cada vez que tuve que levantarme temprano un día sábado para ir a dar una prueba o cuando trasnochaba por alguna prueba.

Agradecer también a mi polola Daniela que me ha visto crecer durante gran parte de mi estadía en la universidad, que siempre tuvo palabras de apoyo en los momentos más difíciles y tuvo que soportarme cuando algo no salía como lo esperaba.

Me siento agradecido también de la gente que conocí en estos 7 años en la universidad, grandes personas que me ayudaron a lo largo de toda la carrera y que estuvieron ahí para brindarme una palabra de aliento. No puedo dejar de mencionar a los grandes amigos que conocí durante esta etapa con los cuales mantengo un vínculo hasta el día de hoy, Emilio, Asdrúbal, Matías, Nicolás, Gonzalo, Rodrigo, Mariela, Gabriela. Darle las gracias al Limón por los consejos a la hora de tomar ramos. Además agradecer a tantos otros con los que perdí algo de contacto pero que cuando logramos vernos siempre pasamos un rato agradable, Alejo, Froko, Ian, Macarena, Juanxo, Vladi y tantos otros. Cada uno de ustedes fue un aporte en este ciclo y se los agradezco de todo corazón.

Agradezco también a cada uno de mis profesores, dado que cada uno contribuyó con un granito de arena. A mi profesor guía Marcelo Mendoza quién fue fundamental para llevar a cabo este proyecto al igual que mi profesor coreferente Mauricio Solar.

Agradezco también al proyecto de FONDEF D11I1060: “DESARROLLO DE UNA PLATAFORMA ASTROINFORMÁTICA PARA LA ADMINISTRACIÓN Y ANÁLISIS INTELIGENTE DE DATOS A GRAN ESCALA” que fue de mucho apoyo, en especial de los astrónomos.

Dedicado a mi abuelita Tita, a mis hermanos, a mis amigos, a mi polola Daniela y en especial a mis padres que sin el apoyo de cada uno, esto no hubiera sido posible.

Resumen

En el presente informe se muestran los pasos necesarios para generar un índice de objetos astronómicos. Para lograr este cometido, se utilizan técnicas de procesamiento de imágenes con el fin de obtener los objetos, extraer sus características y luego guardarlos en una base de datos. Esta base de datos permitirá consultas espaciales mediante coordenadas RA-DEC. Este índice de objetos astronómicos entregará una nueva herramienta a los astrónomos para realizar búsquedas por coordenadas de objetos en contraste con la búsqueda de imágenes por coordenada utilizada actualmente.

Palabras Clave: índice, astronomía, procesamiento de imágenes

Abstract

In this paper, we present the necessary steps to make an index of astronomical objects. To do this, some tools for image processing are necessary to obtain the objects present in the images, extract some of their characteristics and then store them on a data base. This data base will let the users do spatial search with RA-DEC coordinates. This will be a new tool for the astronomers that will allow make spatial search about objects and not about images, as it's done nowadays.

Keywords: index, astronomy, image processing

Tabla de Contenido

Índice de figuras	VIII
Introducción	XII
Capítulo 1 Antecedentes	16
1.1 Descripción del Problema	16
1.2 Objetivos	18
1.2.1 Objetivo Principal	18
1.2.2 Objetivos Específicos	18
1.3 Alcance	18
1.3.1 Procesamiento de imágenes	18
1.3.2 Base de datos	19
1.3.3 Índice de objetos	19
1.3.4 Página Web	19
Capítulo 2 Estado del Arte	20
2.1 Detección de Bordes	20
2.2 Segmentación	21
2.3 Reconocimiento de Patrones	22
2.4 De Imágenes a Catálogos	23
2.4.1 Estimación del Fondo	24
2.4.2 Convergencia	25
2.4.3 Detección	25
2.4.4 Combinación o Unión	26
2.4.5 Fotometría y Clasificación	26
2.5 Transformación de Wavelets	27
2.6 Modelo de Visión Multiescala	28
2.6.1 Definición del Modelo	29
2.6.2 Identificación de objetos mediante Coeficientes de Wavelet .	30
2.6.3 Reconstrucción Parcial	33
2.7 Herramientas para procesar imágenes	34

TABLA DE CONTENIDO

2.7.1	MATLAB	34
2.7.2	Lenguaje R	35
2.7.3	CASA	36
2.8	Observatorios Astronómicos Virtuales	39
Capítulo 3 Propuesta Metodológica		40
3.1	Descripción de la metodología	40
3.2	Procesamiento de Imágenes	41
3.2.1	Imágenes FITS	41
3.2.2	Sistema de Coordenadas	43
3.2.3	Pre Procesamiento	46
3.2.4	Extracción de características	51
3.3	Base de Datos	54
3.3.1	Elección de Base de Datos	56
3.3.2	Base de Datos Espacial	61
3.3.3	Datos Espaciales	62
3.4	Índice de Objetos	63
3.4.1	Acceso espacial	63
3.4.2	Consultas espaciales	69
3.5	Página Web	70
3.6	Resumen	72
Capítulo 4 Aplicación y validación de la Propuesta		74
4.1	Imágenes de prueba	74
4.2	Procesamiento de imágenes	75
4.2.1	Obtención de Coordenadas	75
4.2.2	Pre Procesamiento	78
4.2.3	Extracción de características	83
4.3	Base de Datos	84
4.4	Índice de objetos	86
4.4.1	Consultas espaciales	87
4.5	Página Web	88

TABLA DE CONTENIDO

4.6	Escalabilidad	90
4.6.1	Descarga de imágenes	90
4.6.2	Tiempo de Procesamiento de Consultas	93
Capítulo 5 Conclusiones		94
5.1	Respecto a la metodología	94
5.2	Respecto a los objetivos	95
5.3	Respecto al aprendizaje	96
5.4	Respecto a trabajos futuros	97
Bibliografía		99
A	Anexos	102
1.1	Códigos	102
1.2	Calibración	105
1.2.1	La Atmósfera	106
1.2.2	Parámetros de la Antena	109

Índice de figuras

2.1	Bordes obtenidos de Saturno.	21
2.2	Ejemplo de dato astronómico.	24
2.3	Ejemplo de conectividad en el espacio de wavelet.	31
2.4	Ejemplo de extracción de estrella.	34
2.5	Diagrama de flujo de CASA.	38
3.1	Ejemplo de un header de una imagen FITS.	42
3.2	Esfera celestial con sistema de coordenadas RA, DEC.	45
3.3	Imagen original, visualizada usando filtro en intensidad correspon- diente al rango de observación.	47
3.4	Histograma de intensidad.	48
3.5	Histograma de intensidad para la imagen binarizada.	50
3.6	Imagen binarizada.	51
3.7	Algoritmo de labeled connected components.	52
3.8	A cada componente se le calcula un convex hull para favorecer el cálculo de propiedades geométricas de los objetos.	53
3.9	Características geométricas de los objetos.	54
3.10	Ejemplo de kd-Tree en dos dimensiones.	65
3.11	Ejemplo de R-Tree en dos dimensiones.	66
3.12	Ejecutando una consulta en CasJobs.	71
3.13	Realizando una búsqueda por región.	71
3.14	Resultado de la consulta.	71
3.15	FITS obtenidos de la consulta.	72
4.1	Imagen de prueba a la cuál se le aplicará la metodología.	79
4.2	Threshold de la imagen.	81
4.3	Imagen binarizada y sin ruido.	82
4.4	Objetos etiquetados en la imagen.	82
4.5	Extracción de características de tipo shape.	83
4.6	Extracción de características de tipo moment para obtener las coor- denadas locales.	84

ÍNDICE DE FIGURAS

4.7	Tabla de objetos.	87
4.8	Generación de una región circular.	88
4.9	Página principal.	89
4.10	Consulta mediante región rectangular.	89
4.11	Consulta mediante región circular.	89
4.12	Despliegue de información mediante tabla.	90
4.13	Consulta para obtener imágenes FITS.	91
4.14	Objetos encontrados con la consulta.	92
4.15	Descarga de imágenes.	92
1.1	Transmisión de la data, con diferentes elementos presentes.	109
1.2	Haz principal con sus lóbulos laterales lejanos y cercanos.	112

Glosario

- **ALMA:** Acrónimo de Atacama Large Millimeter/submillimeter Array es un observatorio de radiointerferometría ubicado en el desierto de Chile.
- **CASA:** Acrónimo de Common Astronomy Software Applications es un software para procesar datos astronómicos.
- **CasJobs:** Sitio web utilizado para realizar consultas y obtener datos del observatorio SDSS.
- **Convex-hull:** Envoltorios conexos que sirven para identificar que pertenece al objeto encapsulado.
- **Excentricidad:** Grado de desviación que posee una sección cónica, es decir si puede identificarse como circular o no.
- **FITS:** Acrónimo de Flexible Image Transport System, es un formato de imágenes astronómicas utilizado por gran parte de los observatorios.
- **GiST:** Acrónimo de Generalized Search Tree es un árbol generalizado que permite adaptarse a distintos tipos de datos a indexar.
- **Header:** También conocido como cabecera para este caso particular se refiere a la información que contiene las imágenes FITS.
- **IAU:** Acrónimo de Unión Astronómica Internacional es una cooperación internacional de astrónomos alrededor del mundo.
- **Kernel:** Dícese del núcleo de una función y dependiendo de como sea, lo que se generará por usar la función.
- **LSST:** Acrónimo de Large Synoptic Survey Telescope es un observatorio que será construido en el norte de Chile. Realiza barridos del cielo, también conocidos como surveys.
- **Opening:** Herramienta de procesamiento de imágenes.

- **PSF:** Acrónimo de Point Spread Function algoritmo utilizado para identificar fuentes parciales como estrellas.
- **SDSS:** Acrónimo de Sloan Digital Sky Survey es un observatorio que realiza surveys del cielo.
- **Survey:** Barrido del cielo que abarca grandes porciones del espacio.
- **Threshold:** Identificación de un margen que permita diferenciar el fondo de la imagen de los objetos propiamente tal.

Introducción

La astronomía es la ciencia que estudia los objetos celestiales presentes en el espacio, esta ciencia es una de las más antiguas del mundo que en sus inicios se basaba en observar el cielo durante varias noches de forma metódica con el fin de obtener alguna característica o fenómeno en especial. Los inicios de la astronomía se asocian a la creación de los primeros telescopios que permitieron mejorar de forma considerable la visualización de los objetos celestiales.

En términos de astronomía, los cambios y mejoras en las tecnologías han sido considerables, en un principio esta ciencia se basaba netamente en la observación por lo que solo existían telescopios ópticos que se validaban mediante la observación humana. Hoy en día es imposible pasar todas las observaciones obtenidas de los distintos telescopios por el ojo humano, por lo que se hace necesario la creación de nuevas herramientas que permitan discriminar a priori que tipo de objetos o imagen es la que se obtuvo desde los telescopios.

Además de los telescopios ópticos utilizados en astronomía, se hizo necesario la creación de nuevos telescopios que permitan obtener datos más allá que solo la observación, un ejemplo son los telescopios de radio que permiten obtener características de los objetos en términos de longitud de onda, frecuencia, intensidad entre otros y que además permiten ver más allá del espectro óptico lo que permite estudiar objetos de menor tamaño o no visibles al espectro óptico.

Uno de estos nuevos telescopios de última generación fue instalado en Chile en el desierto de Atacama, es un telescopio de radio denominado ALMA¹ y es uno de los telescopios más grandes del mundo. La instalación de estos nuevos observatorios genera nuevos desafíos de transporte, almacenamiento y procesamiento de datos.

¹ Atacama Large Millimeter/submillimeter Array

Con estas nuevas tecnologías se hace necesaria la creación de nuevas herramientas que soporten estos tipos de datos, en algunos casos se puede realizar un post procesamiento de las imágenes para llevarlas al entorno óptico y luego poder procesarlas como ópticas, cabe destacar que una imagen de radio es equivalente a un cubo de imágenes, donde cada slice de este cubo es equivalente a una imagen óptica.

En la actualidad la astronomía está en un auge creciente que está generando el denominado “Tsunami de datos” que tiene que ver con la cantidad de datos generados por los telescopios que no es factible poder procesarlo de forma completa con las técnicas actuales, por lo que la creación de nuevas tecnologías que faciliten esta tarea son muy necesarias.

En esta memoria se busca generar una herramienta que permita realizar un procesamiento a las imágenes FITS con el fin de identificar los objetos astronómicos presentes en ella y guardarlos para que los astrónomos puedan utilizar esta base de datos y buscar objetos de interés en ciertas coordenadas astronómicas.

El desarrollo de este procesamiento de imágenes busca ser el primer paso para lograr generar una herramienta que pueda catalogar a priori los objetos presentes en las imágenes FITS, con el fin de generar un primer acercamiento hacia los astrónomos y realizar, por ejemplo, búsquedas de acuerdo a las características de un tipo de objeto que podría ser una galaxia o una estrella.

En el capítulo 1 se presentan los antecedentes del problema, en que consta el problema, sus alcances y cuáles son los objetos a lograr con el desarrollo de la solución. En el capítulo 2 se presenta el estado del arte con respecto a las técnicas y herramientas utilizadas para visualizar, procesar y manipular imágenes astronómicas, con el fin de saber que herramientas existen y se pueden utilizar que no es necesario volver a implementarlas sino usarlas. En el capítulo 3 se presenta la propuesta metodológica elegida para solucionar el problema, dentro de esta se explica

paso a paso los procedimientos para realizar una correcta indexación de los objetos. En el capítulo 4 se presenta la aplicación y validación de la metodología con la explicación de como se logró aplicar las herramientas y con imágenes para mostrar ejemplos. Dentro de los capítulos se separa en cuatro capas que son procesamiento de imágenes, base de datos, índice de datos y página web. Y para finalizar se presenta el capítulo número 5 donde se encuentran todas las conclusiones obtenidas mediante la realización del proyecto, así como también los posibles trabajos futuros.

Capítulo 1. Antecedentes

1.1. Descripción del Problema

Hoy en día la astronomía está experimentando un gran auge a nivel mundial y con esto se han generado muchos desafíos en distintos ámbitos. Con el mejoramiento de las tecnologías, las imágenes obtenidas por los observatorios cada vez son de mejor calidad y poseen mayor cantidad de información, lo que a su vez hace que su tamaño se haya ido incrementando considerablemente. Esto hace que todo tipo de manipulación, envío o procesamiento de estas sea más costoso en todo sentido lo que ha llevado a sobrepasar la capacidad de procesamiento y manipulación que el mundo de la astronomía puede soportar. Con esto es necesario la generación de nuevas herramientas que permitan procesar mayores cantidades de imágenes, segmentar de mejor manera las búsquedas que realicen los astrónomos, aumentar los tiempos de transmisión de cada imagen, crear nuevos centros de datos en distintos países entre otros.

El gran volumen de datos astronómicos que hay hoy en día es abrumador y eso sin contar que los nuevos observatorios que se están creando como por ejemplo ALMA o LSST harán crecer el volumen de datos de una forma considerable lo que hará temblar cualquier centro de datos o cualquier ancho de banda que se utilice actualmente.

Cada día es más difícil que los astrónomos puedan observar todas las imágenes obtenidas por los observatorios, por lo que grandes cantidades de imágenes serán archivadas y probablemente nunca sean estudiadas. Este dato es preocupante por el hecho de que importantes fenómenos se están perdiendo en estas imágenes y no tiene mucho sentido obtener cada vez más cantidades de imágenes con distintos filtros si la comunidad no dará abasto. Es por esto que surge la idea de discriminar a priori que tipos de imágenes podrían presentar objetos nuevos o fenómenos distintos a los que acostumbran ver los astrónomos.

Lograr hacer este filtro es una tarea compleja que posee varias aristas, en este proyecto se busca realizar el primer paso para lograr este cometido, que consta de identificar los objetos astronómicos presentes en las imágenes FITS obtenidas de los distintos observatorios con el fin de generar una gran base de datos con cada uno de estos objetos que permita realizar búsquedas de tipo espacial.

Para lograr esto es necesario realizar procesamiento de imágenes sobre la capa de imagen que poseen los archivos FITS y además se requiere procesar la capa del header que viene en el archivo con el fin de obtener la posición en el cielo que ocupa la imagen y así poder obtener las coordenadas de cada objeto presente en esta. Una vez obtenidos los objetos de la imagen con la posición que ocupan estos en el cielo es necesario guardarlos en una base de datos sobre la cual se puedan realizar consultas espaciales, para esto es necesario la creación de un índice que permita realizar estas consultas de una manera óptima. Esta base de datos además puede permitir guardar las características de estos objetos por si en un futuro se quisiera catalogar cada uno de ellos. El problema a resolver posee 4 aristas:

- **Procesamiento de imágenes:** Capa encargada de procesar y obtener los objetos astronómicos.
- **Base de Datos:** Capa encargada de guardar y permitir consultar acerca de estos objetos.
- **Índice de objetos:** Capa encargada de optimizar las búsquedas espaciales.
- **Visualización:** Capa encargada de mostrar los objetos obtenidos mediante una consulta en una página web que pueda ser accedida por los astrónomos.

1.2. Objetivos

1.2.1. Objetivo Principal

Diseñar e implementar una herramienta de software que permita generar un índice de imágenes astronómicas, basado en sus coordenadas espaciales en una base de datos acorde a las necesidades.

1.2.2. Objetivos Específicos

- Investigar sobre las imágenes FITS, con el fin de aprovechar toda la información contenida en la imagen incluyendo su metadata.
- Investigar y elegir una base de datos acorde para guardar los objetos astronómicos.
- Segmentar e identificar objetos que contengan las imágenes.
- Generar un índice de objetos astronómicos basado en sus coordenadas espaciales.

1.3. Alcance

Los alcances del proyecto se describirán en 4 áreas distintas que son procesamiento de imágenes, base de datos, índice de objetos, página web para consultas.

1.3.1. Procesamiento de imágenes

En el ámbito de procesamiento de imágenes los formatos a procesar serán FITS obtenidas de observatorios ópticos. A estas imágenes FITS se les realizará todo el procesamiento para la obtención de los distintos objetos presentes en ésta.

Los objetos encontrados en las imágenes FITS en un principio no se catalogarán, es decir, no se identificará que tipo de objeto es el que está presente, sólo se identificará su posición con alguna otra característica que se pueda extraer de su forma.

El lenguaje a utilizar para el procesamiento será R con el uso de librerías como FITSio para lectura y escritura de FITS, las librerías EBImage y biOps se utilizarán para el procesamiento de imágenes.

1.3.2. Base de datos

En el ámbito de base de datos los datos a archivar por cada objeto serán sus coordenadas RA y DEC como también algún tipo de extracción de características que se logre deducir, como podría ser su excentricidad, su radio mayor, entre otras. Se evaluará la mejor opción de base de datos que cumpla con los requerimientos para el correcto funcionamiento del índice.

1.3.3. Índice de objetos

El índice a utilizar para acceder a cada objeto será un R-Tree por lo que la base de datos utilizada debe permitir la implementación de éste. Este tipo de índice realiza búsquedas espaciales sobre una coordenada o región en específico. Las búsquedas de objetos en un principio serán solo mediante coordenadas RA y DEC, obteniendo como resultados todos los objetos presentes en una cierta área de interés como también poder acceder al FITS superior de donde se obtuvo el objeto.

1.3.4. Página Web

La página web será utilizada para realizar las consultas espaciales sobre la base de datos que posee los objetos previamente archivados e indexados. Será una interfaz simple que permita distintos tipos de consultas espaciales.

Capítulo 2. Estado del Arte

2.1. Detección de Bordes

Un borde es definido como una variación local de la intensidad de la imagen, algunos cambios de intensidad pueden ser muy grandes lo que facilita identificar de mejor manera los bordes, pero otros pueden ser muy suaves y ser difíciles de detectar.

El fin de esta detección es producir un dibujo de líneas desde una imagen más compleja, esto debido a que importantes conclusiones pueden obtenerse mediante solo los bordes de una imagen para luego dar paso a algunos algoritmos de computación que permitan la identificación de la imagen.

Algunos pasos a grandes rasgos que permiten la correcta detección de bordes se muestran a continuación [1]:

- *Suavizar*: Suprimir la mayor cantidad de distorsión posible, teniendo en cuenta la importancia de no suprimir los bordes.
- *Aumentar*: Aplicar un filtro que permita resaltar la calidad de los bordes de la imagen y agudizarlos.
- *Detectar*: Determinar que píxeles de borde deben ser descartados por ser ruido y cuáles deben ser mantenidos, puede variar mucho dependiendo del criterio de detección.
- *Localizar*: Determinar la posición exacta del borde, agudizar y conectar los bordes también forma parte de esta etapa.

En la *Figura 2.1* se muestra lo que se obtiene al utilizar un algoritmo de detección de bordes sobre una imagen de saturno.

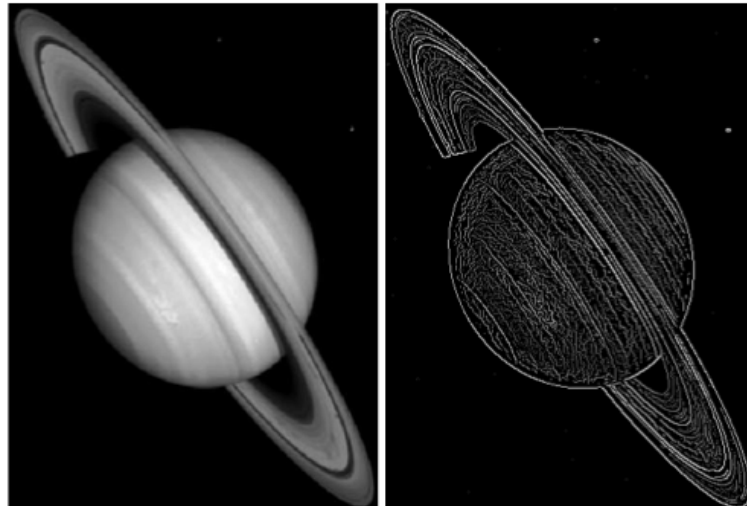


Figura 2.1: Bordes obtenidos de Saturno.

2.2. Segmentación

La segmentación de imágenes es un proceso en el cual se particiona una imagen en regiones o segmentos basado en similitudes dentro de la región o diferencias entre regiones. Una imagen representa una escena en donde participan distintos objetos o regiones, el proceso de separación puede ser difícil de automatizar.

Existen tres acercamientos generales acerca de la segmentación de imágenes:

- *Clasificación por pixel*: Técnica que evalúa pixel por pixel la posición donde se encuentra y que nivel de gris o color (dependiendo el tipo de imagen) posee. Se usa el término clasificación por que se considera que las distintas regiones poseen píxeles de diferentes clases.
- *Método basado en límites*: Técnica que detecta los límites de las regiones, los píxeles encerrados en un margen pueden ser categorizados acorde a éste.
- *Método de crecimiento de regiones*: Técnica basada en la identificación de grupos de píxeles de valores similares, este agrupamiento se realiza generalmente de forma iterativa y permite una relajación acorde al caso.

Existen muchos otros métodos para segmentar una imagen dado que no existe un consenso en esta área de procesamiento de imágenes.

2.3. Reconocimiento de Patrones

El reconocimiento de patrones es un área de estudio donde se busca tomar decisiones de manera automática, por lo general se toma una imagen en forma de matriz donde se revisa si los vectores o columnas cumplen con algún criterio de selección. Dado que es necesario conocer que número de coincidencias posee el vector en estudio con el criterio, se dice que los patrones son una clase de clasificadores, en esta clasificación es donde el creador del reconocimiento de patrones especifica que porcentaje de similaridad debe poseer la imagen o señal en estudio para clasificarla como tal.

En gran parte de los casos de estudio es necesario realizar un arreglo de la imagen o de parte de ella con el fin de reducir los ruidos o distorsiones que ésta pueda tener, dado que podría llevar a conclusiones erróneas si no se procesa la imagen de forma correcta. En general de la imagen se obtiene una matriz con información donde de ésta se deriva a otra del mismo estilo pero dejando de lado toda la información superflua o detalles irrelevantes. Este proceso es denominado “*feature extraction*”[2].

Los componentes de un vector que posee un patrón son comúnmente llamados *features*, características o atributos. Para una correcta extracción de estas características existen dos tipos de enfoque, extracción de bajo nivel y de alto nivel.

La extracción de bajo nivel corresponde por lo general a aplicar una transformación al vector o matriz en estudio para potenciar la separación de clases o reducir la dimensión de la información, esto con el fin de hacer las tareas computacionales más manejables. Algunos esquemas de compresión también son denominados extracción de bajo nivel. Algunos ejemplos de transformaciones usadas son la transformada de *Fourier* y la transformada de *Karhunen-Loève*.

Por otra parte la extracción de alto nivel tiene como objetivo ayudar de una for-

mas más evidente a sacar conclusiones, un ejemplo podría ser el reconocer formas de una imagen donde el vector podría contener información como largo, grosor, entre otros. La extracción de alto nivel es más tediosa que la de bajo nivel y utiliza herramientas más poderosas.

La finalidad de la extracción de características es describir y discriminar los atributos del patrón como también hacer un clasificador más simple. Dentro de los clasificadores se pueden dividir en dos grupos, los supervisados y los no supervisados. El clasificador supervisado posee un mayor conocimiento sobre el objeto en estudio y utiliza un conjunto de parámetros de entrenamiento que definen el clasificador, por otra parte, los clasificadores no supervisados se tiene un menor conocimiento del tema y un ejemplo podría ser la segmentación de imágenes.

Cuando nos referimos a un clasificador hay que tener en cuenta que maneja dos tipos distintos de datos, datos de entrenamiento y de prueba. Los datos de entrenamiento permiten calibrar el clasificador y fijar sus parámetros, por otra parte, los datos de prueba son los que permiten revisar si es que el clasificador entrenado sirve y se puede utilizar con nuevos datos.

2.4. De Imágenes a Catálogos

Extraer información de las imágenes es un paso fundamental para los astrónomos. Por ejemplo para construir catálogos, es necesario identificar correctamente estrellas y galaxias, tanto como realizar una estimación precisa sobre su posición y fotometría ¹. Varios métodos han sido propuestos para lograr esos resultados. Uno de los softwares más usados para lograr este cometido es SExtractor creado por Bertin and Arnouts en 1996 [3]. Posee la habilidad de manejar grandes imágenes (de hasta 60000 x 60000 píxeles) y lograr buenos resultados, por lo general los software incluido SExtractor realiza los siguientes pasos:

¹Rama de la astronomía que se dedica a medir el brillo de los distintos astros.

1. Estimación del Fondo
2. Convergencia
3. Detección
4. Combinación
5. Fotometría y Clasificación

En la *Figura 2.2* se muestra un ejemplo de dato astronómico, donde se observa una fuente particular y otra fuente extensa, además de poseer ruido y un fondo. El objeto extendido que es detectado por el ojo humano podría no ser detectado por un mecanismo de detección estándar.

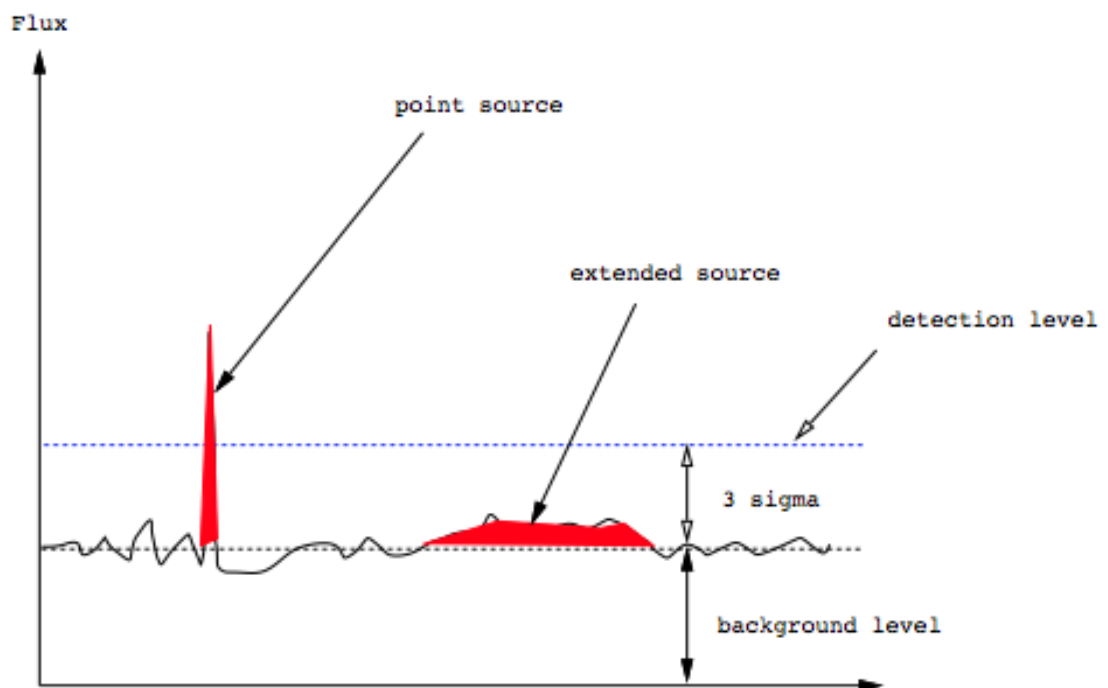


Figura 2.2: Ejemplo de dato astronómico.

2.4.1. Estimación del Fondo

Por lo general los objetos en estudio pasan desapercibidos por pertenecer a una señal plana, a esto se le llama señal de fondo. En los estudios del universo pequeñas variaciones del fondo modifican de manera importante la intensidad y los diámetros

de los diferentes astros, para obtener buenos resultados es fundamental obtener la magnitud correcta en cada punto del fondo del cielo. En algunos píxeles la medición de intensidad puede ser menor al fondo del cielo, esto es debido a la granularidad de las emulsiones provenientes del cielo. Bajo el caso anterior, se hace casi imposible determinar directamente la magnitud que posee el fondo del cielo.

Muchos métodos tratan de solucionar el problema del fondo, algunos de estos tienen el defecto de usar información de los píxeles cercanos lo que lleva a una pérdida de resolución. En general el mayor problema es identificar las diferencias entre las medidas de intensidad y la verdadera intensidad, donde es vital tener en cuenta la granularidad [4].

2.4.2. Convergencia

Para optimizar la detección es necesario que la imagen se haga convergir mediante algún filtro. La forma del filtro optimiza la detección de objetos de forma similar, por esto mismo, para detectar estrellas el filtro óptimo es PSF¹, para objetos extensos, se requieren filtros más largos. De esta forma para detectar objetos de diferentes formas, se debe probar la detección con varios tamaños de filtro, tendiendo a un acercamiento multiescala.

2.4.3. Detección

Luego de que la imagen ha convergido, es decir los puntos significantes de un objeto han sido identificados, se segmenta la imagen mediante su *thresholded*² asignando a cada grupo de píxeles conectados a una etiqueta, para luego separar los objetos conectados y que poseen la misma etiqueta.

Una alternativa a usar la técnica de *thresholded* es detectar los montes de la imagen, cabe destacar que esta técnica funciona bien para detectar estrellas pero no

¹ Point Spread Function describe la respuesta de una imagen a un punto objetivo.

² Método de segmentación que utiliza escala de grises para identificar bordes.

para objetos más extensos. Para el caso de objetos extensos el siguiente paso sería unir los píxeles que pertenecen al mismo objeto.

2.4.4. Combinación o Unión

Este es el paso más delicado, los objetos extensos deben ser considerados como un solo objeto y los múltiples objetos deben ser bien separados. En el software SExtractor cada grupo de píxeles conectados es analizado a distintas intensidades, partiendo con una intensidad alta y luego disminuyendo. El grupo de píxeles puede ser visto como una superficie con valles y montañas. Al comienzo, solo las grandes cumbres son visibles y a medida que va disminuyendo otros objetos comienzan a ser visibles, definiendo varias estructuras. A un nivel determinado dos objetos pueden parecer conectados, ahí es donde se debe tomar una decisión si unirlos o dejarlos separados. La forma de evaluarlo es comparando las intensidades dentro de las cumbres, si el radio entre ellas es muy bajo, se unen. De lo contrario se dejan como objetos independientes.

2.4.5. Fotometría y Clasificación

Existen varios métodos que permiten calcular la fotometría de los objetos detectados. Un fotómetro con apertura adaptativa usa la primera imagen para determinar de que forma se obtiene mayor cantidad de flujo. *Kron* propone una apertura del doble de la primera imagen tomada, que lleva a una recuperación de flujo mayor al 90 % [2]. Dependiendo los objetos a clasificar que métodos se utilizarán, por ejemplo cuando una imagen contiene solo estrellas se especifica un método que puede contener la técnica PSF.

Cuando se quiere clasificar una estrella o galaxia hay una serie de parámetros que se han tenido en cuenta para hacerlo, como por ejemplo, máxima intensidad, área radial, densidad de gradiente, momentos invariantes, por mencionar algunas. Dada la gran cantidad de parámetros que hay para elegir existen diferentes puntos de vista o enfoques, como la calidad de los datos, la facilidad que posee la compu-

tación para medir ciertos parámetros, la relevancia e importancia de los parámetros dependiendo el tipo de salida que se quiere lograr, van gatillando que parámetros usar.

Desde el punto de vista de algoritmos multivariados, se necesita una muestra homogénea de parámetros. Sabiendo esto y conociendo la gran cantidad de métodos para clasificar una galaxia, dos enfoques para seleccionar parámetros parecen ser fuertemente representados.

1. La luminosidad de los objetos a lo largo de los ejes mayores se determina en intervalos discretos. Esto puede realizarse rellenando los contornos elípticos y estudiando la luz de los anillos elípticos ¹. Para lograr ver el perfil de los radios presentes en la galaxia se puede asumir que la galaxia es simétrica con los ejes en estudio y que puede ser ajustada dependiendo los radios en estudio.
2. Otra opción para determinar el perfil es definir la morfología de los parámetros. Puede ser interesante utilizar un modelo que logre definir una galaxia, medir el brillo de la superficie, estudiar el diámetro logarítmico que posee la galaxia, entre otros con el fin de definir futuros parámetros a ser utilizados.

El rango de parámetros a utilizar debe ser consecuente con la complejidad del objeto en estudio. Los parámetros pueden ser derivados de un perfil, pero los parámetros no pueden definir un perfil a estudiar.

2.5. Transformación de Wavelets

La Transformación de Wavelets permite separar la imagen en varios componentes, los cuales se denominan wavelets de la imagen. Una imagen de una galaxia puede estar formada por las wavelets más amplias, donde en gran parte puede estar compuesta por píxeles negros pertenecientes al fondo. Pero también hay wavelets con mayor detalle y nitidez que pueden pertenecer a una estrella de la galaxia. La

¹ Método utilizado por *Lefèvre en 1986*.

imagen en estudio puede ser descompuesta en la cantidad de wavelets que se requiera dependiendo los detalles esperados. Cada wavelets no es una imagen en sí, pero varias wavelets pueden recrear una, de hecho se pueden identificar las wavelets que poseen mayor cantidad de información y utilizarlas para recrear una imagen, dejando de lado algunas wavelets que puedan tener información irrelevante, de esta forma la transformación de wavelets también sirve para comprimir imágenes [5].

Para aplicar la transformación de wavelets a una imagen, se requiere tener la imagen en su forma matricial. En la matriz de la imagen se encuentran los píxeles de ésta, donde sus valores representan la oscuridad de un pixel, siendo 0 el blanco y el valor más alto el negro. Los píxeles de valor central pertenecen a los grises.

Al aplicar la transformación de wavelets a la matriz genera tantos coeficientes como píxeles posee la imagen. Estos coeficientes son más fáciles de comprimir dado que la información está concentrada en sólo algunos de éstos. A su vez a estos coeficientes se les puede aplicar un *threshold* donde todos los coeficientes bajo éste se cambian por 0, dado que son coeficientes que no poseen información importante.

El uso de algunos de los métodos de la transformación de wavelets serán explicados cuando se muestre su uso.

2.6. Modelo de Visión Multiescala

La transformación multiescala de una imagen produce una matriz con tantos coeficientes como píxeles posea la imagen. La imagen original puede ser expresada por la suma de todas las escalas wavelets más la suma de un arreglo suavizado como se muestra en la *ecuación 2.1*:

$$I(k, l) = C_{J,k,l} + \sum_{j=1}^J W_{j,k,l} \quad (2.1)$$

Como la imagen es interpretada como una representación multiescala de píxeles se debe considerar poder representar los objetos presentes en ésta, es decir identificar la representación multiescala del objeto. Claramente la representación depende de la imagen a analizar. Un modelo para representar datos astronómicos es asumir que la imagen I puede ser descompuesta en varios componentes:

$$I(k, l) = \sum_{i=1}^{N_o} O_i(k, l) + B(k, l) + N(k, l) \quad (2.2)$$

En la ecuación 2.2 N_o es el número de objetos, O_i son los objetos presentes en la información (estrellas, galaxias, etc), B es el fondo o background y N el ruido. Para lograr esta descomposición es necesario extraer de forma correcta toda la información de la imagen. Se analiza la representación multiescala y se va segmentando escala por escala. En general para poder identificar un objeto se analiza su conectividad, dado que ocupa una región física, y en esta región se pueden unir píxeles basados en la adyacencia de ellos. El espacio Wavelet de una imagen 2D queda representado en un espacio 3D, por lo que las conexiones en 2D deben ser transformadas a WTS¹. De esta forma para identificar los objetos debemos analizar los píxeles presentes en el WTS.

2.6.1. Definición del Modelo

El Modelo de Visión Multiescala (MVM), describe un objeto mediante un set de estructuras jerárquicas y usa las siguientes definiciones:

- *Coefficientes Wavelets Significantes*: Son significantes cuando sobrepasan un umbral dado dependiendo el ruido que posea la imagen.
- *Estructura*: Una estructura $S_{j,k}$ es un set de conexiones de coeficientes significantes en la misma escala j , donde p es el número de coeficientes significantes en la estructura y W_{j,x_i,y_i} es el coeficiente de wavelet en la escala i en la posi-

¹Wavelet Transform Space.

ción (X_i, Y_i) , como muestra la *ecuación 2.3*:

$$S_{j,k} = \{W_{j,x_1,y_1}, W_{j,x_2,y_2}, \dots, W_{j,x_p,y_p}\} \quad (2.3)$$

- *Objeto*: Un objeto es un set de estructuras, se define el operador L el cual indica a que objeto pertenece la estructura, si pertenece da igual a l de otra forma a 0.

$$O_l = \{S_{j_1,k_1}, \dots, S_{j_n,k_n}\} \quad (2.4)$$

- *Escala de Objeto*: La escala del objeto está dada por su máximo coeficiente de wavelet.
- *Relación interescala*: Criterio que permite conectar dos estructuras en un mismo objeto.
- *Sub-Objeto*: Aparece cuando un objeto posee un máximo de wavelet local. Por lo que puede ser compuesto por varios sub-objetos. Cada sub-objeto puede ser analizado.

2.6.2. Identificación de objetos mediante Coeficientes de Wavelet

Multiresolución soporta segmentación. Al poseer la imagen en multiresolución se tiene a cada escala una imagen booleana, es decir cuando los coeficientes son significantes son igual a 1, de lo contrario a 0. La segmentación consiste en etiquetar las escalas booleanas. Cada grupo de píxeles conectados obtiene una etiqueta entre 1 y L_{max} , donde L_{max} es el número de grupos. Este proceso se repite en cada escala de multiresolución. Se define una estructura $S_{j,i}$ como el grupo de conexión de los píxeles significantes con la etiqueta i en la escala j . En la *figura 2.3* se muestra la conectividad en el espacio de wavelet; coeficientes significantes de wavelet que aparecen contiguos forman una estructura y mediante una relación interescalar un set de estructuras forman un objeto.

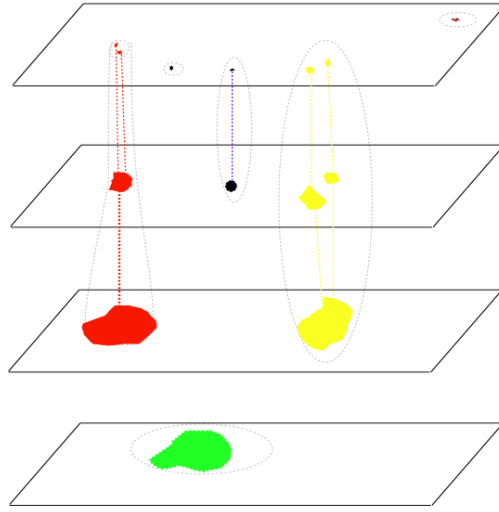


Figura 2.3: Ejemplo de conectividad en el espacio de wavelet.

Grafo de conectividad interescala Un objeto es definido como un set de estructuras jerárquicas. La regla que permite la conexión entre dos estructuras es llamada “*relación interescala*”. En la figura se muestra cómo estructuras de diferentes escalas son unidas y forman objetos. Considerando dos estructuras sucesivas, $S_{j,k}$ y $S_{j+1,l}$. Cada estructura está localizada en parte de la descomposición y corresponden a una región de la imagen donde los coeficientes son significantes. El pixel que posee el coeficiente de wavelet máximo debe estar presente en ambas estructuras para poder conectarlas. Las estructuras que aparecen en imágenes con coeficientes de wavelets sucesivos pueden ser conectados y se denominan objetos pertenecientes al grafo de conectividad interescala.

Filtro. Estadísticamente algunas estructuras pueden ser generadas debido al ruido. Por lo general contienen pocos píxeles y están aislados, por lo que para evitar detecciones falsas los campos aislados pueden ser removidos del grafo de conexión interescalar. Las estructuras detectadas en los bordes de la imagen pueden ser generadas por problemas en el borde y también pueden ser removidas.

Unión o Separación. Objetos que se encuentran muy cerca pueden estar conectados o ser diferentes, en estos casos una decisión debe ser tomada. En el espacio de wavelet la decisión de unir o separar está basada en los máximos locales de las

diferentes estructuras que pertenecen al objeto. Un objeto nuevo es derivado de una estructura si existe al menos otra estructura en la misma escala perteneciente al mismo objeto y cumple con lo siguiente:

- El pixel anterior al máximo de wavelet de la estructura tiene valor 0 y no está conectado a ninguna estructura de una escala menor.
- El máximo coeficiente de wavelet de la estructura $S_{j-1,l}$ es el más cercano al máximo de la estructura $S_{j,k}$.
- Los coeficientes máximos de la estructura $S_{j+1,k}$ pertenecen también a la estructura $S_{j,k}$.

Cuando se cumplen estas condiciones la estructura $S_{j,k}$ y todas las estructuras en escalas menores que están directa o indirectamente conectadas con $S_{j,k}$ definirán un nuevo objeto.

Identificación de Objetos. Resumiendo el método se pueden identificar los objetos presentes en la imagen I .

1. Se aplica la transformación de wavelet $W = WI = \{w_1, \dots, w_j, c_j\}$ que genera el espacio de wavelet del mismo tamaño al de la imagen original.
2. Se determina la desviación estándar del ruido en w_1 .
3. Se deduce el *threshold* en cada escala desde el modelado de ruido.
4. Se determinan las relaciones interescalares.
5. Se determinan los máximos coeficientes de wavelet en el WTS.
6. Se extraen todos los árboles resultantes conectados a los máximos de cada WTS.

2.6.3. Reconstrucción Parcial

Reconstrucción parcial como un problema inverso. Un set de estructuras S_i donde $S_i = \{S_{j,k}, \dots, S_{j',k'}\}$ definen un objeto O_i que puede ser reconstruido separadamente desde otros objetos. Se denota W_i como el set de coeficientes de wavelet pertenecientes al objeto O_i . Por lo tanto W_i es un subset de la transformación de wavelet de O_i , $\tilde{W}_i = WO_i$. De hecho, la última escala de \tilde{W}_i es desconocida, así como también otros coeficientes de wavelet que no han sido detectados. El problema de reconstrucción consiste en buscar una imagen O_i tal que su transformación de wavelet reproduce los coeficientes W_i es decir pertenecen a las mismas estructuras S_i . Si W describe el operador de transformación de wavelet y P_w el operador de proyección en el subespacio de los coeficientes detectados, es decir los coeficientes donde no se detectó nada se toman como valor 0. La solución se busca minimizando la *ecuación 2.5*.

$$J(O_i) = \|W_i - A(O_i)\| \quad (2.5)$$

Se debe resolver el problema inverso que consiste en determinar O_i conociendo A y W_i . El tamaño de la imagen restaurada es arbitrario y puede ser mayor que los coeficientes conocidos. Por lo general la solución a la *ecuación 2.5* posee infinitas soluciones, depende de la naturaleza de la imagen.

Algoritmos de reconstrucción. El método de mínimos cuadrados puede ser utilizada para resolver la relación $W_i = A(O_i)$ que lleva a buscar la imagen O_i que minimiza la distancia $\|W_i - A(O_i)\|$. $\|W_i - A(O_i)\|$ es mínimo si y solo si O_i es una solución de la *ecuación 2.6*.

$$\tilde{A}(W_i) = (\tilde{A} \circ A)(O_i) \quad (2.6)$$

W^{-1} es el operador de reconstrucción de wavelet. De una estructura de wavelete W_i , una imagen O_i es restaurada mediante la suma de las escalas de wavelet y la última imagen suavizada. W_i no es necesariamente la transformación de wavelet de una imagen, por lo que $W^{-1}WO_i$ puede no representar la imagen.

Un algoritmo de reconstrucción puede descartar las capas que posean información irrelevante y aplicar la transformada inversa de wavelet para reconstruir la imagen con algunas capas.

Ejemplo: Extracción de estrella de NGC2997. En el ejemplo se aplica MVM a la galaxia NGC2997. Dos imágenes fueron creadas sumando objetos detectados de la primera y segunda escala, y desde la tercera a la sexta escala. Se muestran en la *figura 2.4* respectivamente arriba a la derecha y abajo a la izquierda. La imagen de arriba a la izquierda muestra la imagen original y la imagen ubicada a la derecha abajo muestra la diferencia entre la imagen original y la imagen que contiene los objetos de las escalas 1 y 2. Como se puede ver, al remover los pequeños objetos la galaxia puede ser mejor analizada.

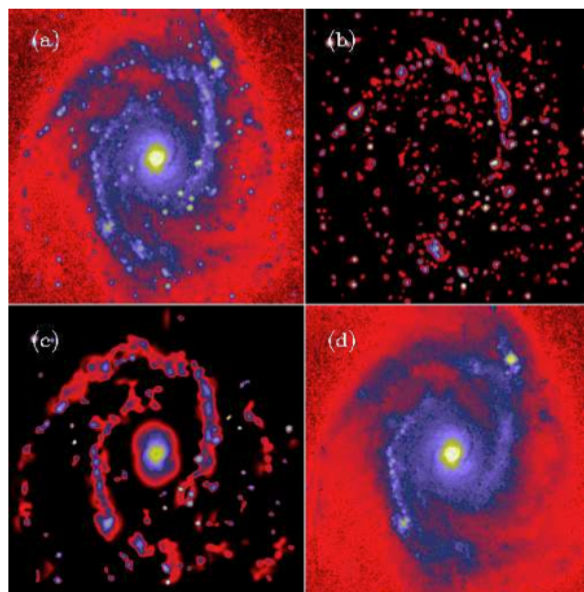


Figura 2.4: Ejemplo de extracción de estrella.

2.7. Herramientas para procesar imágenes

2.7.1. MATLAB

MATLAB viene de *Matrix Laboratory* y es un software que posee un lenguaje de alto nivel en un ambiente interactivo que permite realizar intensas tareas de

computación de mejor manera que los lenguajes de programación tradicionales. Su enfoque principal es facilitar el desarrollo de las operaciones matriciales.

MATLAB a su vez también posee una caja de herramientas destinadas al procesamiento de imágenes y tiene una gran variedad de operaciones que pueden ser utilizadas:

- Operaciones geométricas.
- Operaciones por bloque y por vecinos.
- Filtros lineales y filtros especiales.
- Transformaciones (Fourier, Wavelet, entre otras).
- Análisis de imágenes y aumento.
- Operación sobre imágenes binarias.

Además soporta varios formatos de imágenes por nombrar algunos: BMP, GIF, HDF, JPEG, PNG, FITS, entre otros [6].

Al igual que MATLAB, *Octave*[7] y *Scilab*[8] son programas utilizados para realizar cálculos numéricos, también poseen herramientas para el procesamiento de imágenes y poseen un lenguaje muy similar al de MATLAB, la gran diferencia que poseen es que son software libre a diferencia de MATLAB.

2.7.2. Lenguaje R

R es un software libre parecido a MATLAB pero además de tener casi las mismas funcionalidades que MATLAB permite muchas otras dado que es un lenguaje de programación más completo y no tan enfocado en los números. El enfoque principal que tiene R es el manejo de estadísticas y de manejo de imágenes para procesar, posee grandes cantidades de librerías con herramientas para abarcar filtros, métodos y otras funciones del procesamiento de imágenes.

Es de uso público y puede ser utilizado en casi todos los sistemas operativos, este sistema nació como una mejora del premiado lenguaje S que no poseía un entorno gráfico o ambiente tan amplio como R y que estaba enfocado netamente en estadísticas. Al ser un lenguaje de programación R entrega gran flexibilidad al usuario, permitiendo generar sus propios métodos y funcionalidades según sea necesario. Es un lenguaje orientado a objetos que además posee amplia conectividad con las distintas bases de datos que existen en la actualidad. R forma parte de un proyecto colaborativo que tiene gran cantidad de usuarios que día a día están sacando nuevas funcionalidades o librerías para hacer este software cada vez más completo.

Es una herramienta más completa que MATLAB por su gran versatilidad, además de su facilidad de uso de bases de datos. Librerías como EBImage o biOps poseen gran cantidad de herramientas y filtros que pueden ser aplicados a cualquier tipo de imágenes, así como también permiten adecuarlos para que funcionen de forma óptima con imágenes astronómicas. Posee una librería para el manejo de archivos en formato FITS.

En primera instancia resalta como una buena alternativa para realizar el procesamiento de las imágenes [9].

2.7.3. CASA

Descripción El *Common Astronomy Software Applications* es un software en proceso de desarrollo que tiene como meta soportar el post proceso de la data generada por los telescopios de radio de última generación como ALMA, así como también calibrar la data. Puede procesar interferometría y data de un solo plato. Actualmente está siendo desarrollado por un consorcio internacional constituido por la *National Radio Astronomical Observatory (NRAO)*, *el European Southern Observatory (ESO)*, *la National Astronomical Observatory of Japan (NAOJ)*, *el CSIRO Australia Telescope National Facility (CSIRO, ATNF)* y *el Netherlands Institute for Radio Astronomy (ASTRON)* bajo el mando de la NRAO.

CASA consiste en un set de herramientas realizadas en C++ unidas en una interfaz de iPython. Esta estructura posee flexibilidad de poder realizarse mediante scripts de python o mediante una interfaz de tareas. La última versión de CASA actualmente es la 4.0.1 [10].

Almacenamiento CASA guarda la data en forma de set de medidas (*Measurement Set (MS)*). De manera lógica guarda data generalizada desde un interferómetro o telescopio de un disco y de manera física son varias tablas en un directorio del disco. Las tablas en CASA son directorios que contienen subtablas. Las calibraciones o imágenes también son guardadas de la misma forma en directorios y sub directorios.

Cada fila de la tabla principal (*main table*) contiene una medida de tiempo, un espectro y un baseline. Cada tabla contiene gran cantidad de columnas. El directorio DATA contiene la data original, el directorio CORRECTED mantiene la data calibrada, el directorio MODEL contiene el modelo de Fourier utilizado e IMAGING_WEIGHT contiene lo relacionado con las imágenes. Cabe destacar que teniendo la data original y los modelos correspondientes para su calibración se puede generar el data calibrado. No así al revés, es decir no se puede obtener la data original con la data calibrada y el modelo.

En la *Figura 2.5* se ve el diagrama de flujo de CASA desde cargar los data sets hasta mostrar la imagen, pasando por el proceso de calibración y examinación de la información.

Funcionalidades

- **Importar:** CASA permite importar archivos previamente procesados, como también archivos de data originales provenientes de ALMA o VLA. Otros formatos que se pueden importar son UVFITS, FITS.
- **Examinar:** Permite observar data de diferentes formas, mediante un viewer,

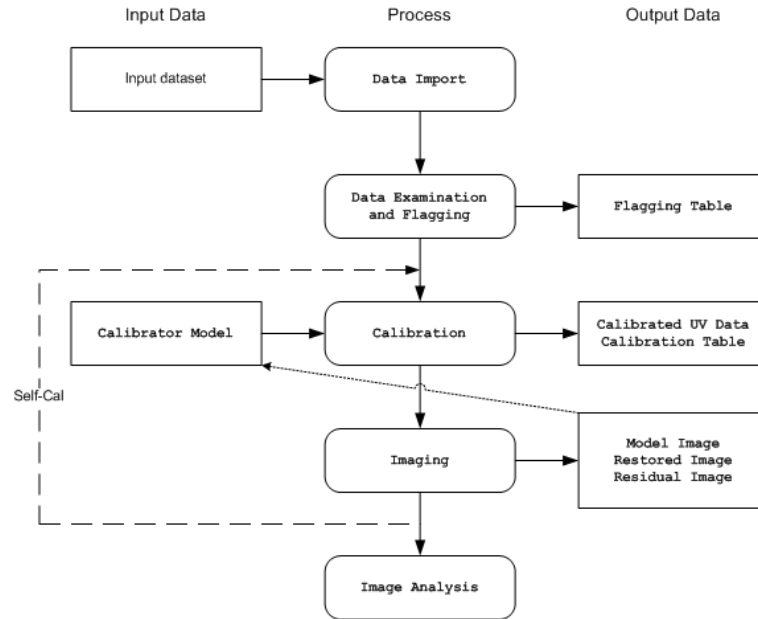


Figura 2.5: Diagrama de flujo de CASA.

mediante diferentes tipos de plots, permite observar un resumen del MS y permite marcar ciertas entradas o datos de interés en las tablas, entre otras.

- **Calibrar:** Durante la calibración el usuario especifica un set de calibración para pre aplicar al procesamiento de la data. El usuario especifica una tabla de calibración para guardar el resultado de ésta y para terminar la calibración es aplicada al data set y luego guardado en el directorio CORRECTED. Dentro de las calibraciones que se pueden realizar se encuentran las de polarización, de frecuencia, de dependencia de tiempo, entre otras.
- **Visualizar:** Tiene ciertas funcionalidades para visualizar los cambios realizados a la data o la data como tal. Existen algunos comandos que permiten la visualización de la data solo en el área visible o aplicando alguna transformación de Fourier por ejemplo.
- **Analizar:** Una de las funcionalidades esenciales de CASA, la manipulación y análisis de los datos. Permiten computar, calcular, substraer ciertas partes de la data para mejor análisis, generar diferentes grid entre otras.
- **Exportar:** CASA permite exportar la data en formato FITS y UVFITS.

2.8. Observatorios Astronómicos Virtuales

Los Observatorios Astronómicos Virtuales hacen el esfuerzo de tener toda la información astronómica del mundo mediante un acceso fácil, utilizando una interfaz web. Los observatorios permiten que los usuarios puedan obtener información astronómica que se encuentre en algún lugar del mundo con el fin de poder analizarla [11].

Los astrónomos construyen nuevos telescopios cada día, la cantidad de información que éstos colectan crece exponencialmente. Los astrónomos intentan estudiar detenidamente cada pieza de la información obtenida por los telescopios, para ayudar en esta tarea la información puede ser analizada por más personas a lo largo del mundo y a la vez permite relacionar la información con otros casos similares.

El desafío que tienen los observatorios virtuales es tremendo, dado que la cantidad de información que disponen crece cada día, otro desafío no menor es la conectividad que debe haber entre los distintos telescopios y los distintos observatorios virtuales, con el fin de que la información sea entregada de la manera más rápida posible y sin poseer errores que puedan entorpecer el análisis.

Capítulo 3. Propuesta Metodológica

En este capítulo se describe la metodología a utilizar para la generación del índice de objetos astronómicos. Para comenzar se hará un desglose de los pasos a seguir para la obtención del índice, para luego especificar con más detalles en que consiste cada uno de estos pasos y las posibles herramientas a utilizar para desarrollarlos.

3.1. Descripción de la metodología

La metodología propuesta en este proyecto pretende plantear el procedimiento necesario para la generación del índice de objetos astronómicos, así como también explicar todos los pasos intermedios para poder lograrlo. Dentro de esta metodología se encuentra la etapa de investigación acerca de las distintas herramientas que serán utilizadas y puestas a prueba en una etapa posterior de aplicación y validación de éstas.

Para realizar de mejor manera el proyecto, se decidió identificar las distintas áreas involucradas en éste y desarrollarlas de manera secuencial e independientes una de otra. Las distintas áreas son procesamiento de imágenes, base de datos, índice de objetos y página web.

- **Procesamiento de imágenes:** En el área del procesamiento de imágenes se analizarán las distintas herramientas para procesar las imágenes FITS, partiendo por leer la imagen, luego aplicarle ciertos filtros que permitan identificar de mejor manera donde están ubicados los objetos y que características se pueden extraer de ellos. Dentro de esta área también se contempla la investigación de los tópicos de astronomía necesarios para obtener la información necesaria de los headers de las imágenes FITS para luego poder archivarlos e indexarlos correctamente. Dado que las imágenes FITS es más que una simple foto y posee gran cantidad de metadata, la manipulación de esta data también será parte de esta área.

- **Base de Datos:** Dentro de esta área se analizarán las distintas bases de datos que se pueden utilizar con el fin de elegir la más adecuada que permita implementar un índice espacial que cumpla con lo requerido. De esta misma forma esta área está ligada fuertemente con la generación del índice, dado que las opciones de índice para las distintas base de datos es distinta.
- **Índice de objetos:** En el área de creación del índice se analizarán los distintos índices que permitan búsquedas espaciales para ver cuál cumple con las necesidades del proyecto. Principalmente enfocarse en que pueda y realice de manera eficiente búsquedas espaciales en torno a una cierta región que puede ser de distintas formas y no sólo una región rectangular.
- **Página Web:** Dentro del área de la página web se analizarán herramientas que faciliten la conexión con la base de datos para poder ejecutar las búsquedas que el usuario necesite. Principalmente se busca una herramienta simple que permita realizar las búsquedas y desplegar la información obtenida de la mejor manera posible.

3.2. Procesamiento de Imágenes

3.2.1. Imágenes FITS

Las imágenes FITS son el formato estándar utilizado en astronomía para las imágenes obtenidas de los observatorios, el formato fue aprobado por la NASA y por la Unión Astronómica Internacional (IAU) ¹.

A diferencia de los formatos convencionales de imágenes, el formato FITS permite archivar espectros, imágenes en dos dimensiones y cubos en tres dimensiones, además de permitir guardar metadata en el header del FITS con información valiosa para los astrónomos. El formato FITS al ser creado exclusivamente para imágenes astronómicas posee herramientas para guardar información de fotometría, como

¹Cooperación internacional de astrónomos alrededor del mundo.

también de la calibración espacial.

Su característica principal es que almacena metadata relevante para los astrónomos en el header de la imagen. Este header puede ser leído por personas directamente dado que está escrito en ASCII lo que hace muy fácil su lectura y permite a los astrónomos saber la procedencia del FITS si es que quisieran investigarla.

El formato FITS se estandarizó en 1981, desde entonces ha ido evolucionando y nuevas versiones han salido [12].

A pesar de que el formato FITS es estándar, la forma de guardar información en el header no lo es del todo. Algunos pares de valores pueden estar presente con distinto nombre solamente o en otro formato de coordenadas por ejemplo. Por consiguiente es necesario evaluar los header de los FITS generados por los observatorios para así poder acceder a la información correcta. En la Figura 3.1 se muestra un ejemplo de parte de un header de una imagen FITS obtenido mediante R. Se puede observar también que los pares de datos parecieron no estar relacionados y son desplegados uno al lado del otro, por ejemplo el valor de la clave "SIMPLE" es el valor "T", esto hace que la búsqueda del valor de una clave por ejemplo SIMPLE no sea tan directo, pero de igual forma mantiene un orden secuencial.

```
[1] "SIMPLE"      "T"          "BITPIX"
[4] "-32"        "NAXIS"      "2"
[7] "NAXIS1"      "1207"      "NAXIS2"
[10] "1289"       "EPOCH"     "0."
[13] "EQUINOX"     "0."        "OBSRA"
[16] "2.60000000000000E+02" "OBSDEC"    "-3.59166666666667E+01"
[19] "CRPIX1"      "8.26000000000000E+02" "CRVAL1"
[22] "2.60000000000000E+02" "CDELT1"    "-2.22222222222222E-03"
[25] "CTYPE1"      "RA---SIN"  "CRPIX2"
[28] "3.32250000000000E+02" "CRVAL2"    "-3.59166666666667E+01"
[31] "CDELT2"      "2.22222222222222E-03" "CROTA2"
[34] "0.00000000000000E+00" "CTYPE2"    "DEC--SIN"
[37] "BMAJ"        "6.66666666666667E-03" "BMIN"
[40] "6.66666666666667E-03" "BPA"       "0.00000000000000E+00"
[43] "ORIGIN"      "-----"   "OBJECT"
[46] "NGC6357_4    #09994" "BLANK"     "-99999999"
[49] "BLANKV"      "-99999999.5" "BLANKT"
[52] "1."          ""
[55] ""           "END"
```

Figura 3.1: Ejemplo de un header de una imagen FITS.

Lograr entender cada clave del header resulta una tarea tediosa, más aún cuando algunos valores solo son de interés de un grupo específico de astrónomos. Los valo-

res necesarios para este proyecto son los tamaños de la imagen, y las demás claves que nos permitan obtener la posición de la imagen FITS en el espacio para luego poder calcular las posiciones de los objetos que van a ser archivados.

Cabe destacar que las imágenes a ser procesadas serán de formato FITS obtenidas desde observatorios ópticos, en especial del “Sloan Digital Sky Survey”[13] que es un observatorio que realiza surveys o barridos del cielo. Este observatorio trabaja con 5 bandas de observación las cuáles podrán ser procesadas. Las bandas del Sloan son:

- u' : Banda de colores ultravioleta.
- g' : Banda de color verde.
- r' : Banda de color rojo
- i' : Banda de colores infrarojo.
- z' : Banda z que pertenece a colores cercanos y bajo el infrarojo.

Las imágenes FITS que sean de cualquiera de las bandas antes mencionadas podrán ser procesadas y luego archivar cada uno de los objetos presentes en ella. Una misma imagen apuntando al mismo lugar del cielo en distintas bandas puede poseer distintos objetos, es por esto que es importante evaluar cada imagen en todas sus bandas si es posible, de esta forma se abarcará una mayor cantidad de objetos.

3.2.2. Sistema de Coordenadas

Un sistema de coordenadas es un sistema de referencia que permite localizar un objeto o persona de forma inequívoca en una posición cualquiera dentro de un espacio dimensional. Normalmente cuando se habla de un sistema de coordenadas se hace con referencia a un sistema de coordenadas geográficas, es decir, que permiten localizar posiciones dentro de la tierra. Son de uso común dentro de los dispositivos móviles actuales que permiten saber la posición del dispositivo y generar rutas o

alguna clase de información ligada a esta posición [14].

Por otra parte cuando se quiere localizar un objeto astronómico en el espacio este tipo de coordenadas no es útil, para este tipo de objetos se utilizan sistemas de coordenadas astronómicas. Las coordenadas que se utilizarán serán *Right Ascension* y *Declination*.

Ascensión Recta (RA) y Declinación (DEC) Ascensión Recta (Right Ascension) y Declinación (Declination) son un sistema de coordenadas similar al geográfico pero para el espacio. Este sistema está basado en una esfera imaginaria que rodea la tierra y que posee radio infinito. Esta esfera está fija con respecto al universo, su orientación no cambia, sin embargo dado que la tierra gira de oeste a este un observador situado en la tierra verá la esfera celestial girar de este a oeste.

- **Declinación:** Al igual que las coordenadas geográficas dos coordenadas definen un punto en la esfera celestial. La rotación de la tierra define una dirección en el universo y es necesario definir esa rotación y dirección. Para esto se define el ecuador celestial que está en la misma posición del ecuador en la altura de la esfera. El norte y el sur celestial están de igual manera sobre el norte y el sur geográfico. La coordenada indicando la posición entre estos dos polos es llamada declinación.

La declinación es medida desde el ecuador celestial. Se extiende desde 0 en el ecuador hasta +90 grados en el norte celestial y desde 0 hasta -90 en el sur celestial. A pesar que en astronomía no se refieren a la declinación en términos de norte y sur, la latitud y la declinación están relacionadas.

- **Ascensión Recta:** La ascensión recta es la segunda coordenada de este sistema, es análoga a la longitud pero no es igual. Al igual que la longitud que posee el meridiano de Greenwich como referencia a su valor 0 la ascensión

recta también posee un punto de referencia. Este punto es el Equinoccio vernal (Vernal Equinox Point).

Dado que la tierra rota, desde la perspectiva de la tierra, la esfera celestial rota 1 vez cada 24 horas de tiempo sidereal ¹. Por lo tanto la ascensión recta es medida en horas de tiempo sidereal, 0h a 24h hacia el este del Equinoccio Vernal, es decir hacia el este de este punto la ascensión recta va incrementando. El medio círculo con ascensión recta 0h es llamado el círculo de 0 horas. Como un círculo tiene 360 grados, 1 hora en ascensión recta equivalen a 15 grados [15]. En la Figura 3.2 se muestra una esfera celestial, donde se pueden observar todos los conceptos expuestos anteriormente.

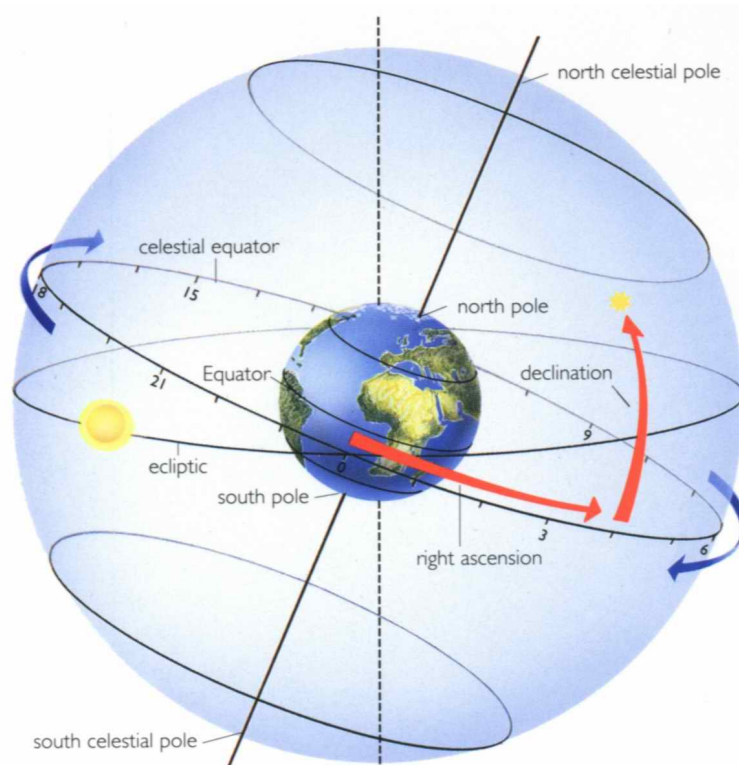


Figura 3.2: Esfera celestial con sistema de coordenadas RA, DEC.

Coordenadas Locales Las coordenadas locales no se expresan en términos de ascensión o declinación dado que su sistema de coordenadas está basado solo en la

¹ Tiempo medido por el movimiento diurno aparente del Equinoccio Vernal.

imagen, es decir en términos de x e y . Los valores de x y de y dependen del número de píxeles a la derecha o hacia arriba que estén desplazados respectivamente. Por ejemplo un punto ubicado en el píxel número 100 hacia la derecha y ubicado en el píxel número 200 de abajo hacia arriba tendrá unas coordenadas de x e y igual a 100 y 200 respectivamente.

Cabe destacar que estas coordenadas solo tienen valor dentro de la imagen y no pueden ubicarse en el espacio.

Coordenadas Absolutas Las coordenadas absolutas a diferencia de las locales que solo se basan en la imagen en estudio pertenecen al sistema de coordenadas basado en Ascensión Recta y Declinación, es decir, permiten transformar las coordenadas locales que no entregan mucha información en coordenadas celestiales para poder ubicar los objetos de una manera más estándar.

Para la obtención de las coordenadas locales, que son las coordenadas que entregan valor, dado que las locales sirven solo referenciando la imagen, es necesario el análisis del header de las imágenes a procesar, para esto es necesario un primer paso de identificar de donde provendrán las imágenes a analizar y verificar que toda la información necesaria para el cálculo de coordenadas esté presente en el header de la imagen.

Es vital que todas las imágenes a analizar estén en el sistema de coordenadas RA-DEC, en caso de venir en otro sistema de coordenadas será necesario un paso previo para transformar estas coordenadas a RA-DEC.

3.2.3. Pre Procesamiento

Una imagen en FITS se compone de dos secciones, una cabecera denominada header y una sección de datos. Mediante la librería FITSio presente en R es posible leer ambas partes de la imagen lo cual permite una manipulación interna de la sec-

ción de los datos. El primer paso antes del procesamiento es realizar un corrimiento del eje z el cual representa la intensidad de la imagen, con el fin de obtener el rango de observación adecuado donde los objetos se puedan dislumbrar de mejor manera. Este procedimiento se demuestra en la Figura 3.3 dando como resultado una imagen más nítida y con los objetos más resaltados.

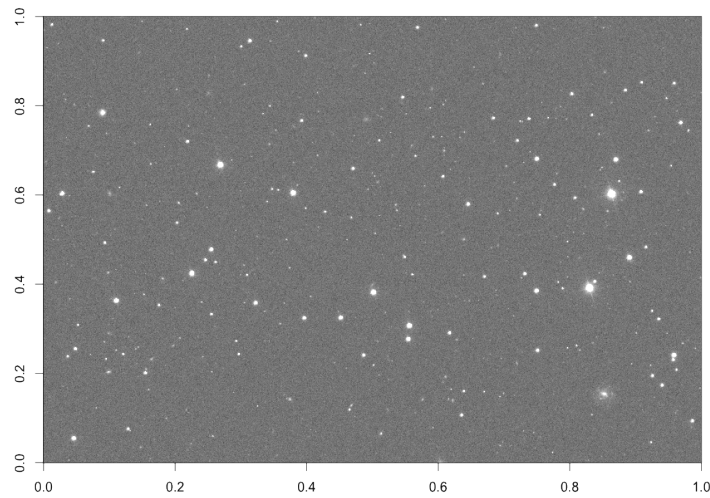


Figura 3.3: Imagen original, visualizada usando filtro en intensidad correspondiente al rango de observación.

En la Figura 3.3 se puede observar que la imagen de prueba posee gran cantidad de objetos de interés para poder ser procesados. Sin embargo, estos objetos son bastante pequeños, donde la mayoría de ellos podrían definirse como fuentes puntuales de luz. Dadas las dimensiones de la imagen, la proporción en información entre objetos y background es desigual dado que la mayoría de la imagen podría catalogarse como fondo de la imagen correspondiente al color negro, esto dificulta el proceso de caracterización de los objetos. Este problema se puede observar al ver el histograma que posee la imagen actualmente, en la Figura 3.4 se puede observar este comportamiento donde la mayoría de los píxeles de la imagen corresponden al color negro.

Es posible ajustar manualmente un filtro que permita enfatizar las características

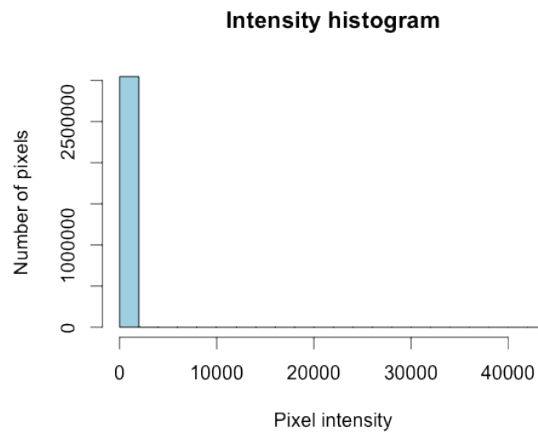


Figura 3.4: Histograma de intensidad.

discriminativas del histograma de intensidades. Sin embargo, el diseño manual de un filtro de intensidades no es recomendable ya que se requiere un procedimiento completamente automático.

Para esto es necesario la investigación y utilización de un algoritmo de threshold adaptativo el cual permite obtener un umbral más adecuado para la correcta binarización de la imagen y considerar de mejor manera que parte de la imagen realmente corresponde al fondo y que parte corresponde a los objetos propiamente tal [16] .

Los algoritmos de threshold son muy utilizados cuando se busca diferenciar el fondo de la imagen con los objetos presentes en ésta, la mayoría de los algoritmos de reconocimiento de patrones en ciertas imágenes requieren ajustar de buena manera el threshold de la imagen, dado que las condiciones ideales que pueda presentar una imagen no siempre son así, es decir, la variación que puede haber de una imagen a otra en cuanto a intensidad, luminosidad y otras variables obliga a utilizar una herramienta que normalice estas variables, lo que se logra con algoritmos de threshold. Cuando se habla de un threshold adaptativo se dice que el threshold no es fijo a lo largo de toda la imagen, sino que es dinámico de acuerdo a las condiciones que la imagen presenta. Esto es de mucha utilidad dado que algunas estrellas presentan cierto degradé en los bordes dado que la intensidad va disminuyendo paulatinamente y no de golpe.

Mediante el algoritmo de threshold se busca binarizar la imagen de manera óptima. La condición que debe satisfacer este algoritmo de búsqueda de threshold es evitar un cambio distribucional en la intensidad de la imagen, para lograr esto es necesario la utilización de un algoritmo basado en kernels.

Los algoritmos basados en kernels utilizan como base una matriz de convolución de diferentes tipos según sea el caso en estudio, para el caso de las imágenes astronómicas el kernel que se muestra más auspicioso es el de suavizado de Gauss.

El operador de Gauss realiza una convolución de dos dimensiones donde se busca remover detalles de la imagen y a la vez eliminar el ruido. El kernel de Gauss está basado en la función de la campana de Gauss, donde nuestros objetos en estudio formarían parte de esta campana y el fondo de la imagen no, es decir permite sobresaltar los objetos en estudio [17].

El algoritmo de thresholding adaptivo convoluciona un kernel de Gauss con la imagen original, con lo cual se obtiene la imagen original suavizada. Luego esta imagen resultante es comparada con la original donde se aplica una condición de binarización. La condición de binarización se calcula sobre el pixel resultante a partir de un pixel-wise AND sobre el vecindario del pixel, dicho de otra manera se estudian las vecindades del pixel para evaluar las intensidades y se define la intensidad resultante como la mayoría de intensidad en este vecindario.

El proceso de binarización de la imagen cumple un rol vital antes de procesar la imagen para registrar e indexar los objetos, dado que todas las observaciones independiente de su rango de observación es normalizada hacia un mismo dominio o rango de intensidades. Esto facilita su procesamiento posterior dado que todas las imágenes quedan similares y no poseen grandes cambios que puedan ensuciar las muestras.

Luego de binarizar la imagen al analizar su histograma se pueden apreciar diferencias frente al histograma antes de binarizar. En el histograma antes de binarizar presente en la Figura 3.4 solo se puede apreciar el fondo, mientras que después de binarizar la imagen se pueden apreciar algunos valores en torno al valor 1. Esto se puede apreciar en la Figura 3.5 luego de aplicar el threshold adaptativo.

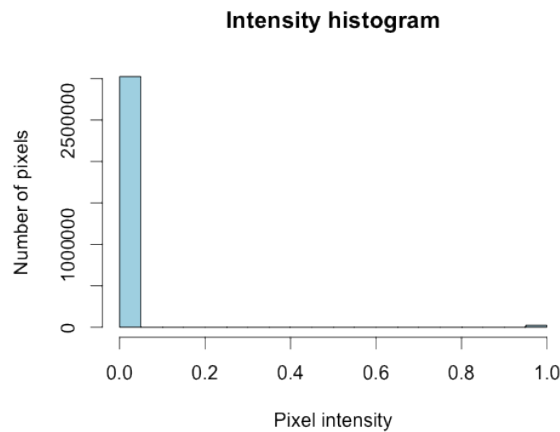


Figura 3.5: Histograma de intensidad para la imagen binarizada.

Cabe destacar que la imagen sigue teniendo un amplio fondo y es por eso que la concentración de intensidades en torno al valor 1 es mucho menor que los en torno al valor 0, el uso del kernel Gaussiano favorece la reducción de irregularidades en la imagen, donde por ejemplo los bordes con cortes abruptos son suavizados y además reduce la cantidad de artefactos introducidos en el proceso de adquisición de la imagen.

La imagen binarizada no representa la información de intensidad, sino que lo que se logra es que todas las fuentes presentes en la imagen se agrupen en un mismo bin o valor del histograma. Con esto las propiedades de los objetos se resaltan y es más fácil obtener características de geometría de los objetos, pero a su vez se debe destacar que esto genera un costo en términos de información dado que los objetos poseen una intensidad o brillo homogéneo. Para efectos prácticos de caracterizar objetos geoméricamente este proceso es de mucha ayuda, ahora si en un futuro se quisiera indexar por brillo o comportamiento de intensidades este proceso

no debería realizarse.

En la Figura 3.6 se puede observar la imagen de referencia obtenida luego del proceso de binarización.

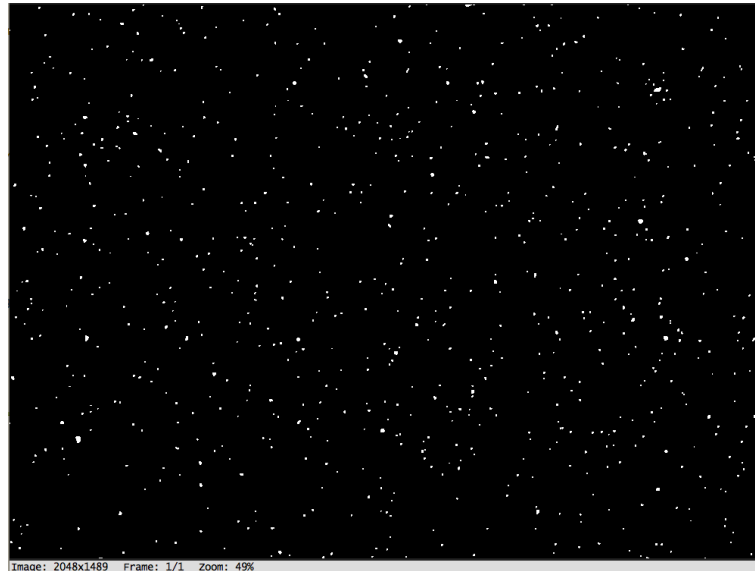


Figura 3.6: Imagen binarizada.

El proceso de binarización de la imagen se realiza mediante un algoritmo de *Opening*. El término de *Opening* se utiliza en el procesamiento de imágenes, es un procedimiento que elimina parte del ruido de fondo. El filtro de opening lo que hace es remover los ruidos presentes en el fondo de la imagen que puedan estarse confundiendo con los objetos, por lo general lo que remueve son pequeñas manchas que puedan estar entre medio de un objeto y otro por lo que en algunos casos los objetos se catalogan juntos, para evitar unir estos objetos se realiza este procedimiento de apertura de la imagen lo que limpia el fondo de la imagen dando como resultado la imagen binarizada previamente expuesta [18].

3.2.4. Extracción de características

Con el proceso anterior los objetos ahora están mejor definidos por lo que se procede a extraer características de cada uno de ellos. El primer algoritmo que será

utilizado es el *Labels connected components* [19], un algoritmo que recorre la imagen como si fuera una lista de adyacencia de un grafo. El algoritmo etiqueta las componentes conexas de la imagen, lo cual corresponde al proceso de asignar un ID a cada objeto de la imagen. El background se representa por el valor neutro. La Figura 3.7 ilustra el proceso de asignación de IDs a cada objeto de la imagen.

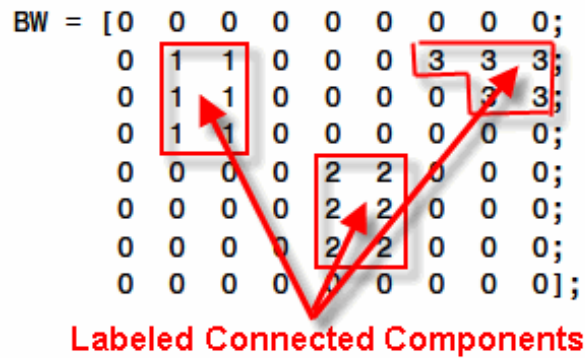


Figura 3.7: Algoritmo de labeled connected components.

La asignación de IDs se realiza de izquierda a derecha, comenzando de la esquina superior izquierda de la imagen, luego barriendo hacia la derecha y hacia abajo, similar a un proceso de lectura de texto. El algoritmo va juntando píxeles de acuerdo a un grafo de adyacencia y no se detiene hasta encontrar parte del fondo, es decir cuando se acaba el objeto, luego encuentra el siguiente objeto al cual le asigna el ID siguiente y así sucesivamente hasta recorrer toda la imagen.

Este proceso sirve para etiquetar cada uno de los objetos presentes en la imagen, por lo que después de este proceso ya se sabe el número de objetos presentes en la imagen y de los cuales se extraerán sus características geométricas.

Luego del etiquetado se comienza con la extracción de características, para esto se utiliza un cálculo de convex hulls sobre la imagen [20]. Los convex hulls son conocidos también como envoltorios conexos y lo que hacen es encerrar cada objeto en la forma geométrica más pequeña que contenga al objeto completamente. Con el paso anterior donde los IDs de cada objeto quedan totalmente definidos así como también todos los píxeles pertenecientes a cada objeto el método de aplicar convex

hulls se torna una tarea fácil.

Por lo general este tipo de métodos no son del todo sencillos de aplicar por los bordes, pero ya se solucionaron esos problemas previamente. La utilización de estos convex hulls es para facilitar y favorecer la estimación de algunos parámetros geométricos, como por ejemplo la excentricidad. La Figura 3.8 ilustra los resultados que se pueden obtener sobre una imagen al correr un algoritmo de convex hulls, como se puede apreciar en el ejemplo la imagen no es binarizada por lo que los rangos de intensidad de cada objeto son variables lo que dificulta el encontrar los convex hulls presentes en la imagen.

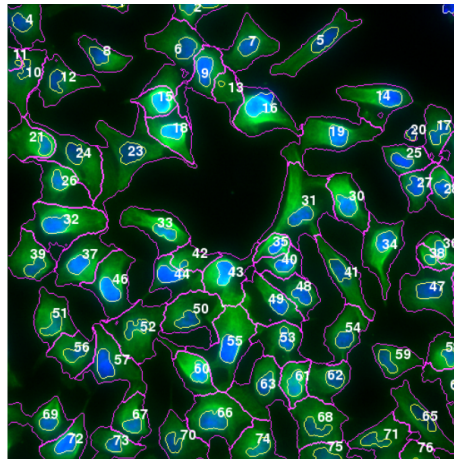


Figura 3.8: A cada componente se le calcula un convex hull para favorecer el cálculo de propiedades geométricas de los objetos.

Luego de obtener los convex hulls es posible realizar una extracción de características [21]. La extracción de características es un algoritmo que permite reducir la data de entrada para así generar un vector con características que se puede utilizar después para continuar procesando sin necesidad de utilizar la data de entrada sino que solo el vector. Es utilizada para optimización de algoritmos donde se repite una forma o patrón muchas veces, para el caso particular se optimiza y se obtienen características de los objetos presentes en la imagen, mediante esta extracción se obtienen los datos necesarios para ser guardados en la base de datos y las coordenadas para ser indexadas, por lo que no se realiza un futuro procesamiento luego de

la extracción.

Cuando se realizan análisis de grandes cantidades de información es necesario realizar esta clase de optimizaciones y extracciones, son utilizadas generalmente para luego realizar clasificación de objetos, por lo que si se quisiera llegar más allá en un futuro el vector de características sería de mucha utilidad y describe de manera fidedigna la data de entrada.

En la Figura 3.9 se muestran algunas de las posibles características que se pueden extraer de cada uno de los objetos presentes en la imagen mediante un *computeFeatures* presente en la librería EImage de R.

```
> ft
```

	x.0.m.cx	x.0.m.cy	x.0.m.majoraxis	x.0.m.eccentricity	x.0.m.theta	x.0.s.area	x.0.s.perimeter	x.0.s.radius.mean
1	360.705882	2.294118	5.552178	0.70864238	-4.586406e-01	17	12	1.894053
2	1123.857143	1.928571	5.449354	0.81653996	-1.249893e-01	14	11	1.658186
3	1236.465347	3.287129	20.226869	0.93781905	-8.783935e-02	101	44	5.965818
4	1251.352941	1.941176	6.585671	0.87572674	-7.518571e-02	17	13	1.895080
5	1385.850000	1.950000	7.731011	0.91039281	-5.095918e-02	20	15	2.131025
6	1461.857143	1.928571	5.449354	0.81653996	-1.249893e-01	14	11	1.658186
7	1666.363636	1.909091	4.346130	0.70844375	-2.630332e-01	11	9	1.416837
8	1078.857143	5.857143	6.110101	0.71224197	-7.853982e-01	21	13	2.182547
9	519.275000	8.075000	8.817736	0.73150164	-1.446208e+00	40	21	3.122624
10	81.266667	9.366667	6.910370	0.60007238	-7.091128e-01	30	16	2.646768
11	1697.421053	10.894737	5.544698	0.63228090	-1.347036e+00	19	13	1.986977
12	715.600000	13.250000	8.649424	0.72740874	1.192032e+00	40	20	3.147061
13	334.250000	15.325000	8.862435	0.75671957	-4.677388e-01	40	20	3.174386
14	1737.894737	23.421053	5.544698	0.63228090	-2.237600e-01	19	13	1.986977
15	215.434783	25.391304	6.667282	0.75754452	-1.463262e+00	23	15	2.241941
16	728.343750	25.625000	6.488646	0.25997945	-1.211123e+00	32	17	2.723756
17	172.903226	26.838710	7.695354	0.75017128	-1.246848e+00	31	17	2.720127
18	1277.894737	25.421053	5.544698	0.63228090	-2.237600e-01	19	13	1.986977
19	1809.171429	28.400000	8.573774	0.79735111	-1.508030e+00	35	19	2.914997
20	412.400000	31.400000	4.618802	0.48989795	-7.853982e-01	15	11	1.716806

Figura 3.9: Características geométricas de los objetos.

3.3. Base de Datos

La elección de base de datos tiene un papel preponderante en el proyecto, dado que toda la información obtenida al procesar las imágenes debe ser almacenada en una base de datos para su posterior uso. De esta misma forma esta base de datos debe ser capaz de implementar índices espaciales los cuales son requeridos para realizar las consultas de una manera más rápida y eficiente.

Por lo general la mayoría de las base de datos que se usan actualmente permi-

ten implementar un índice espacial, pero claramente la forma de implementación y funcionamiento son muy distintas.

La información a guardar en la base de datos en primera instancia es la obtenida mediante el procesamiento de las imágenes, es decir las coordenadas astronómicas de cada objeto, así como también el nombre del archivo FITS desde donde fueron obtenidos como también alguna otra característica si fuera necesaria.

El modelo de base de datos a utilizar será una base de datos relacional. En la actualidad es uno de los modelos más utilizados en base de datos dado que es utilizado para modelar problemas reales y administrar datos de forma dinámica. Este modelo se basa en el uso de relaciones, estas relaciones se conocen también como conjuntos de datos denominados *tuplas*. Cada relación se considera una tabla donde las filas representan los registros de la tabla y las columnas son los campos de la tabla. A diferencia de otros modelos jerárquicos de base de datos la forma y el lugar donde se almacenan los datos no tiene mayor importancia, dado que la información es recuperada mediante consultas que ofrecen gran flexibilidad para administrar la información.

El lenguaje más habitual de consultas es el SQL¹ que a su vez es muy utilizado en la astronomía. En astronomía también utilizan bases de datos relacionales por el hecho de mantener grandes cantidades de tablas y relaciones, donde el correcto uso del SQL es vital para lograr consultas eficientes y no sobrecargar más de lo necesario las bases de datos.

Por lo general en astronomía los grandes archivos de datos permiten el uso de SQL, claro que las consultas están limitadas y poseen ciertas reglas para asegurar su buen funcionamiento, un ejemplo son la cantidad de filas que se pueden extraer. Un ejemplo de un sitio que permite realizar consultas en SQL para obtener información

¹ Structured Query Language.

del observatorio *Sloan Digital Sky Survey* es Casjobs [22].

Casjobs permite a cualquier persona natural poder crearse una cuenta e implementar algunas consultas. En un principio es un tanto complejo dado que las tablas poseen gran cantidad de información y cuesta saber realmente en que parte está la información que uno necesita. Para comenzar se pueden realizar consultas simples y luego ir investigando y puliendo de mejor forma las consultas.

3.3.1. Elección de Base de Datos

SQLite SQLite es un motor de base de datos que soporta consultas SQL. Es de uso público para cualquier clase de proyectos ya sean públicos o privados. A diferencia de otras base de datos que necesitan de un servidor, SQLite permite leer y escribir directamente en un archivo del disco duro, lo que facilita su instalación y utilización para casos donde se necesita algo rápido y sencillo, mediante solo un archivo en el disco duro se pueden almacenar, las tablas, índices, triggers, etc. SQLite funciona tanto en 32 bits como en 64 bits, con solo copiar el archivo de una plataforma a otra sin tener que hacer algún cambio. SQLite es una buena solución cuando se busca software abierto y algo simple, posee una comunidad bien amplia de desarrolladores que trabajan en nuevas funciones y en solucionar los bugs [23].

SQLite posee el índice espacial R-Tree. La forma en como funciona es primero teniendo una tabla normal que posea datos de los objetos además de poseer un cuadro que pertenece a una región, es decir posee 2 coordenadas máximas y 2 coordenadas mínimas que generan una región en el espacio. Luego sobre esa tabla es necesario generar un índice que puede ser de 2 e incluso 3 dimensiones. Para realizarlo es necesario generar una tabla virtual que contendrá solo los datos de ubicación, es decir los valores a agregar por cada objeto al índice son el id (el cuál sirve para ligar los demás datos de características pertenecientes en otra tabla) y 4 valores que corresponden a las coordenadas mínimas y máximas. Para utilizar las capacida-

des del índice es necesario realizar el query mediante la tabla virtual y luego hacer un join sobre la otra tabla para recuperar los detalles del objeto.

Para activar el módulo de R-tree es necesario agregar el comando *DSQLITE_ENABLE_RTREE=1* y generar una tabla virtual de la forma *CREATE VIRTUAL TABLE <name>USING rtree(<column-names>)* [24].

Un punto en contra que posee SQLite es que para el índice R-Tree necesita que se archiven regiones y no puntos, cosa que puede ser un problema para archivar los objetos dado que se tienen las coordenadas del centroide de cada objeto (es decir un punto, no una región). Otra cosa que llama la atención es que las consultas no varían mucho con respecto a una consulta por rangos de un valor y rangos de otro valor que se puede hacer normalmente sobre una tabla, ahora esto no quiere decir que el índice no funcione sino que traduce esa consulta para poder utilizar de forma correcta el índice.

MySQL MySQL es uno de los motores de base de datos más usados actualmente, a pesar de que es un software de tipo open source, de igual forma es necesario pagar muchas de las funcionalidades que éste motor posee, sobre todo las más complejas. Las funcionalidades básicas de MySQL son gratuitas pero cuando uno quiere profundizar un poco u ocupar ciertas herramientas deja de ser gratis. Llama la atención que incluso para leer algunos de sus tutoriales o guías pida autenticación de usuario con cuenta de Oracle.

A diferencia de SQLite, MySQL es una organización mucho más grande y con gente más dedicada al tema. Se han ganado un prestigio por las grandes empresas que utilizan sus servicios como lo es Facebook. Las características que dicen tener por sobre las otras bases de datos son [25] :

- Escalabilidad y Flexibilidad: Utilizando herramientas de última generación hace que sus bases de datos funcionen de manera óptima sin importar el nú-

mero de usuarios. Poseen grandes clientes que lo respaldan.

- **Alto Rendimiento:** Al igual que en la escalabilidad, poseen un procesamiento de alto rendimiento logrando grandes optimizaciones y bajos tiempos de respuesta.
- **Alta disponibilidad:** Poseen bajas tasas de servicios caídos o no funcionales. Son muy estables.
- **Protección de datos:** Ofrecen herramientas para la protección de los datos de todos sus usuarios, con el fin de dar confianza y seguridad.
- **Soporte 24x7:** Poseen sistema de soporte todos los días a toda hora.

Al investigar acerca de los índices espaciales que ofrece MySQL se deja ver de inmediato que posee mayor cantidad de herramientas para generar consultas espaciales que SQLite. Las consultas a diferencia de SQLite pueden ser por regiones geométricas, que pueden ser cuadrados por ejemplo o polígonos más complicados. A diferencia de SQLite donde los campos utilizados para las búsquedas espaciales son números aquí pueden corresponder a guardar figuras geométricas que no necesariamente son cuadrados (como en SQLite) y además permiten guardar puntos, que es lo que se busca para el índice de objetos, dado que se indexarán mediante su centroide.

Al poseer mayor cantidad de herramientas y tipos de datos más complejos de entender y utilizar el uso e implementación del índice es más compleja, pero a su vez permite hacer búsquedas óptimas y elaboradas.

Una de sus grandes debilidades es el hecho de que la mayoría de sus herramientas son pagadas, por lo que la instalación y utilización de estas bases de datos es más tedioso.

PostgreSQL PostgreSQL es otro motor de base de datos basado en el modelo relacional. Es público para el que quiera usarlo y provee buenas herramientas para

instalar e iniciar sus servicios. La instalación de PostgreSQL viene con un idea llamada pgAdmin que permite crear bases de datos, realizar consultas sobre las tablas presentes entre otras cosas. Es una interfaz sencilla y viene configurado con lo básico para llegar y usar [26].

PostgreSQL debe ser un punto intermedio entre SQLite y MySQL donde SQLite es muy simple por lo que algunas funcionalidades no las tiene y MySQL siendo más complejo y completo. Posee una gran cantidad de herramientas para interactuar con las bases de datos que pueden ser utilizadas en casi todos los sistemas operativos.

Al parecer PostgreSQL es una buena decisión de base de datos que no requiere mayores capacitaciones ni gastos en herramientas extras, si posee herramientas de pago pero también posee muchas herramientas open source.

En términos de búsquedas espaciales la configuración de PostgreSQL se vuelve un poco más tediosa, dado que hay que instalar un paquete extra denominado PostGIS que a su vez requiere de otras librerías presentes en el servidor [27]. Dentro de esta librería están los índices espaciales y los objetos geométricos parecidos a los que utiliza MySQL, donde las consultas por región son del estilo este punto está dentro de esta región que puede ser un cuadrado o polígonos más complejos acorde a las necesidades.

El paquete de PostGIS está separado de la configuración básica de PostgreSQL dado que posee una gran cantidad de herramientas que no son tan básicas para utilizar, que a su vez permiten gran flexibilidad para realizar y responder consultas de manera rápida y eficiente. PostGIS dice ser el complemento más completo en búsquedas espaciales y tiene recursos dedicados al buen funcionamiento del paquete lo que lo hace una herramienta muy poderosa con nuevas actualizaciones de vez en cuando.

En la Tabla 3.1 se muestra una comparación de los tres posibles motores de bases de datos a utilizar, tomando en cuenta los puntos expuestos anteriormente. Cabe destacar que en funcionalidades, se analizan las que son necesarias para llevar a cabo el índice de objetos astronómicos, así como también en complementos pagados.

Competencia	SQLite	PostgreSQL	MySQL
Instalación	Sencilla	Sencilla	Normal
Configuración avanzada	Limitada	Extensa	Extensa
Índice espacial	Si soporta	Si soporta	Si soporta
Tipo de índice	R	GiST	R
Funcionalidades	No las cumple	Si las cumple	Si las cumple
Complementos extra	No necesita	PostGIS	Librerías de geometría
Escalabilidad	Limitada	Si	Si
Flexibilidad	Limitada	Gran flexibilidad	Flexible
Complementos pagados	No	No	Si

Tabla 3.1: Comparación de los puntos importantes de las bases de datos.

Al momento de elegir entre una y otra base de datos es necesario saber las características que esta va a cumplir, el modelo de datos que ocupará la cantidad de tablas y relaciones que pueda tener para lograr un funcionamiento óptimo. Para el proyecto la decisión esta entre SQLite y PostgreSQL dado que MySQL requiere ciertas licencias para utilizar las búsquedas espaciales. Por otra parte SQLite al parecer es demasiado básico en búsquedas espaciales por lo que no se podría sacar demasiado provecho de éstas, la elección más aceptada parece ser PostgreSQL que no requiere licencias para la utilización de PostGIS y permite realizar búsquedas más complejas.

3.3.2. Base de Datos Espacial

Las bases de datos espaciales pueden estar basadas solo en datos de ubicación espacial o como en el caso de este proyecto poseer algunos campos especiales en una tabla que permitan fijar la ubicación del objeto en un espacio previamente definido. Para el caso de astronomía y como se mencionó anteriormente se usan las coordenadas de tipo RA-DEC que tienen como unidad los grados, se pueden asimilar mucho a las coordenadas geográficas que utiliza cualquier sistema de posicionamiento en la tierra por lo que no es necesario procesarlas de alguna forma para ubicarlas en el espacio.

En este estilo de bases de datos es necesario definir un cuadro de referencia que permite definir la localización y relación entre los objetos, ya que los datos tratados en este tipo de bases de datos tienen un valor relativo al cuadro de referencia. Los sistemas de referencia espacial pueden ser georreferenciados como no georreferenciados, los georreferenciados se establecen sobre la superficie terrestre.

La construcción de una base de datos geográfica implica un proceso de abstracción para pasar de la complejidad del mundo real o en este caso el espacio a una representación simplificada que pueda ser procesada y entendida por las computadoras actuales.

Este proceso de abstracción tiene diversos niveles y normalmente comienza con la concepción de la estructura de la base de datos, generalmente en capas; en esta fase, y dependiendo de la utilidad que se vaya a dar a la información a compilar, se seleccionan las capas temáticas a incluir.

La estructuración de la información espacial procedente del mundo real en capas conlleva cierto nivel de dificultad. En primer lugar, la necesidad de abstracción que requieren los computadores implica trabajar con primitivas básicas de dibujo, de tal forma que toda la complejidad de la realidad ha de ser reducida a puntos,

líneas o polígonos.

En segundo lugar, existen relaciones espaciales entre los objetos geográficos que el sistema no puede obviar; la topología, que en realidad es el método matemático-lógico usado para definir las relaciones espaciales entre los objetos geográficos puede llegar a ser muy compleja, ya que son muchos los elementos que interactúan sobre cada aspecto de la realidad.

3.3.3. Datos Espaciales

Un modelo de datos geográfico es una abstracción del mundo real que emplea un conjunto de objetos dato, para soportar el despliegue de mapas, consultas, edición y análisis. Los datos geográficos, presentan la información en representaciones subjetivas a través de mapas y símbolos, que representan la geografía como formas geométricas, redes, superficies, ubicaciones e imágenes, a los cuales se les asignan sus respectivos atributos que los definen y describen [28].

Un dato espacial es una ubicación en el espacio que posee una forma particular, estos datos por lo general son de forma vectorial los cuales se construyen y pueden ser expresados por tres tipos de objetos espaciales.

- **Puntos** Definidos por un par de coordenadas, que en este caso son RA-DEC que denotan un hito en el espacio.
- **Líneas** Objetos abiertos que cubren una distancia dada, se usan generalmente para calcular rutas entre un punto y otro, mediante varias líneas es posible trazar un polígono también por lo que aunque lo de rutas no es necesario para el índice, trazar líneas quizás si.
- **Polígonos** Figuras cerradas que son la unión de varios objetos abiertas por ejemplo líneas, cubren un área determinada.

De esta forma la información contenida en puntos, línea o polígonos se almacena como un conjunto de pares coordenados. La ubicación puntual de un objeto se define como una coordenada (RA,DEC), las características lineales se almacenan

como un conjunto de coordenadas y los polígonos también se almacenan como un conjunto de coordenadas solo que son cerrados.

3.4. Índice de Objetos

3.4.1. Acceso espacial

Para evitar recorrer los datos de forma secuencial lo que genera una carga innecesaria, se crean índices que reducen la cantidad de elementos a visitar para responder una consulta. En este caso particular el índice tiene que ser de tipo espacial es decir tiene que estar basado en algún dato espacial presente en la tabla a indexar.

La indexación clásica de datos mediante B-Tree no sirve para índices espaciales dado que los valores no tienen un orden único. Por este motivo es necesario la utilización de algún método de acceso espacial. Existen tres categorías de métodos de acceso espacial, las PAM (Point Access Method), los árboles R-Tree y las SAM (Spatial Access Method), el punto de acceso a utilizar depende del tipo de dato espacial por el cual se está consultando.

Aunque hay un montón de benchmarks frente a los diferentes tipos de acceso espacial, ninguno es muy concluyente pero lo que si se recomienda al menos utilizar uno, dado que al realizar consultas sin índice espacial tomará mucho más tiempo del necesario.

B-Tree Los árboles B son estructuras de datos que se encuentran comúnmente en las implementaciones de bases de datos y de sistemas de archivos. Son árboles balanceados donde cada nodo puede poseer más de dos hijos. Con esto las inserciones y eliminaciones de datos se realiza de manera más óptima.

La búsqueda se realiza de forma similar a los árboles binarios donde se comienza buscando por la raíz y se va recorriendo el árbol hacia abajo dependiendo del valor del subnodo. El árbol B se utiliza para indexar un valor de datos de la tabla,

por lo que para las búsquedas espaciales no sirve [29].

kd-Tree Un árbol kd se refiere a un árbol de k dimensiones, es una estructura de datos que organiza los puntos en un espacio Euclídeo. Un árbol kd emplea solo planos perpendiculares a uno de los ejes del sistema de coordenadas. Todos los nodos del árbol almacenan un punto, desde el nodo de la raíz hasta las hojas.

La construcción del árbol se hace de forma de que todos los ejes de coordenadas vayan iterando, por ejemplo en un árbol de dos dimensiones el nodo de la raíz representa al eje x , los nodos hijo de la raíz representan al eje y , los nietos de la raíz vuelven a representar el eje x y así sucesivamente. En caso de tener más dimensiones funciona de la misma forma.

En la Figura 3.10 se muestra un ejemplo de kd tree en dos dimensiones. Se puede observar que todos los nodos guardan información sobre un punto y no es necesario llegar a la raíz para encontrar todos los puntos, esto puede acortar la consulta sobre el árbol pero también puede extenderla más de la cuenta si el árbol queda demasiado desbalanceado. También se puede observar que el kd-tree abarca toda el área a pesar de que hay partes donde no hay puntos u objetos de interés. Se puede observar también que los puntos que están sobre una línea vertical corresponden a una partición en el eje x y eso se puede ver reflejado gráficamente en la imagen y también en el árbol.

R-Tree Los árboles R son estructuras de datos similares a los árboles B, pero a diferencia de los árboles B que indexan una sola variable, los árboles R permiten almacenar variables multidimensionales por lo que sirven para las búsquedas espaciales. La estructura de datos divide el espacio en forma jerárquica en conjuntos que podrían estar sobrepuestos.

Cada nodo interno del árbol R almacena dos datos, uno es la identificación del nodo hijo y además guarda los límites de coordenadas que puede tener el nodo,

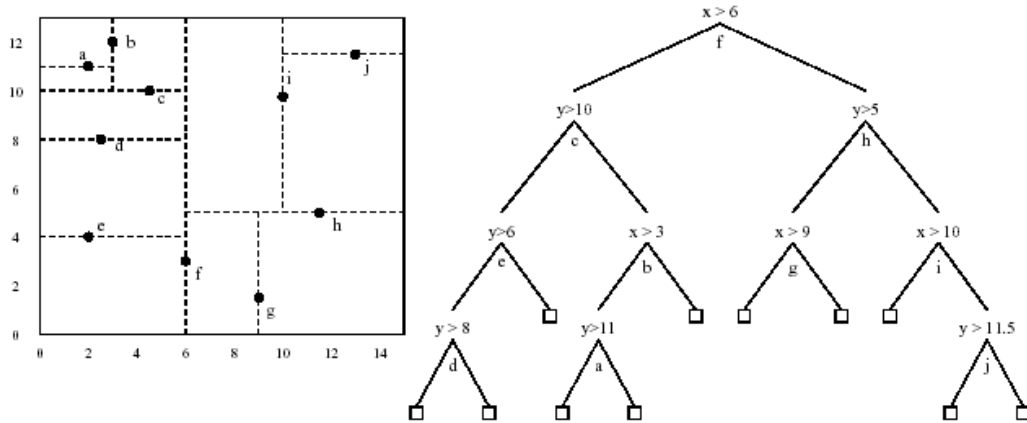


Figura 3.10: Ejemplo de kd-Tree en dos dimensiones.

con el fin de que todos los hijos estén dentro de esos límites para ser guardados en ese nodo. La cantidad de hijos que puede contener cada nodo se define al generar el árbol. Los algoritmos para insertar y borrar datos de los árboles se fijan en los conjuntos límites de los nodos para asegurar que todos los elementos cercanos pertenecan a la misma hoja. De esta misma forma al momento de insertar un nuevo punto o región al árbol se verifica cuál nodo sufriría el menor aumento en su conjunto límite para poder insertarlo. De forma similar a la inserción, los métodos de búsqueda sobre árboles R se fijan en los conjuntos límites de los nodos, por lo que requieren recorrer una menor parte para encontrar la región buscada, por lo que facilitan mucho el uso de base de datos con búsquedas espaciales.

Los árboles R no garantizan un buen rendimiento en los peores casos, pero si logran facilitar las búsquedas comúnmente hechas como encontrar todas las farmacias en una región definida por cuatro pares de coordenadas [30].

En la Figura 3.11 se muestra un ejemplo de un R-tree donde la parte superior muestra las regiones que corresponden a los conjuntos límites de cada nodo y en la parte inferior se muestra como un árbol. Se puede observar de manera gráfica que la región número 3 posee dentro las regiones 8,9 y 10, y en la parte inferior se puede observar que corresponde que estas 3 regiones son nodos hijo de la región número 3.

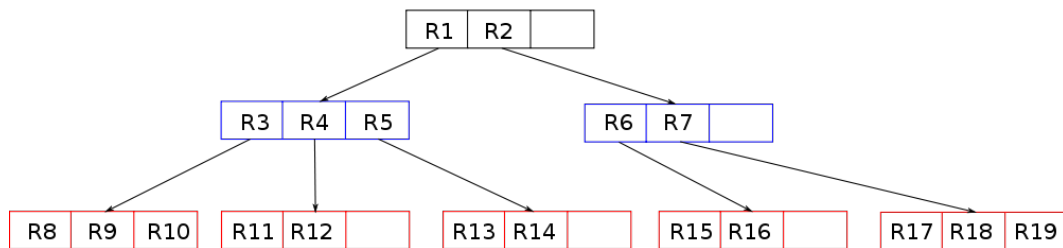
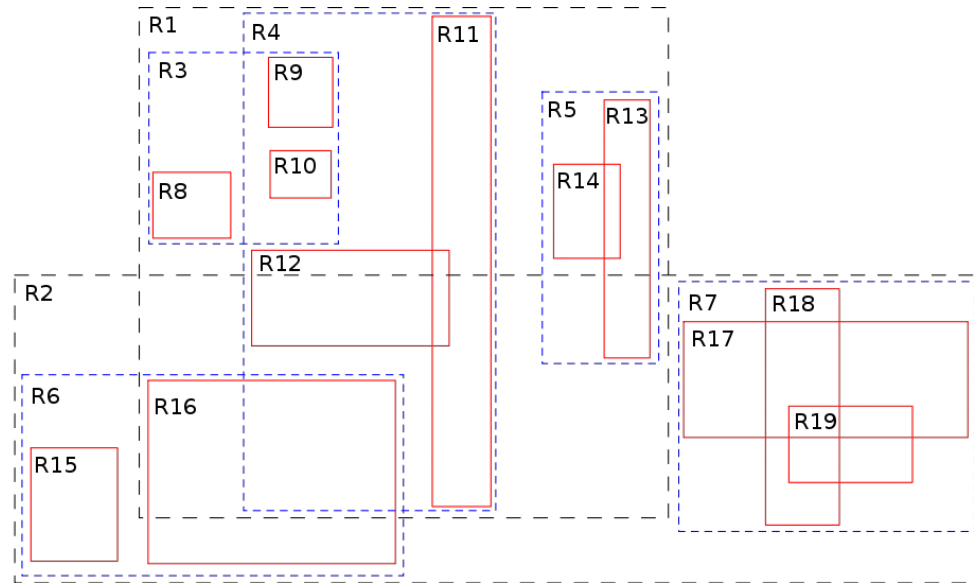


Figura 3.11: Ejemplo de R-Tree en dos dimensiones.

Como se puede observar en el ejemplo esta clase de índice se usa cotidianamente cuando por ejemplo se busca algún restaurante cercano a la posición actual de una persona con alguna aplicación que tenga geolocalización. De forma similar permite establecer regiones de interés en el espacio, aunque para la indexación de los objetos astronómicos se archivará solo un punto en el índice, el cuál corresponde a la posición del centroide del objeto.

Diferencias entre kd-tree y R-tree

- Los árboles R son balanceados, lo que hace que todas las consultas se demoren el mismo tiempo, por otra parte los árboles kd no lo son lo que requiere aplicar técnicas extras de optimización para utilizarlos.
- Los árboles R están orientados para su uso en disco, por lo que organizan la información de tal forma que al querer guardarlas en una base de datos por

ejemplo se torna fácil. Por otra parte los árboles kd están orientados a trabajar en memoria, por lo que escribir en disco puede ser tedioso.

- Los árboles R no cubren todo el espacio si es que no hay objetos presentes en alguna área remota de este, por otra parte los árboles kd siempre cubren todo el espacio.
- Los árboles R particionan los datos en rectángulos, mientras que los árboles kd hacen una partición binaria. Por esto mismo los rectángulos del árbol R pueden sobreponerse, lo que no es necesariamente un problema, pero se trata de minimizar.
- Los árboles kd son fáciles de implementar en memoria, el cual es su enfoque. A diferencia de los árboles R.
- Los árboles R pueden guardar rectángulos y polígonos, mientras que los árboles kd solo guardan puntos dado que no soportan la sobreposición.
- Los árboles R vienen con varias estrategias de optimización como diferentes estrategias a la hora de insertar, reinsertar o borrar algún punto.

Es por estos motivos recién descritos que la elección entre árbol R y árbol kd se inclina hacia el árbol R. Ambos podrían funcionar perfectamente pero el que se desempeña de manera óptima en las circunstancias del proyecto es el árbol R, esto por que requiere escribir continuamente en base de datos, requiere optimizar las consultas y el hecho de ser desbalanceado puede perjudicar mucho la implementación del índice, requiere poder guardar formas más complejas si en un futuro los objetos se caracterizan de forma más completa, entre otras.

GiST GiST viene de la sigla Generalized Search Tree y es una estructura de datos que permite crear diferentes tipos de árboles. GiST es una generalización del árbol B que provee una herramienta de búsqueda mediante un árbol balanceado, sin realizar asunciones del tipo de dato a guardar o del tipo de consulta que se está realizando. GiST puede ser utilizado para implementar los índices más comúnmente

usados como lo son el árbol B, el árbol R, el árbol kd, entre otros. De la misma forma que permite la utilización de distintos tipos de índices y tipos de datos por el cual indexar, también permite realizar índices especiales sobre nuevos tipos de datos, tiene como restricción que los árboles a implementar deben ser balanceados en cuanto a la altura, por lo que no sirve para generar índices como quad tree ¹. GiST puede ser usado con cualquier tipo de data siempre y cuando pueda ser ordenada de forma jerárquica [31].

GiST es una estructura de datos extensible en términos de tipos de dato y tipos de árboles, también permite un manejo sobre las consultas a realizar donde estas también pueden realizarse especialmente para un caso particular. Como se muestra es una herramienta que extiende de manera óptima el manejo de base de datos, permite la correcta evolución de las bases de datos para soportar nuevos tipos de árboles.

El mayor uso de GiST en la actualidad está en las bases de datos relacionales PostgreSQL, esta implementación incluye soporte para largos variables de los datos, datos compuestos, concurrencia y recuperación, éstas están presente en todas las extensiones de GiST. Además de esto hay una gran cantidad de módulos siendo desarrollados con GiST y distribuidos mediante PostgreSQL.

Con el hecho de permitir el uso de GiST, PostgreSQL suma otro punto más a la hora de elegir una base de datos para indexar o guardar datos espaciales, su implementación simple no es complicada pero si permite un nivel de personalización que puede transformarlo en una implementación más compleja pero que se adapta mejor a las necesidades.

¹ Árbol basado en la descomposición recursiva.

3.4.2. Consultas espaciales

Al trabajar con índices espaciales el tipo de consultas a realizar puede ser distinto, en los ejemplos más básico de un árbol R en una base de datos SQLite el tipo de consultas no varía mucho con respecto a una sin índice, es decir se realiza una consulta que incluye un WHERE con las coordenadas mayores y menores de RA-DEC para así abarcar una región cuadrada, la única diferencia es que la consulta se realiza sobre el índice que es una tabla virtual en vez de la tabla de datos directamente.

En MySQL y PostgreSQL las consultas no son tan sencillas y permiten mayor complejidad. Ambas bases de datos realizan consultas sobre una forma poligonal previamente definida y buscan objetos dentro de éste, por lo que en vez de utilizar un WHERE definen los bordes del polígono con las coordenadas mayores y menores de RA-DEC.

Los tipos de consultas espaciales pueden realizarse de muchas formas dependiendo de que tan complejo es lo que se busca, para efecto del índice de objetos astronómicos se quiere que el índice sea capaz de realizar las tres siguientes consultas:

- **Búsqueda por región:** La búsqueda por región queda definida por cuatro valores, dos valores inferior y superior de coordenada RA y dos valores inferior y superior de coordenada DEC. Con esto se forma una región cuadrada donde se buscarán todos los objetos presentes en esta región. Tiene como obligación los cuatro datos de entrada.
- **Búsqueda circular:** La búsqueda circular queda definida mediante un par de coordenadas RA-DEC y un radio medido en grados al igual que las coordenadas. Con esto se forma una región circular donde se buscan todos los objetos dentro de ésta. Cabe destacar que para este tipo de búsqueda no existe un objeto círculo listo para consultar, sino que es necesario generar una región mediante un cálculo de distancias hacia el centro, esto quiere decir que puede

ser más costoso en términos de ejecución que la búsqueda por región.

- **Búsqueda de vecinos cercanos:** La búsqueda de los vecinos cercanos lo que busca es darle como datos de entrada un par de coordenadas que permitan generar un punto en el espacio y sobre ese punto trazar un círculo de un largo definido con un radio no muy largo con el fin de encontrar los objetos pertenecientes a el que se podrían catalogar como vecinos por cumplir una distancia acotada. La búsqueda es parecida a la búsqueda circular solo que el radio queda definido previamente.

Las tres consultas expuestas anteriormente pueden ser realizadas mediante MySQL y PostgreSQL, para el caso de SQLite probablemente también se puedan realizar pero el generar regiones circulares puede ser una consulta un tanto tediosa.

3.5. Página Web

Como paso final se necesita una interfaz que permita realizar consultas espaciales sobre la base de datos utilizada, de manera óptima mediante el índice espacial utilizado. Cabe destacar que en esta etapa de la metodología no es necesario que este presente nada de los códigos de procesamiento de imágenes dado que la página web trabajará sobre la base de datos que ya posee todos estos objetos archivados previamente.

No se requiere una gran interfaz para lograr el cometido, debe ser sencilla y permitir realizar los tres tipos de consultas expuestos previamente. En la Figura 3.12 se puede observar un ejemplo de un formulario para realizar una consulta, en este caso la consulta no está escrita por debajo con los valores que el usuario da sino que el usuario da la consulta también.

En la Figura 3.13 se muestra un ejemplo de un formulario de búsqueda por región o búsqueda rectangular. los parámetros necesarios para realizar esta consulta son los mismos que se utilizarán para la búsqueda por región, solo que el ejemplo busca sobre imágenes FITS y no sobre objetos. También obtenido de CasJobs.

Imaging CrossID for DR7

[Scroll down for Help](#)

Search type	The nearest primary object
Search radius [arcmin] (Max 3.0 arcmin)	0.5
Number of preceding non-data columns	1

Cut and paste the **upload list** containing ra,dec or run,rerun,camcol,field,obj here:

```

name ra    dec
A1  195.2  2.5
A2  194.5  2.6
A3  193.6  2.8
  
```

Or upload it as text file [Seleccionar archivo](#) No se ha seleccionado ningún archivo

Type your **SQL query** here (see below for help).

```

SELECT
  p.objID, p.ra, p.dec, p.run, p.rerun, p.camcol, p.field,
  dbo.fPhotoTypeN(p.type) as type,
  p.modelMag_u, p.modelMag_g, p.modelMag_r, p.modelMag_i, p.modelMag_z
FROM #x x, #upload u, PhotoTag p
WHERE u.up_id = x.up_id and x.objID=p.objID
  
```

☒ HTML
 ☐ XML
 ☐ CSV

Figura 3.12: Ejecutando una consulta en CasJobs.

Rectangular Search

NOTE: To be fair to other users, queries run from SkyServer search tools are restricted in how long they can run and how much output they return, by **timeouts** and **row limits**. Please see the [Query Limits help page](#). To run a query that is not restricted by a timeout or number of rows returned, please use the [CasJobs batch query service](#).

195.5	ra	195.58
2.5	dec	2.6

Figura 3.13: Realizando una búsqueda por región.

En la Figura 3.14 se muestra un ejemplo para desplegar la información obtenida por la consulta. La forma de despliegue de información será similar, un estilo de tabla con los valores que correspondan a las características de los objetos.

name	objID	ra	dec	run	rerun	camcol	field	type	modelMag_u	modelMag_g	modelMag_r	modelMag_i	modelMag_z
A1	587726032792060432	195.20342417	2.49847107	1462	40	4	466	GALAXY	25.095808	22.037075	21.243845	20.3144	19.75523
A2	587726032791732324	194.49819749	2.59933786	1462	40	4	461	GALAXY	24.202631	23.874922	23.214554	21.871979	22.201593
A3	587726015611732736	193.59791178	2.80120013	1458	40	4	459	GALAXY	24.05102	22.871647	21.586775	21.633905	20.429312

Figura 3.14: Resultado de la consulta.

En el futuro si es que fuera requerido se puede agregar como funcionalidad que cada objeto tenga ligado a su vez el FITS desde el cual fue obtenido, y de la misma forma permitir descargarlo. En la Figura 3.15 se muestra un ejemplo que permite descargar la imagen FITS después de la consulta.

link	run	rerun	camcol	fields
http://das.sdss.org/www/cgi-bin/segments?LIST=t58Gdc&RUN=1458&RERUN=40&CAMCOL=4	1458	40	4	459
http://das.sdss.org/www/cgi-bin/segments?LIST=t58Gdc&RUN=1462&RERUN=40&CAMCOL=4	1462	40	4	461 466

Figura 3.15: FITS obtenidos de la consulta.

3.6. Resumen

Se hará un resumen con los datos de entrada y salida necesarios para cada paso expuesto de la metodología, así como también algunas herramientas o lenguajes a utilizar.

- **Procesamiento de imágenes** Como datos de entrada para el procesamiento de imágenes se requieren las imágenes FITS a ser procesadas, también se requiere habilitar una tabla en una base de datos para guardar los datos obtenidos en el procesamiento, puede no necesitarse en un principio el índice dado que se archivarán solamente los objetos. Esta capa del proyecto será desarrollada en lenguaje R dado que posee grandes herramientas para la extracción de características, el cual es un paso vital del procesamiento. Con el uso de R se usarán las librerías que sean necesarias para procesar y para luego lograr la conexión a la base de datos y guardar los objetos. Dentro de este mismo paso se calcularán las coordenadas de los objetos para ser archivados, para esto es necesario procesar el header de la imagen. Y para terminar como datos de salida el procesamiento insertará todos los objetos encontrados en la tabla de la base de datos a utilizar.
- **Base de Datos** La capa de base de datos tiene como datos de entrada todos los objetos encontrados, así como también las características extraídas y la posición de cada uno, para ser insertados en una tabla de la base de datos. La base de datos tiene que ser previamente creada al procesamiento de imágenes

para poder guardar la información encontrada. Como datos de salida tendrá la tabla con los objetos y sus características listo para ser indexado para las futuras consultas.

- **Índice de Objetos** La capa del índice necesita tener previamente generada la base de datos y la tabla que se indexará. El índice es solo utilizado al momento de realizar las consultas sobre la base de datos, por lo que no es necesario que esté presente mientras se procesan las imágenes. Generará como datos de salida un índice al cuál se le podrán hacer consultas espaciales y que permitirá obtener los objetos que cumplan con la consulta.
- **Página Web** La capa de la página web está completamente desligada de la capa de procesamiento, dado que lo que necesita de datos de entrada es la base de datos donde están todos los objetos previamente procesados. Este paso es la capa visual de todo el proyecto dado que aquí se desplegarán los resultados de las consultas. Esta capa necesita algún lenguaje que permita la comunicación fluída con base de datos para ejecutar las consultas sobre el índice, la elección de estas herramientas será en torno a desplegar la información de una manera adecuada y de manera sencilla, con el fin que el usuario pueda entenderla y ocuparla. Como datos de salida desplegará los objetos que resulten de cada consulta.

Capítulo 4. Aplicación y validación de la Propuesta

En este capítulo se muestra como se aplicó la propuesta explicada en el capítulo anterior, si es que se siguió paso a paso o tuvo algunos cambios así como también detallando que va ocurriendo en cada paso de la aplicación. Al igual que en el capítulo anterior la aplicación se dividirá en las mismas 4 secciones de interés que son: Procesamiento de imágenes, Base de Datos, Índice de Objetos y Página Web.

Se explicará que fue necesario implementar en el código y se apoyará con imágenes de los procesos, en el capítulo anterior se describió el proceso con una imagen de prueba común y corriente, en este capítulo se realizarán los procesos con imágenes obtenidas directamente desde el observatorio *Sloan Digital Sky Survey*.

Para terminar con la validación de la propuesta donde se evaluará que tan buenos fueron los resultados obtenidos después de aplicar la propuesta.

4.1. Imágenes de prueba

Para aplicar la metodología antes descrita es necesario como primer paso obtener imágenes a las cuales aplicarles todos los pasos de la metodología. Para esto en un principio se logró obtener algunas imágenes provenientes de observatorios astronómicos con la ayuda de Diego Mardones, astrónomo de la Universidad de Chile. Se logró esta cooperación por el hecho de pertenecer a un proyecto en común que busca generar herramientas para el observatorio ALMA ubicado en Chile. Si bien ALMA no trabaja con imágenes ópticas, sino que trabaja con imágenes de radio es necesario poner a prueba las herramientas con imágenes ópticas en un principio por el hecho que son de más fácil acceso y son más estándar. Una vez que empiece a funcionar ALMA completamente será más fácil obtener imágenes y teniendo las herramientas hechas es más fácil el proceso de adaptarlas para soportar imágenes de radiofrecuencia, es más se pueden transformar los cubos de imágenes de radio en varias imágenes mediante varias técnicas que están siendo estudiadas por otra parte

del proyecto en común.

Las imágenes en cuestión son de algunas galaxias o de algunos surveys del cielo. La imagen de prueba para la metodología era parte de estas imágenes. Cabe destacar que antes de realizar cualquier tipo de procesamiento con las imágenes es necesario estudiarlas para lograr obtener de forma correcta la información contenida en el header de la imagen, de lo contrario los cálculos realizados sobre ésta no servirán y los objetos no serán una obtención fidedigna. Por la falta de estandarización en el uso de las claves de los headers es necesario este paso, por otra parte al procesar la matriz que corresponde solo a la imagen y no a su metadata no requiere un estudio previo por cada tipo de imagen, dado que cumplen los mismos patrones.

Como se dijo en la propuesta las bandas que contienen las imágenes a utilizar son cinco: u, g, r, i y z. Se espera que cada imagen de entrada sea procesada en las cinco bandas antes mencionadas con el fin de abarcar la mayor cantidad de objetos y no dejar alguno de lado. Dado que no es incorrecto que un objeto esté presente en las cinco bandas distintas y ocupando la misma posición, pero si es un problema dejar un objeto de lado porque no se tomó la banda a la que pertenece.

4.2. Procesamiento de imágenes

4.2.1. Obtención de Coordenadas

En un principio se parte por analizar las imágenes de prueba a ser procesadas. Para esto es necesario analizar e identificar las distintas claves necesarias del header de la imagen que permiten obtener las coordenadas de la imagen propiamente tal con el fin de luego poder obtener las coordenadas absolutas de cada objeto presente en ésta.

Para obtener estas coordenadas es necesario obtener algunos parámetros desde el header de la imagen en estudio, las claves necesarias son las siguientes:

- **NAXIS1 y NAXIS2:** Correspondientes al tamaño de la imagen, el valor de la claves NAXIS1 trae la cantidad de píxeles que posee el eje x , y la clave NAXIS2 la cantidad de píxeles del eje y de la imagen. En el caso de tratarse de una imagen de radio aparece una nueva variable NAXIS3 que denotaría el tamaño de la dimensión z .
- **CRPIX1 y CRPIX2:** Estas claves corresponden a las coordenadas x e y de un píxel de referencia de la imagen. Este píxel no siempre está en la misma posición por eso es necesario saber en que posición de la imagen está.
- **CRVAL1 y CRVAL2:** Al igual que CRPIX1 y CRPIX2 estas claves están relacionadas con el píxel de referencia, en este caso CRVAL1 y CRVAL2 poseen la coordenada absoluta del píxel de referencia, es decir poseen su coordenada en términos de RA y DEC.
- **CTYPE1 y CTYPE2:** En este caso las claves de CTYPE corresponden al tipo de coordenadas que poseen los valores CRVAL, para este caso y la mayoría de los FITS este tipo corresponde a RA y DEC, que son ascensión recta y declinación, pero también pueden ser otros como coordenadas galácticas.
- **CD1_1, CD1_2, CD2_1, CD2_2:** Los valores CD demuestran la rotación que tiene la imagen, al ser una imagen óptica son solo 4 valores, de ser una imagen radio estos valores pasan a ser 9. Se utilizan para calcular de manera correcta las coordenadas absolutas, de ser 0 estos valores o no estar presente se asume que la imagen no posee rotación.

Para el cálculo de las coordenadas absolutas se debe tener en cuenta la rotación que tiene la imagen, para esto se genera una matriz de rotación que posee los valores de CD, para las imágenes ópticas esta matriz es de 2×2 .

$$\begin{vmatrix} \Delta_{wcs1} \\ \Delta_{wcs2} \end{vmatrix} = \begin{vmatrix} CD1_1 & CD1_2 \\ CD2_1 & CD2_2 \end{vmatrix} * \begin{vmatrix} \Delta_x \\ \Delta_y \end{vmatrix} \quad (4.1)$$

Como se observa en la Ecuación 4.1 se calcula un Δ_{wcs} para cada coordenada, este valor final entrega el desplazamiento que posee el objeto en estudio con respecto al píxel de referencia, este desplazamiento se expresa en las mismas unidades en las que están las coordenadas CRVAL1 y CRVAL2, por lo general RA y DEC.

$$CRABS1 = CRVAL1 - \Delta_{wcs1} \quad (4.2)$$

$$CRABS2 = CRVAL2 - \Delta_{wcs2} \quad (4.3)$$

Con las ecuaciones 4.2 y 4.3 se obtienen finalmente las coordenadas absolutas del objeto ubicado en las coordenadas locales x e y . El valor de CRABS1 pertenece a la coordenada absoluta RA del objeto mientras que el CRABS2 pertenece a la coordenada absoluta DEC.

Las coordenadas absolutas de los objetos son el dato esencial para poder insertarlos en la base de datos y luego consultar acerca de estos. En esta parte del proceso se obtiene solo la posición de los objetos en el cielo. Cabe destacar que el procesamiento de la imagen para obtener los objetos es previo a éste, pero se explica en primera instancia dado que es necesaria la investigación y análisis del header previamente al procesamiento de la imagen.

Para obtener los datos necesarios desde el header se utiliza la Función 1 presente en el Anexo A. Esta función lo que hace es buscar todas las claves necesarias y escribirlas en una lista que luego puede ser utilizada de manera rápida con el fin de evitar tener que buscar las claves en el header cada vez que se requieran. Este proceso se realiza una vez al comienzo antes de empezar a procesar la imagen y la función recibe como parámetro el header de la imagen.

Para obtener las coordenadas locales de cada objeto se utiliza la Función 3 pre-

sente en el Anexo A, esta función recibe como parámetros la información del header previamente obtenida con la Función 1 y además la coordenada local RA y DEC de cada objeto, por ende es necesario llamar esta función por cada objeto encontrado. El proceso de calcular todas las coordenadas de una vez lo realiza la Función 4 también presente en el Anexo A.

4.2.2. Pre Procesamiento

Para el pre procesamiento de la imagen se utilizan herramientas y librerías de R que facilitan esta tarea. En un principio se realiza la lectura de la imagen a procesar, para esto se utiliza la librería FITSio [32] que permite leer y escribir imágenes en formato FITS en conjunto con su metadata presente en el header. Para la lectura de las imágenes se utiliza el método `readFITS` que recibe el nombre de la imagen FITS como parámetro, cabe destacar que la imagen debe estar en el mismo working directory que se puede cambiar en R mediante el comando `setwd`. Al utilizar el `readFITS` la imagen FITS queda guardada en una variable, esta variable tiene tres capas distintas: una capa es el header, otra los datos de la imagen y por último datos de las coordenadas y tamaño de la imagen. Estas tres capas tienen el nombre `header`, `imDat` y `axDat` respectivamente. Para el cálculo de coordenadas es necesario utilizar la información del header y no la capa de `axDat` dado que en esta no está presente la rotación de la imagen, solo la coordenada de referencia con su respectiva coordenada absoluta.

En la Figura 4.1 se muestra la imagen que se utilizará para mostrar los efectos que producen cada filtro durante el procesamiento.

Como previamente se extrajo la información para la obtención de coordenadas ahora la atención se centra en la capa `imDat` que posee una matriz que representa la imagen. Como se menciona en la metodología el primer paso a realizar es identificar la intensidad de la imagen y realizar un corrimiento en el eje z que permita aumentar estas intensidades. Para realizar este procedimiento se utiliza la herramienta *image* de R, que permite aplicarle el filtro a la capa de la imagen del archivo FITS y ade-

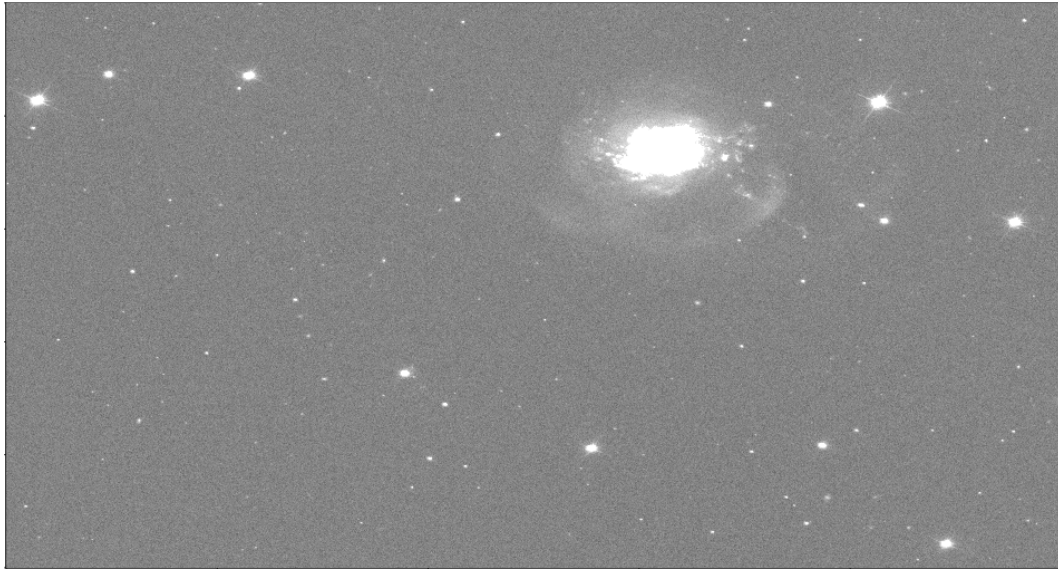


Figura 4.1: Imagen de prueba a la cuál se le aplicará la metodología.

más permite fijar rangos inferiores y superiores de colores con el fin de adecuarse de mejor manera a las condiciones de las imágenes astronómicas. De esta misma forma es factible aplicar un cambio de colores a la imagen aplicando una escala de grises de acorde a las necesidades.

Tal como se predecía en la metodología, al realizar este procedimiento se obtiene una imagen donde gran parte de esta es el fondo, donde los grupos de intensidades se agrupan en su gran mayoría cercano al valor 0. Después de algunas investigaciones y pruebas acerca de fijar un threshold que permita diferenciar lo que son realmente datos en la imagen de lo que pertenece al fondo de la imagen y ver las herramientas que provee R, se dió que la solución se encuentra en la librería EBImage mediante el método de thresh.

El método thresh es una herramienta de threshold adaptativo que es justo el necesario para este proceso dado que este threshold debe adaptarse a las condiciones de la imagen y además a la posición de los objetos dentro de la imagen, dado que no es suficiente fijar un threshold basándose solo en el total de la imagen, sino que es necesario realizar esta evaluación mediante regiones más pequeñas. Para la utilización del método thresh es necesario definir el tamaño que tendrán estas pequeñas

regiones que evaluarán el threshold en la imagen. Los cambios que genera el tamaño de la región en estudio son considerables cuando la región es muy cercana o proporcional al tamaño de la imagen, obteniendo un resultado no muy óptimo. De esta manera es mejor utilizar un tamaño de región más bien pequeño para lograr una mejor estimación del fondo. Cabe destacar que mientras más pequeña la región el algoritmo se demora un poco más en procesar la imagen, pero tampoco es que crezca exponencialmente ni mucho menos.

Luego de varios intentos y pruebas con los tamaños de la región a utilizar se llegó a la conclusión que definir una región de tamaño 10 píxeles por 10 píxeles cumple con el objetivo a cabalidad y no demora más de un segundo en aplicar el threshold a imágenes de 30 MBs.

En la Figura 4.2 se muestra la imagen luego de aplicarle el filtro de threshold, como se puede observar en la imagen este algoritmo lo que hace es diferenciar el fondo de la imagen de cada objeto, la imagen es de color blanca y el fondo o threshold se demuestra mediante el color negro, es por esto que la imagen tiene gran presencia del color blanco, en los objetos más grandes se puede vislumbrar de mejor manera que los bordes de cada uno quedan marcados totalmente con el color negro.

Luego de identificar correctamente el fondo de la imagen se procede a binarizar la imagen mediante la extracción del ruido de ésta, es decir lo único que se quiere dejar presente en la imagen son los objetos astronómicos y eliminar todo el resto de ruido. Para lograr este cometido se utiliza la función de *Opening* y *makeBrush* presentes en la librería EImage de R. El proceso de *Opening* busca eliminar todos los ruidos presentes en la imagen y para esto necesita saber que tipo de objetos están presentes en la imagen, ahí es donde es necesaria la definición de un *brush* que vendría siendo la forma que tendrá el filtro limpiador. Para los efectos de este proyecto la forma que poseen las estrellas es circular y para esto es necesario utilizar un *brush* o filtro de forma de disco.

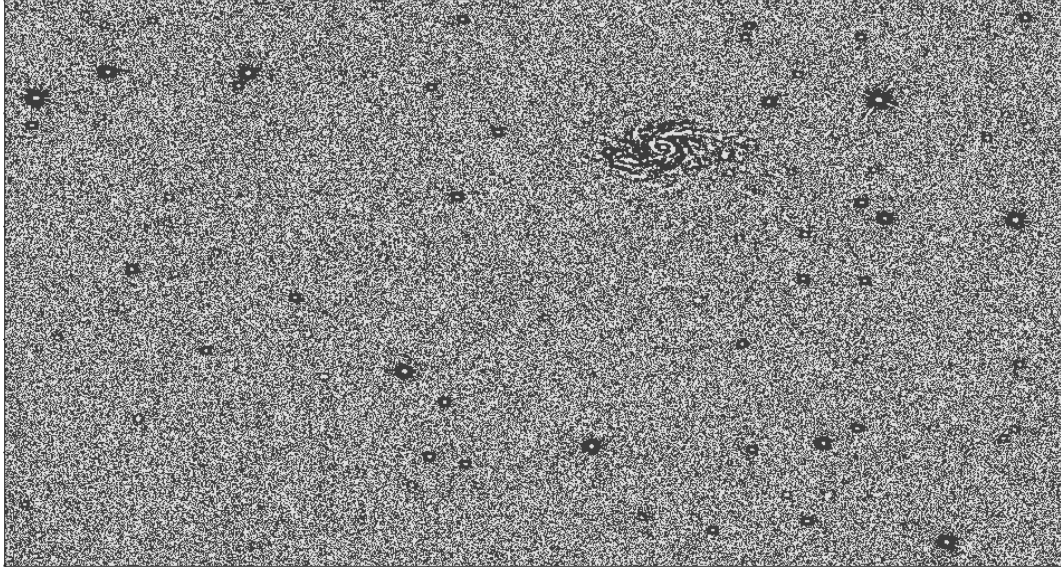


Figura 4.2: *Threshold de la imagen.*

En la Figura 4.3 se muestra la imagen luego de aplicarle el algoritmo de *Opening*, como se puede observar la imagen ahora es mucho más clara y la cantidad de ruido se redujo considerablemente. Es por esto que ahora los objetos se ven demarcados de mejor manera en ésta, donde todo el color negro corresponde al fondo y todos los puntos blancos corresponden a un objeto astronómico. Las diferencias que poseen la Figura 4.1 que corresponde a la imagen original y la Figura 4.3 que está completamente binarizada son notorias, se puede observar que en un principio la imagen poseía una gran mancha blanca que parecía venir de una gran fuente puntual pero luego de adaptar el threshold de la imagen y de reducir el ruido presente en ésta se puede observar que más que una fuente puntual es una concentración grande de estrellas.

Luego de poseer la imagen completamente binarizada y sin ruido externo se realiza el procedimiento de etiquetado de cada objeto, para esto se utiliza la función de *bwlabel* presente en la librería *EBImage* y lo que hace es barrer la imagen de arriba hacia abajo y de izquierda a derecha buscando objetos y poniéndoles un label distinto a cada uno. Al aplicar este procedimiento a las imágenes se ve que los colores de cada objeto es distinto.

En la Figura 4.4 se puede observar que los objetos presentes en la imagen han

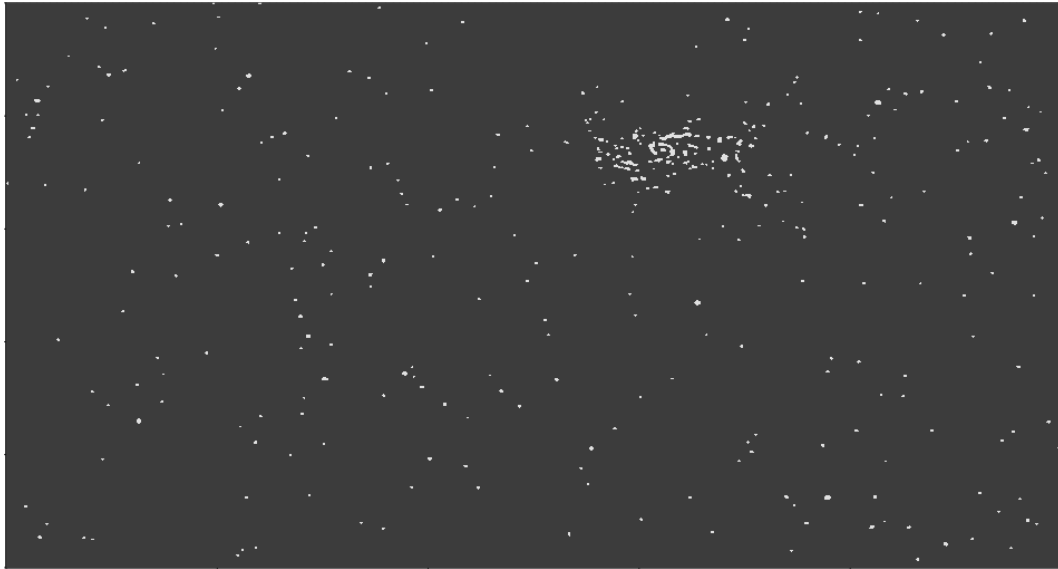


Figura 4.3: Imagen binarizada y sin ruido.

cambiado de color, como la escala de colores utilizada para estas imágenes es la escala de grises, los colores o etiquetas de cada objeto son de los mismos colores, es por esto que mientras más arriba y más a la izquierda estén los objetos en la imagen serán de color blanco, y a medida que se va bajando o yendo hacia la derecha van cambiando a color negro. Esto no quiere decir que los de arriba sean más importante que los de abajo, solo se usa esa convención de colores para mantener un orden.

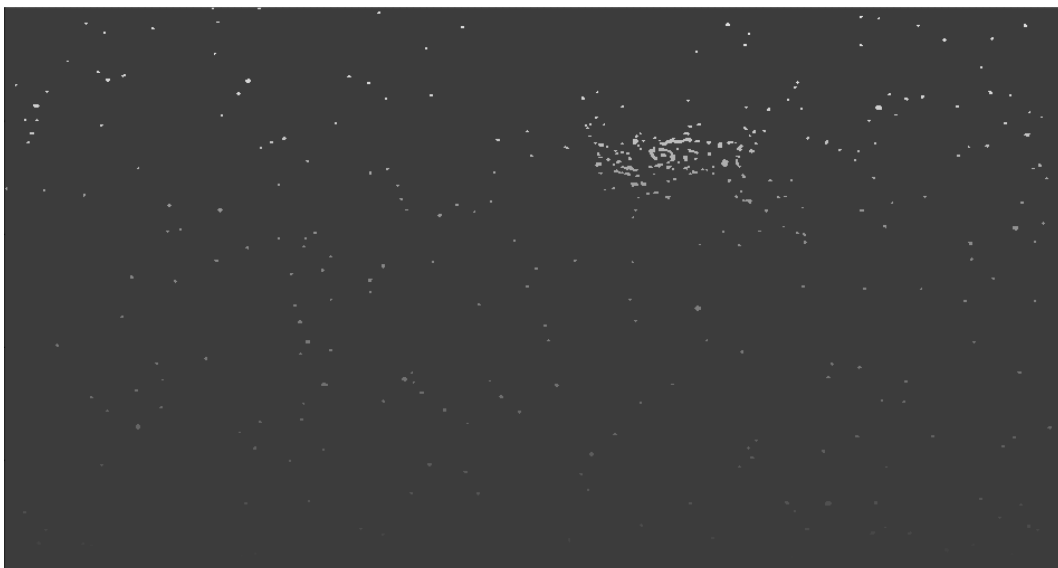


Figura 4.4: Objetos etiquetados en la imagen.

4.2.3. Extracción de características

Para la extracción de características, R mediante la librería *EBImage* presenta una gran cantidad de herramientas que dependiendo el tipo de configuración que se le da puede extraer distintos tipos de características. El método que permite hacer esto se llama *computeFeatures* y la herramienta que utilizaremos es la de *shape*, esto para que extraiga características de la forma de cada objeto presente en la imagen. Cabe destacar que todos los valores de mediciones obtenidos desde estos algoritmos están en unidades de píxeles, por lo que si se quieren guardar y utilizar posteriormente es necesario transformarlos a RA-DEC.

En la Figura 4.5 se muestra lo obtenido por R al aplicar la herramienta de extracción *shape*, la cantidad de objetos encontrados en la imagen fueron 355 por lo que solo se muestran las características de los primeros 14 objetos a modo de ejemplo. Como se puede observar en la tabla obtenida, los valores corresponden al área del objeto, a su perímetro, a su radio mayor, a su radio menor y a la media de radios. Todos estos datos sirven para aplicarle una etapa posterior que permite catalogarlos e identificar a que clase de objetos pertenecen de acuerdo a su forma.

```
> fts <- computeFeatures.shape(x)
> fts
```

	s.area	s.perimeter	s.radius.mean	s.radius.min	s.radius.max
1	36	17	2.939007	2.359550	3.452595
2	30	15	2.657957	2.357023	2.981424
3	21	12	2.157379	2.000000	2.236068
4	21	12	2.157379	2.000000	2.236068
5	28	14	2.592902	2.121320	2.915476
6	32	16	2.732493	2.549510	2.915476
7	89	28	4.882301	4.305460	5.428219
8	21	12	2.157379	2.000000	2.236068
9	28	14	2.592902	2.121320	2.915476
10	53	21	3.714450	2.764367	4.621256
11	30	15	2.657957	2.357023	2.981424
12	45	20	3.352834	2.450000	4.289814
13	43	18	3.300461	2.713137	3.876568
14	33	16	2.822305	1.855103	3.382515

Figura 4.5: Extracción de características de tipo *shape*.

Cabe destacar que mediante la extracción de características se pueden obtener una información mucho mayor a la mostrada previamente, dado que la herramienta de features provee una gran cantidad de métodos y además permite personalizarlos

para realizar búsquedas más exactas de información.

Para obtener la ubicación relativa de los objetos en la imagen se utiliza el método de *moment* que da como resultado la posición x e y de cada objeto en la imagen. Cabe destacar que para todos los métodos de extracción que se utilicen, los objetos en estudio son siempre los mismos que estén presente en la imagen, es decir para este caso particular la obtención de sus coordenadas será para los 355 objetos previamente identificados.

En la Figura 4.6 se muestra que entrega como resultado la función *moment* del módulo de *computeFeatures*, son las coordenadas x e y del centroide de cada objeto encontrado en la imagen.

```
> xy <- computeFeatures.moment(x)[, c('m.cx', 'm.cy')]
> xy
```

	m.cx	m.cy
1	1769.61111	22.86111
2	450.63333	33.63333
3	1525.00000	45.00000
4	460.00000	48.00000
5	1951.50000	52.50000
6	485.50000	54.50000
7	1824.51685	63.05618
8	169.00000	76.00000
9	1875.50000	76.50000
10	65.86792	81.11321
11	151.63333	80.36667
12	1093.44444	88.95556
13	1371.25581	94.53488
14	732.42424	103.03030

Figura 4.6: Extracción de características de tipo *moment* para obtener las coordenadas locales.

4.3. Base de Datos

En la propuesta se expusieron los pro y contra que poseían cada una de las bases de datos SQLite, PostgreSQL y MySQL. Se pusieron a prueba las tres bases de datos para ver cuál es la más indicada y si correspondía con la decisión a priori que sería PostgreSQL.

- **SQLite:** La implementación de una base de datos en SQLite es sencilla y

rápida, al igual que la generación de una tabla para guardar los datos de los objetos. Para implementar el índice de búsquedas espaciales que utiliza un R-Tree es necesario generar una tabla virtual extra, además de la que ya poseen los datos de los objetos. Esta tabla solo posee las columnas de posicionamiento de los objetos, por lo que para utilizar este índice es necesario realizar las consultas sobre la tabla virtual y luego ligarla a la tabla de detalles de los objetos. Las búsquedas espaciales que provee SQLite son básicas, es decir se utilizan los WHERE para definir por ejemplo coordenadas mayores y menores de RA-DEC para abarcar una región rectangular. Realizar una consulta de tipo circular no se ve sencillo en primera instancia, si es que la soportara.

- **MySQL:** La implementación de la base de datos de MySQL no es tan sencilla como la de SQLite pero aún así no es complicada. A diferencia de SQLite el índice se genera sobre la tabla y no es necesario generar otra tabla virtual para realizar las consultas. También permite el uso de R-Tree para generar el índice. Permite realizar búsquedas espaciales mediante polígonos, es decir se puede hacer una consulta de un cuadrado formado por coordenadas mayores y menores de RA-DEC, y a su vez permite las búsquedas circulares. Cabe destacar que MySQL en su módulo abierto tiene limitadas este tipo de consultas, por lo que las de mayor complejidad no pueden ser utilizadas a menos que se pague una licencia.
- **PostgreSQL:** Por último la implementación de una base de datos en PostgreSQL mediante PgAdmin es de manera sencilla y rápida. Para utilizar las características de búsquedas espaciales es necesario descargar el módulo de PostGIS que tiene todo lo relevante a búsquedas espaciales, polígono y otros objetos de interés. La instalación de este módulo en algunos casos puede ser un poco tediosa por tener algunos problemas con otras librerías, pero una vez instalado funciona sin problemas. Al igual que MySQL, PostgreSQL permite realizar búsquedas espaciales complejas como regiones de interés que tengan forma cuadrada o circular mediante el uso de polígonos. Pero PostgreSQL no requiere licencia para este tipo de búsquedas por lo que se puede ocupar el

módulo completo. La implementación del índice R-Tree en las últimas versiones de PostgreSQL se realiza mediante la utilización de un índice GiST. El índice GiST se describió previamente y permite utilizar el árbol acorde a las necesidades de los objetos, por lo que no es necesario especificarle el uso de R-tree.

Toda la investigación apunta al uso de PostgreSQL como motor de base de datos para este tipo de índices y consultas. Por lo que fue la base de datos utilizada en este proyecto.

Para la utilización de PostgreSQL mediante R, para poder insertar los objetos encontrados en la base de datos es necesario utilizar las librerías `sqldf` y `RPostgreSQL` para generar la conexión a la base de datos.

4.4. Índice de objetos

El índice de objetos en PostgreSQL debe ser sobre una columna de tipo geométrica, esto quiere decir que esa columna puede ser un punto, como también puede ser un rectángulo o un polígono más complejo. Para el caso particular del índice de objetos astronómicos se guardó una variable `centroid` que es de tipo geométrica que corresponde al punto del centroide de cada objeto. Para la vista el valor que posee esta columna no es fácil de entender dado que está expresado en muchos números, es por esto que igual se guardan las posiciones RA y DEC del centroide en otras columnas.

En la Figura 4.7 se muestra la tabla utilizada para almacenar los objetos astronómicos. Por el hecho de darle énfasis al índice no se guardaron otras características de cada objeto pero agregar estas características a la tabla es trivial y depende de qué clase de datos se quieren extraer de estos. Las columnas que tiene la tabla son `id`, `centroid` que es de tipo geométrico, `ra`, `dec` y `fitsimage`. `Fitsimage` corresponde al nombre del FITS de donde se extrajo el objeto, se guarda este dato con el fin de que si quiere indagar más en el objeto pueda ir a la fuente a obtener mayor información.

id integer	centroid geometry	ra real	dec real	fitsimage character varying(80)
1	0101000020E61000007607980169816840D03296B05868EF3F	196.044	0.981488	drC-000752-g6-0350.fits
2	0101000020E61000005772E30769816840253422523C51ED3F	196.044	0.916166	drC-000752-g6-0350.fits
3	0101000020E6100000A346015B6481684028EC816EEDC4ED3F	196.044	0.930289	drC-000752-g6-0350.fits
4	0101000020E61000003D9CB297638168404A570C53CBDFEB3F	196.043	0.871069	drC-000752-g6-0350.fits
5	0101000020E6100000E4948488628168407C3789A663C6EE3F	196.043	0.961717	drC-000752-g6-0350.fits
6	0101000020E61000009BA02AEC5E816840F831236283B4EE3F	196.043	0.959535	drC-000752-g6-0350.fits
7	0101000020E6100000BDA3BFE0568168405FBF44097D21ED3F	196.042	0.910338	drC-000752-g6-0350.fits
8	0101000020E61000001916AD7C4281684088B00535406BF03F	196.039	1.02618	drC-000752-g6-0350.fits
9	0101000020E61000001C2C69D940816840A7C203EEDA26EC3F	196.039	0.879743	drC-000752-g6-0350.fits
10	0101000020E61000005A2E01A54081684078E650168497EA3F	196.039	0.830996	drC-000752-g6-0350.fits
11	0101000020E610000026504B7B34816840866986197FE6ED3F	196.038	0.934387	drC-000752-g6-0350.fits

Figura 4.7: Tabla de objetos.

4.4.1. Consultas espaciales

Operador && El Operador && se utiliza para saber si una región o polígono en estudio se sobrepone y cubre totalmente a otra región o polígono en estudio. Para el caso particular de las búsquedas espaciales se utilizará para saber si el punto del centroide del objeto pertenece al rectángulo o círculo dependiendo sea el caso.

Consultas por regiones rectangulares Para realizar una consulta mediante una región rectangular y aprovechar el índice generado sobre la columna geométrica centroid es necesario no utilizar la cláusula \geq o \leq sobre el valor RA o DEC sino que se evalúa si un objeto pertenece al rectángulo en cuestión mediante el operador &&. Para realizar una consulta rectangular es necesario generar un rectángulo mediante dos puntos, el punto inferior izquierdo (que corresponde al RA y DEC menor) y el punto superior derecho (que corresponde al RA y DEC mayor), para esto se utiliza el método ST_MakeBox2D que recibe como parámetro 2 puntos y genera un rectángulo. La implementación de la consulta rectangular se puede ver en el Anexo 8.

Consultas por regiones circulares Para realizar una consulta circular se utiliza la función ST_Buffer(geom,distancia) donde el geom corresponde a un punto geométrico y la distancia vendría siendo el radio. Esta función no retorna un círculo exacto, el círculo es generado mediante un polígono aproximado de 8 lados, la pre-

cisión de este círculo se puede aumentar agregando un tercer parámetro a la función que corresponde a la cantidad de lados del polígono. En la Figura 4.8 se muestra como se realiza el círculo mediante un polígono [33].

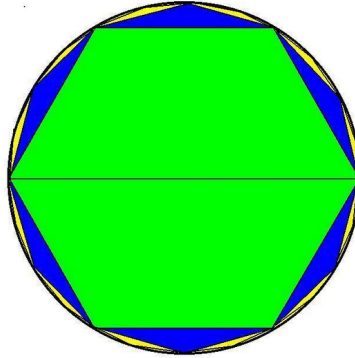


Figura 4.8: Generación de una región circular.

La consulta circular requiere como parámetros el punto central mediante coordenadas RA-DEC y el radio del círculo. Para la consulta de los vecinos cercanos se utiliza algo similar solo que el radio está previamente definido y tiene un valor más pequeño. Ambos tipos de consulta están presentes en el Anexo 8.

4.5. Página Web

La página web se generó con la herramienta Yii, que es un framework de php que permite la comunicación con las bases de datos de una manera rápida y sencilla. Como página principal, se encuentra la posibilidad de seleccionar una de las tres consultas a realizar: en primer lugar, la consulta mediante región rectangular, la consulta mediante un radio y un punto que corresponde a una consulta circular, y por último una consulta de búsqueda de vecinos cercanos, que posee un radio previamente definido. En la Figura 4.9 se observa la pantalla principal de la página web del índice.

La interfaz de la página web es intuitiva, con el fin de facilitar su uso y obtener los datos de una manera ordenada. En la Figura 4.10, se observan los campos necesarios para realizar una consulta rectangular. Los campos son coordenadas RA-DEC menores y mayores.

En la Figura 4.11, se muestra la pestaña que corresponde a la consulta circular.

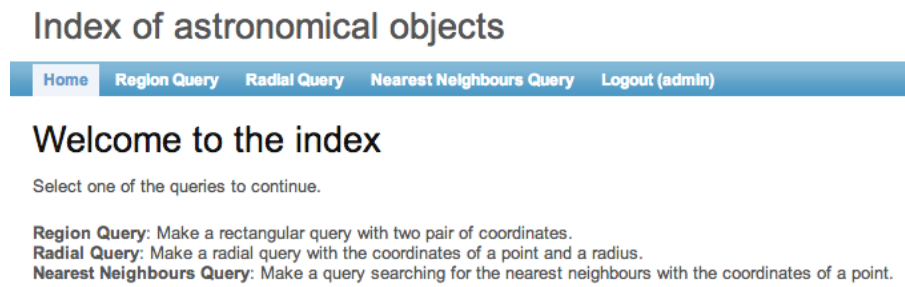


Figura 4.9: Página principal.

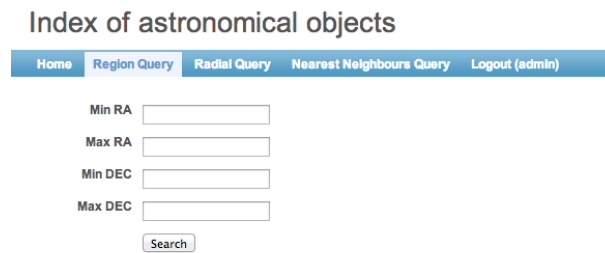


Figura 4.10: Consulta mediante región rectangular.

Para realizar esta consulta es necesario ingresar las coordenadas RA-DEC que corresponden a un punto en el espacio, y además, el radio del círculo. La consulta de vecinos cercanos funciona de manera similar, pero no posee el campo de radio dado que fue previamente configurado.

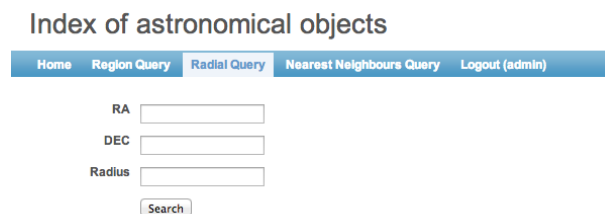


Figura 4.11: Consulta mediante región circular.

En la Figura 4.12, se muestra la tabla con las respuestas de las consultas previamente expuestas. Se utiliza una tabla para mantener un orden: ésta tabla permite ordenar mediante uno de los campos haciendo click en el nombre, además permite buscar un valor en particular para refinar la precisión de la consulta.

Index of astronomical objects

Home Region Query Radial Query Nearest Neighbours Query Logout (admin)

[Back](#)

Displaying 1-10 of 314459 results.

ID	Ra	Dec	Fitsimage
1	196.044	0.981488	drC-000752-g6-0350.fits
2	196.044	0.916166	drC-000752-g6-0350.fits
3	196.044	0.930289	drC-000752-g6-0350.fits
4	196.043	0.871069	drC-000752-g6-0350.fits
5	196.043	0.961717	drC-000752-g6-0350.fits
6	196.043	0.959535	drC-000752-g6-0350.fits
7	196.042	0.910338	drC-000752-g6-0350.fits
8	196.039	1.02618	drC-000752-g6-0350.fits
9	196.039	0.879743	drC-000752-g6-0350.fits
10	196.039	0.830996	drC-000752-g6-0350.fits

Go to page: < Previous 1 2 3 4 5 6 7 8 9 10 Next >

Figura 4.12: Despliegue de información mediante tabla.

4.6. Escalabilidad

4.6.1. Descarga de imágenes

Para probar el funcionamiento del índice y de los algoritmos de procesamiento, se utilizó una pequeña cantidad de imágenes obtenidas desde observatorios ópticos. Para validar este comportamiento es necesario obtener una mayor cantidad de imágenes para poner a prueba los algoritmos, así como también la base de datos bajo una carga mayor, con el fin de observar el desempeño del índice de objetos.

Para obtener grandes cantidades de imágenes, es necesario realizar un *mass download* desde algún observatorio. El observatorio escogido fue el SLOAN, dado que sus imágenes son *surveys* del cielo donde cada imagen posee grandes cantidades de objetos. El SLOAN tiene una herramienta que permite buscar archivos FITS dada unas ciertas coordenadas RA-DEC. Dado que es un trabajo tedioso ingresar una a una todas las coordenadas de los objetos que se quieren descargar, se recurrió a generar pares de coordenadas de forma aleatoria.

En la Figura 4.13, se observa el formulario de SLOAN para obtener imágenes FITS. En la parte superior se define un radio y el criterio de búsqueda que se implementará frente al punto RA-DEC, como por ejemplo, el radio mínimo. Si se desea, se puede buscar el objeto que calce mejor con la descripción. En el segundo formulario se puede poner una lista de puntos a buscar con el siguiente formato “A1 23 56”, que quiere decir que el punto se llama A1 y sus coordenadas RA son 23 y DEC 56. Para obtener grandes cantidades de imágenes se ingresaron 800 puntos distintos en la búsqueda. Por último la query que está en el tercer formulario se dejó bajo la configuración predeterminada.

Imaging CrossID for DR7 [Scroll down for Help](#)

Search type	The nearest primary object
Search radius [arcmin] (Max 3.0 arcmin)	0.5
Number of preceding non-data columns	1

Cut and paste the **upload list** containing ra,dec or run,rerun,camcol,field,obj here:

name	ra	dec
A1	195.2	2.5
A2	194.5	2.6
A3	193.6	2.8

Or upload it as text file [Seleccionar archivo](#) No se ha seleccionado ningún archivo

Type your **SQL query** here (see below for help).

```
SELECT
  p.objID, p.ra, p.dec, p.run, p.rerun, p.camcol, p.field,
  dbo.fPhotoTypeN(p.type) as type,
  p.modelMag_u, p.modelMag_g, p.modelMag_r, p.modelMag_i, p.modelMag_z
FROM #x x, #upload u, PhotoTag p
WHERE u.up_id = x.up_id and x.objID=p.objID
```

Submit Format ☒ HTML ☐ XML ☐ CSV Reset

Figura 4.13: Consulta para obtener imágenes FITS.

Al realizar la consulta, varios objetos fueron encontrados y desplegados como se muestra en la Figura 4.14. Al final de la página que muestra los objetos, se entrega la opción de subir esa lista al DAS¹ de SLOAN para descargarlos, es posible descargarlos uno a uno o también descargar un archivo .lis que posee la lista de todas las url de imágenes FITS, que mediante un wget pueden ser descargadas secuencialmente.

En la Figura 4.15 observa el detalle de la descarga de un total de 835 archivos FITS que pesan 4,8Gb, los cuales fueron descargados en aproximadamente 5 horas.

¹Data Archive Server, servidor con imágenes y datos del SLOAN.

CAPÍTULO 4 APLICACIÓN Y VALIDACIÓN DE LA PROPUESTA

name	objID	ra	dec	run	rerun	camcol	field	type	modelMag_u	modelMag_g	modelMag_r	modelMag_i	modelMag_z
A1	587726032792060113	195.23053537	2.49935472	1462	40	4	466	STAR	19.63641	19.532883	19.294407	19.293533	19.227642
A2	587726032791732324	194.49819749	2.59933786	1462	40	4	461	GALAXY	24.202631	23.874922	23.214554	21.871979	22.201593
A3	587726015611732736	193.59791178	2.80120013	1458	40	4	459	GALAXY	24.05102	22.871647	21.586775	21.633905	20.429312
A6	587738066730812091	120.00666052	52.99905345	4264	40	3	119	STAR	23.557898	21.760603	21.300106	20.874533	20.47286
A7	587737825142440003	129.99830245	56.00368614	4208	40	1	173	STAR	25.512268	21.749025	20.186476	19.321476	18.902908
A17	758882837984117518	271.0022513	64.99887461	6123	648	5	47	STAR	22.841148	21.818892	20.826866	20.56538	20.315094
A22	587742061075825333	191.99527532	21.99890501	5194	40	3	507	GALAXY	23.165104	23.384497	23.011908	21.29986	20.439083
A24	587733427624083779	222.99923254	53.00390291	3184	40	2	40	STAR	25.234882	22.764456	21.479504	20.650129	19.881014
A26	587739708486910056	223.99966184	25.00020658	4646	40	5	193	STAR	24.305717	21.344507	19.947701	19.267078	18.892612
A28	587732772105618361	128.0007783	5.99545538	3031	40	5	54	STAR	23.289736	23.726362	22.358698	20.782436	19.892338
A31	587725576960802952	259.99443139	62.99848133	1356	40	3	41	STAR	20.653118	19.469181	18.99972	18.817648	18.69426
A42	587722984438431919	212.00215518	0.9962087	752	40	6	457	STAR	22.620003	20.089096	18.763145	18.010387	17.616531
A46	587733081887539717	187.99744419	55.99625537	3103	40	6	167	GALAXY	22.137472	22.11487	22.347452	22.122898	22.615499
A50	587729228226953778	242.99677642	51.99896921	2206	40	4	149	GALAXY	21.769201	20.580881	20.151167	19.824341	19.765667
A52	587742013271179781	154.00622964	18.9988941	5183	40	2	154	STAR	24.611309	22.934444	21.749525	20.474977	19.739904
A58	588882136301503491	90.01232742	82.99945079	5960	648	2	164	GALAXY	27.618044	21.760239	21.373543	20.086634	19.52314
A63	587738066187649268	110.99736286	40.00337711	4264	40	2	23	GALAXY	25.879105	22.647738	21.335228	20.351328	19.866024
A67	587729229301023141	244.99270838	51.99917918	2206	40	6	154	STAR	22.595703	22.681452	21.756088	21.244165	21.114985
A71	587741532769420170	132.00008151	18.9997581	5071	40	3	119	GALAXY	25.062782	23.950916	22.60853	22.316145	21.432148
A73	588010031368034654	130.00331019	0.00806708	7520	41	6	61	STAR	23.306164	23.187403	21.406381	21.320068	20.714848

Figura 4.14: Objetos encontrados con la consulta.

Cada archivo pesa alrededor de 6 Mbs.

```
--2013-11-05 23:07:48-- http://das.sdss.org/www/cgi-bin/drC?RUN=7243&RERUN=40&C
AMCOL=6&FIELD=96&FILTER=z
Reusing existing connection to das.sdss.org:80.
HTTP request sent, awaiting response... 200 OK
Length: unspecified [image/fits]
Saving to: 'www/cgi-bin/drC-007243-z6-0096.fits'

[      <==>      ] 6,117,120   260KB/s   in 35s

2013-11-05 23:08:23 (173 KB/s) - 'www/cgi-bin/drC-007243-z6-0096.fits' saved [61
17120]

FINISHED --2013-11-05 23:08:23--
Total wall clock time: 5h 1m 40s
Downloaded: 835 files, 4.8G in 4h 51m 3s (286 KB/s)
```

Figura 4.15: Descarga de imágenes.

Para procesar todas estas imágenes se realizó una función que obtenía todos los nombres de los archivos presentes en una carpeta, y les aplicaba el algoritmo de procesamiento. Su ejecución demoró aproximadamente 5 horas y dió como resultado 314.000 objetos encontrados y archivados en la tabla objetos de la base de datos. La ejecución del algoritmo, para todas las imágenes, se demoró aproximadamente 6 horas, entregando resultados óptimos sin ningún error.

Cabe destacar, que el algoritmo funciona de forma adecuada siempre y cuando el formato del FITS sea el mismo. Si algún otro observatorio quisiera indexar objetos dentro de la misma base de datos habría que adaptar las claves de ubicación al mismo tipo que las que poseen las del SLOAN.

4.6.2. Tiempo de Procesamiento de Consultas

Para poder validar el desempeño del índice, se realizaron los tres tipos de consulta previamente expuestos, obteniendo distintas magnitudes de objetos en cada una, lo que permite verificar los distintos tiempos de respuesta obtenidos. En la Tabla 4.1 se muestra el detalle de las 12 consultas realizadas. En la columna consulta se muestra el tipo de consulta, en la columna parámetros se muestran los parámetros utilizados, que son distintos para cada tipo de consulta. En la columna número de objetos se muestran los objetos obtenidos al realizar la consulta, en la columna tiempo se expresa el tiempo que tomó realizar la consulta medida en milisegundos y por último en la última columna se muestran las columnas que fueron pedidas mediante la consulta. Como se puede observar los mayores tiempos ocurrieron al consultar por todas las columnas y abarcando todos los objetos en el índice, éste tiempo tiene una baja considerable al preguntar por sólo una columna, lo cuál se podría aprovechar consultando por solo las columnas necesarias.

Consulta	Parámetros	Nº Objetos	Tiempo	Columnas
Region Query	RA:(-30,36), DEC:(-30,360)	314459	15257 ms	Todas
Region Query	RA:(-30,36), DEC:(-30,360)	314459	2317 ms	id
Region Query	RA:(-30,36), DEC:(-30,360)	314459	3953 ms	id,ra
Region Query	RA:(100,150), DEC:(50,100)	11077	646 ms	Todas
Region Query	RA:(0,200), DEC:(30,50)	25178	1417 ms	Todas
Radial Query	Punto:(180,180), Radio:210	314459	16367 ms	Todas
Radial Query	Punto:(50,100), Radio:40	19884	1023 ms	Todas
Radial Query	Punto:(50,100), Radio:40	19884	458 ms	id,ra,dec
Radial Query	Punto:(200,50), Radio:10	1620	178 ms	Todas
Neighbour Query	Punto:(190,52)	1064	146 ms	Todas
Neighbour Query	Punto:(78,62)	3209	247 ms	Todas
Neighbour Query	Punto:(161,64)	961	137 ms	Todas

Tabla 4.1: Tiempos de procesamiento de consultas.

Capítulo 5. Conclusiones

5.1. Respecto a la metodología

Desarrollar una metodología que permitiera abarcar todo el desarrollo del índice no fue una tarea fácil. Por esto mismo, fue necesario dividir el desarrollo en 4 capas distintas de forma, para trabajar iterativamente sobre cada parte por separado. Este tipo de división permite el poder ir avanzando de forma simultánea en más de una capa de la herramienta, donde en algunos casos una capa dependía de otra, pero se aprovechaban estas dependencias para avanzar y construir las demás etapas.

Dentro de las 4 etapas en que se dividió el proyecto, cabe destacar que la que tomó mayor tiempo de desarrollo fue la de procesamiento de imágenes, debido a que requirió una mayor cantidad de investigación para lograr los objetivos y su funcionamiento óptimo. Sin lugar a dudas es la parte primordial del proyecto donde se puso mayor énfasis.

La obtención de las coordenadas absolutas fue algo tedioso, dado que requería un mayor entendimiento sobre astronomía, por lo que fue necesario pedir ayuda a un astrónomo perteneciente al proyecto: FONDEF. Una vez entendido como funcionaba, ésta redujo su complejidad.

La extracción de características es un proceso complejo que requiere un buen entendimiento de las herramientas para lograrlo, mediante la ayuda de R y sus librerías esta tarea pudo ser lograda con buenos resultados. Se requirió una calibración especial y más específica para que funcionara de manera adecuada con las imágenes astronómicas.

La necesidad de desplegar la información obtenida desde la base de datos de una forma fácil y efectiva, dió como resultado la utilización de un framework que fue especialmente diseñado para manejar datos de una manera fácil y rápida, por eso

se utilizó Yii que es un framework basado en php. Es versátil, por lo que permite acoplarse a cualquier otro sitio sin mayores inconvenientes.

5.2. Respecto a los objetivos

La investigación sobre el formato FITS resultó como se esperaba: se logró obtener toda la información necesaria para obtener las coordenadas absolutas de cada objeto presente en la imagen, a pesar que el header de los FITS no es estándar, igual los nombres de las claves no son complejos de entender, y aunque se escriban con notación diferente, los valores corresponden a los mismos índices. Dependiendo el observatorio, esto requiere un pequeño ajuste que no toma mucho tiempo.

La elección del motor de base de datos fue acertada, dado que las consultas que se necesitan realizar fueron soportadas de manera óptima. La mayoría de los sitios que hacen benchmark de las bases de datos en términos de búsquedas espaciales, recomiendan usar PostgreSQL por su versatilidad. A veces puede complicarse su uso por la instalación del paquete PostGIS, pero una vez instalado todo funciona se espera.

La segmentación mediante el algoritmo de convex hull funcionó de manera apropiada y permitió identificar los objetos en la imagen. La identificación de objetos compuestos como galaxia requieren de un trabajo mayor, que puede ser profundizado en una investigación posterior.

La generación del índice se logró a cabalidad, cumpliendo con todos los objetivos esperados. El software fue desarrollado de forma escalable, probando alrededor de 314.000 objetos en una base de datos donde la consulta tipo no tomó más de dos segundos y fue probado con un computador personal como servidor.

El tiempo de respuesta del índice en su peor caso es de 16367 ms, el cual es

acorde a la cantidad de columnas requeridas y la cantidad de objetos obtenidos. Cabe destacar que el índice en la realidad poseerá una cantidad de objetos considerablemente mayor, pero a su vez, las regiones de consulta serán más pequeñas.

5.3. Respecto al aprendizaje

En un comienzo, la investigación cumplió un rol fundamental, donde los requerimientos no estaban claros y los objetivos fueron variando acorde a las necesidades que se iban presentando. El tópico de mayor aprendizaje tuvo que ver con el área de astronomía, es un tema un tanto lejano de los informáticos en la actualidad, pero que tiene un gran campo que va creciendo a pasos agigantados.

Los informáticos tienen mucho que aportar a los proyectos de astronomía, donde gran parte de los mayores desafíos que tienen que derribar requieren de una herramienta informática que les permita visualizar, optimizar, transportar y procesar grandes cantidades de imágenes. La comunidad de astrónomos de Chile y el mundo, posee un carácter colaborativo muy amplio, dado que requieren de la experiencia y habilidad de distintos perfiles de profesionales para investigar los diferentes fenómenos que el espacio sideral ofrece.

El procesamiento de imágenes es una tarea que se ha abarcado desde distintas aristas y de múltiples formas con el paso del tiempo, por este mismo motivo, hay grandes cantidades de información al respecto de las herramientas que benefician ciertos comportamientos o que ayudan a buscar ciertos rasgos en una imagen. Por otra parte, las herramientas de astronomía están en pleno desarrollo y no poseen la misma madurez. Por lo que el apoyo de los astrónomos fue vital en varias partes del proyecto, lo que además tuvo un gran impacto en el aprendizaje obtenido.

5.4. Respecto a trabajos futuros

En el paso de procesamiento de imágenes, cuando se busca lograr la binarización, se hace con un método de brush circular con un radio definido de 5 píxeles. Este radio puede ser incrementado para buscar objetos de mayor tamaño, lo que podría ayudar a identificar una galaxia, por ejemplo, y un proceso de optimización que podría generar buenos resultados, es realizar este procesos con distintos tamaños de radio con el fin de buscar objetos de mayor tamaño.

Como proceso de validación, sería ideal comprobar si el objeto está realmente en la coordenada encontrada, y si realmente corresponde a un objeto astronómico y no una interferencia. Este proceso se podría realizar catalogando los objetos encontrados de acuerdo a sus características. Sin embargo, este trabajo no es trivial, y requiere mucho trabajo, pero si sería sumamente útil y poderoso. Otra forma de validación posible, puede ser encontrar dos imágenes FITS que se superpongan y ver que en la regiones de superposición hayan los mismos objetos. Esto podría ayudar a descartar interferencia, así como también posibles cambios a lo largo del tiempo, dependiendo cuando fueron tomadas las dos imágenes. La parte interesante que este desarrollo podría entregar, es el encontrar nuevos objetos.

La siguiente etapa para soportar imágenes provenientes de ALMA, es tomar el cubo de imágenes proveniente del mismo y tomar cada slice del cubo como si fuera una imagen independiente, para luego aplicarle la transformada de Fourier, llevando la imagen de radio al espectro óptico, y finalmente, aplicar el procesamiento a todas las slice del cubo. Esta extensión a otros observatorios se puede realizar con los que sea necesario, tomando en cuenta la naturaleza de los datos en el encabezado y los tipos de imagen.

El índice generado puede ser adaptado para funcionar con todos los observatorios del mundo, quizás en algunos requiera ciertas modificaciones, pero pueden adaptarse. Sería interesante tener un índice espacial que abarque imágenes de to-

dos los observatorios del mundo, pero es un sueño algo utópico de acuerdo a las herramientas y cantidad de información que se maneja hoy en día. Para lograr este cometido, sería vital guardar mayor cantidad de información en la base de datos con el fin de identificar a cabalidad la proveniencia del objeto.

Las herramientas de astronomía están en pleno crecimiento, donde se hace necesario innovar y generar nuevos acercamientos a los mismos problemas para obtener distintos resultados. En Chile, con la instalación de los nuevos observatorios, se abre una ventana de oportunidad que puede permitir el crecimiento astronómico de la región, y transformarlo en una potencia mundial de astronomía. En términos de observación del cielo, Chile poseerá el mayor porcentaje de observación, por lo que es necesario formar nuevos profesionales de la astronomía o de las distintas carreras que puedan colaborar con ésta, a fin de explotar todo su potencial de manera local y que no sea necesario pedir a profesionales de otros países que se hagan cargo del desarrollo astronómico de nuestro país.

Bibliografía

- [1] Edge detection. <<http://www.cse.unr.edu/~bebis/CS791E/Notes/EdgeDetection.pdf>>. [en línea] [consulta: 27-10-2012].
- [2] Fionn. STARCK, Jean-Luc y MURTAGH. *Handbook of Astronomical Data Analysis*. Springer-Verlag, 2009. Fifth Edition.
- [3] SExtractor. <<http://www.astromatic.net/software/sextractor>>. [en línea] [consulta: 27-10-2012].
- [4] A. BIJAOU. Sky background estimation and application. *Astronomy and Astrophysics*, Apr. 1980. Vol. 84, no. 1-2, p. 81-84.
- [5] Transformación de wavelets. <<http://sabia.tic.udc.es/gc/Contenidosadicionales/trabajos/Imagenyvideo/compresion/3.2.3.2.htm>>. [en línea] [consulta: 29-10-2012].
- [6] Matlab. <<http://www.mathworks.com/products/matlab/>>. [en línea] [consulta: 05-11-2012].
- [7] Octave. <<http://www.gnu.org/software/octave/>>. [en línea] [consulta: 05-11-2012].
- [8] Scilab. <<http://www.scilab.org>>. [en línea] [consulta: 05-11-2012].
- [9] R project. <<http://www.r-project.org/>>. [en línea] [consulta: 30-12-2012].
- [10] Casa: Common astronomy software application. <<http://casa.nrao.edu/>>. [en línea] [consulta: 11-05-2013].
- [11] Virtual observatory. <<http://www.virtualobservatory.org/faq.aspx>>. [en línea] [consulta: 05-11-2012].
- [12] Fits format. <<http://fits.gsfc.nasa.gov/>>. [en línea] [consulta: 05-03-2012].

BIBLIOGRAFÍA

- [13] Sloan digital sky survey. <<http://www.sdss.org/>>. [en línea] [consulta: 25-03-2012].
- [14] Sistema de coordenadas geográficas. <<http://astrologia.about.com/od/glosario/g/Sistema-De-Coordenadas-Geogr-Aficas.htm>>. [en línea] [consulta: 30-03-2013].
- [15] Celestial equatorial coordinate system. <http://astro.unl.edu/naap/motion1/cec_units.html>. [en línea] [consulta: 05-04-2013].
- [16] Oleg Skylar. Adaptive thresholding, ebimage r package, 2007.
- [17] Gaussian smoothing. <<http://homepages.inf.ed.ac.uk/rbf/HIPR2/gsmooth.htm>>. [en línea] [consulta: 10-05-2013].
- [18] Morphological image processing. <<http://www.cs.auckland.ac.nz/courses/compsci773s1c/lectures/ImageProcessing-html/topic4.htm>>. [en línea] [consulta: 25-05-2013].
- [19] Gregoire Pau. Binary segmentation, ebimage r package, 2009.
- [20] G. Pau and O. Skylar. Fill holes in objects, ebimage r package, 2009.
- [21] K. Haralick, R. Shanmugan and Deinstein H. Textural features for image classification, iee transactions on systems, man and cybernetics, 1973. 3(6):610-621.
- [22] Casjobs. <<http://casjobs.sdss.org/dr7/en/>>. [en línea] [consulta: 20-06-2013].
- [23] Sqlite. <<http://sqlite.org/>>. [en línea] [consulta: 25-06-2013].
- [24] Sqlite r-tree módulo. <<http://sqlite.org/rtree.html>>. [en línea] [consulta: 25-06-2013].
- [25] Top reasons to use mysql. <<http://www.mysql.com/why-mysql/topreasons.html>>. [en línea] [consulta: 10-07-2013].

BIBLIOGRAFÍA

- [26] Postgresql. <<http://www.postgresql.org/>>. [en línea] [consulta: 15-07-2013].
- [27] Postgis. <<http://postgis.net/>>. [en línea] [consulta: 15-07-2013].
- [28] Ralf Hartmut G. Spatial database systems tutorial. Oct. 1994. Special issue on spatial database systems of the VLDB Journal (Vol. 3, No4).
- [29] B-trees definition. <<http://www.cse.ohio-state.edu/~gurari/course/cis680/cis680Ch13.html>>. [en línea] [consulta: 25-07-2013].
- [30] Antonin Guttman. R-trees: A dynamic index estructura for spatial searching. Jun. 1984. University of California, ACM New York, p 47-57.
- [31] Gist indexes in postgresql. <<http://www.postgresql.org/docs/8.1/static/gist.html>>. [en línea] [consulta: 01-08-2013].
- [32] Andrew Harris. Fitsio: Fits (flexible image transport system) utilities, 2013.
- [33] Circle in postgis. <<http://theworldofapenguin.blogspot.com/2008/06/circles-in-postgis.html>>. [en línea] [consulta: 20-11-2013].

A. Anexos

1.1. Códigos

```
1 getCoordData <- function(header){
2   return (list(CRVAL1=as.numeric(
3     header[which(header=="CRVAL1")+1]),
4     CRVAL2=as.numeric(header[which(header=="CRVAL2")+1]),
5     CRPIX1=as.numeric(header[which(header=="CRPIX1")+1]),
6     CRPIX2=as.numeric(header[which(header=="CRPIX2")+1]),
7     CD1_1=as.numeric(header[which(header=="CD1_1")+1]),
8     CD1_2=as.numeric(header[which(header=="CD1_2")+1]),
9     CD2_1=as.numeric(header[which(header=="CD2_1")+1]),
10    CD2_2=as.numeric(header[which(header=="CD2_2")+1]),
11    NAXIS1=as.numeric(header[which(header=="NAXIS1")+1]),
12    NAXIS2=as.numeric(header[which(header=="NAXIS2")+1])))
13 }
```

Función 1: Obtención de la información necesaria del header para el cálculo de coordenadas.

```
1 y <- im$imDat
2 image(y, zlim=c(1100,1200), col=gray.colors(256))
3 x <- thresh(y, 10, 10, 0)
4 x = opening(x, makeBrush(5, shape='disc'))
5 x <- bwlabel(x)
6 fts <- computeFeatures.shape(x)
7 xy <- computeFeatures.moment(x)[, c('m.cx','m.cy')]
8 return(list(xy,fts))
9 }
```

Función 2: Obtención de los objetos de la imagen.

```
10 wcs1 <- data$CD1_1*(xloc-data$CRPIX1) +
11 data$CD1_2*(yloc-data$CRPIX2)
12 wcs2 <- data$CD2_1*(xloc-data$CRPIX1) +
13 data$CD2_2*(yloc-data$CRPIX2)
14 xabs <- data$CRVAL1 - wcs1
15 yabs <- data$CRVAL2 - wcs2
16 return (list(xab=xabs,yab=yabs))
17 }
```

Función 3: Obtención de las coordenadas absolutas de cada objeto.

```
1 dataObject <- matrix(, nrow=length(objects)/2,ncol=8)
2 count <- 1
```

```

3 lengthObjects <- length(objects)/2
4 lengthFts <- length(fts)/5
5 for (i in objects) {
6   if(count > length(objects)/2){
7     break}
8   coordAbs <- getAbsCoord(data,
9     objects[count],objects[lengthObjects+ count])
10
11   dataObject[count,1] <- objects[count]
12   dataObject[count,2] <- objects[lengthObjects +
13     count]
14
15   dataObject[count,3] <- coordAbs$xab
16   dataObject[count,4] <- coordAbs$yab
17   radius <- ceiling(fts[lengthFts*4 +count])
18   Xmin <- objects[count] - radius
19   Xmax <- objects[count] + radius
20   Ymin <- objects[lengthObjects + count] - radius
21   Ymax <- objects[lengthObjects + count] + radius
22   aux <- getAbsCoord(data, Xmin, Ymin)
23   aux2 <- getAbsCoord(data, Xmax, Ymax)
24   if(aux$xab >aux2$xab)
25   {
26     dataObject[count,7] <- aux$xab
27     dataObject[count,5] <- aux2$xab
28   }
29   else
30   {
31     dataObject[count,5] <- aux$xab
32     dataObject[count,7] <- aux2$xab
33   }
34   if(aux$yab >aux2$yab)
35   {
36     dataObject[count,8] <- aux$yab
37     dataObject[count,6] <- aux2$yab
38   }
39   else
40   {
41     dataObject[count,6] <- aux$yab
42     dataObject[count,8] <- aux2$yab
43   }
44   count <- count+1
45 }
46 dimnames(dataObject) = list(c(1:(count-1)),
47   c("xloc","yloc","xabs","yabs","minRA",
48     "minDEC","maxRA","maxDEC"))
49 return(dataObject)

```

50 }

Función 4: Traspaso de todas las coordenadas locales a absolutas.

```

1 {
2   count <- 1
3   query <- ""
4   while(1){
5     if(count > length(objects)/8) break
6     aux <- sprintf("(%s,%s,ST_SetsRID("+
7       "ST_MakePoint(%s,%s),4326),'%s')",
8       objects[count,3],objects[count,4],
9       objects[count,3], objects[count,4],fits)
10    if(count > 1) { aux <- sprintf(",%s",aux) }
11    query <- sprintf("%s %s",query,aux)
12    count <- count+1
13  }
14  query <- sprintf("INSERT INTO Objects(ra,dec,"+
15    "centroid, fitsImage) VALUES %s", query)
16  print(query)
17  dbSendQuery(database, query)
18 }
```

Función 5: Función que inserta los objetos en la tabla de la base de datos.

```

1 mainFunction <- function(database, fits){
2   ptm <- proc.time()
3   im <- readFITS(fits)
4   xy <- getObjects(im)
5   coordData <- getCoordData(im$hdr)
6   absCoord <- calculateCoords(coordData, xy[[1]],xy[[2]])
7   writeToDataBase(database, absCoord, fits)
8   print("Process finished!")
9   print(sprintf("Imagen Fits: %s",fits))
10  print(sprintf("%s Objects Identified and archived.",
11    (length(xy[[1]])/2)))
12  proc.time() - ptm
13 }
```

Función 6: Función principal que realiza todos los cálculos antes descritos.

```

1 drv <- dbDriver("PostgreSQL")
2 con <- dbConnect(drv, dbname="Memoria","XXX.XXX.XXX.XXX"
3   , user="postgres", password="admin", port="5432")
4 dbSendQuery( conn = con, "CREATE TABLE Objects
```



```

5 (id serial primary key, centroid geometry,ra real,
6   dec real, fitsImage varchar(80))")
7 dbSendQuery(conn=con, "CREATE INDEX objects_index ON
8   Objects USING gist (centroid)")

```

Función 7: Configuración de Base de datos.

```

1 selectSectionRectangleIndex <- function(minRA,maxRA
2   ,minDEC,maxDEC){
3   ptm <- proc.time()
4   query <- sprintf("SELECT id FROM Objects WHERE
5     centroid && ST_MakeBox2D(ST_Point(%s,%s),
6       ST_Point(%s,%s))", minRA,minDEC,maxRA,maxDEC);
7   res <- dbSendQuery(conn=con, query)
8   data <- fetch(res,n=-1)
9   proc.time() - ptm
10  return(data)
11 }
12 selectSectionCircleIndex <- function(ra,dec,radius){
13   query <- sprintf("SELECT id FROM Objects WHERE
14     centroid && ST_Buffer(ST_Point(%s,%s), %s)",
15     ra, dec, radius);
16   res <- dbSendQuery(conn=con, query)
17   data <- fetch(res,n=-1)
18   return(data)
19 }
20
21 selectSectionNearIndex <- function(ra,dec){
22   query <- sprintf("SELECT id FROM Objects WHERE
23     centroid && ST_Buffer(ST_Point(%s,%s), 0.5)",
24     ra, dec);
25   res <- dbSendQuery(conn=con, query)
26   data <- fetch(res,n=-1)
27   return(data)
28 }

```

Función 8: Consultas espaciales.

1.2. Calibración

Anexo de la investigación previa realizada sobre la calibración de las antenas para lograr una óptima observación, estos pasos son previos a lo desarrollado en este proyecto dado que la imagen que se analiza ya viene calibrada y lista.

1.2.1. La Atmósfera

Los telescopios ubicados en la tierra sufren la atenuación de la señal recibida al pasar por la atmósfera terrestre. Además de esta disminución en la señal, las emisiones atmosféricas generan la refracción de ésta lo que produce cambios en el recorrido de la señal. Para una correcta interpretación de las medidas obtenidas es necesario tener en cuenta estos cambios, por lo que deben conocerse las propiedades de propagación así como también la radiación de la atmósfera.

Las ondas de radio recibidas se verán afectadas por la atmósfera dependiendo las frecuencias de onda que estas posean. A grandes frecuencias las ondas casi no perciben atenuación, pero a medida que se acercan al rango crítico que se encuentra regularmente bajo los 10 MHz se perciben distorsiones por los electrones presentes en la ionósfera. Este rango crítico puede variar dependiendo la densidad de electrones presentes. Como se puede apreciar muchas medidas radio astronómicas se realizan por sobre este rango, por lo que no perciben este efecto sin embargo a bajas frecuencias de medición este efecto puede ser muy importante. El área de la geofísica tiene un especial enfoque en esta región de observación por lo que se compensan estos efectos mediante el uso de resolución gran angular e imágenes de baja frecuencia.

Para las longitudes de onda que varían entre cm, mm y con mayor influencia aún las que están bajo los mm, la absorción realizada por la tropósfera se debe tomar en cuenta.

Nubes de agua pueden absorber y dispersar ondas de radio incluso a frecuencias muy bajas como 6 GHz. Una larga nube de lluvia por ejemplo puede generar una atenuación de hasta 1.5 dB, donde el promedio de atenuación de un cielo despejado es de 0.2 dB. A medida que la frecuencia aumenta, la absorción astronómica también lo hace.

La atmósfera seca por sobre los 80 km posee una mezcla mayoritariamente de Nitrógeno (78,09 %), Oxígeno (20,95 %) y Argón (0,93 %). Esta mezcla es casi constante a menores alturas pero puede variar un poco la concentración de cada uno dependiendo la altura y el tiempo. La molécula más importante a considerar es el vapor de agua, su ratio de mezcla varía dependiendo el clima y la altura. Otra molécula que ha incrementado los últimos años es el Dióxido de Carbono que también varía mucho dependiendo el clima. Otra molécula a considerar es el Ozono, que a diferencia del Dióxido de Carbono su concentración ha ido disminuyendo los últimos años, esta ubicado a una altura entre 20 y 30 Km. Ozono varía mucho dependiendo las estaciones del año y la ubicación geográfica. Dado que la capa de Ozono es responsable de la absorción de los rayos UV provenientes del sol, el que su concentración haya disminuído tiene una importancia práctica para los telescopios. Las líneas espectrales del Ozono están presente en frecuencias de 67.36 GHz hacia arriba. Las líneas de emisión en el *zenith*¹ alcanzan una temperatura aproximada de 60 grados K para el aire seco, por otra parte con una cantidad aproximada de $2\frac{g}{cm^2}$ la temperatura se mantiene bajo los 5-6 grados K.

La presión atmosférica decrece a una razón cercana a la exponencial con la altitud h

$$P(h) = P_0 e^{\frac{-h}{H}}. \quad (1.1)$$

Para determinar H que corresponde a la altura de la escala se utiliza la siguiente ecuación que tiene un valor aproximado de

$$H = \frac{R T}{u g} \approx 7998m \quad (1.2)$$

donde u es la masa molecular del aire, R la constante del gas, g la aceleración de gravedad y T la temperatura del gas. El gas en estudio en este caso es el aire.

La cantidad de agua precipitable medida en mm a una cierta altura h_0 es una

¹ Posición de observación ubicada exactamente sobre el observador.

integral sobre la línea de vista. Por lo general la cantidad de H_2O es determinada mediante medidas de radio a 225 GHz de frecuencia combinada con modelos de la atmósfera. Dependiendo la ubicación del telescopio, medidas a 183 GHz de las líneas espectrales del vapor de agua pueden ser usadas para estimar la cantidad total de H_2O en la atmósfera. Para sitios ubicados a nivel del mar se utilizan frecuencias de 22.235 GHz para estimar la cantidad. La escala de altura del agua $H_{H_2O} \approx 2km$, es considerablemente menor que la escala de altura del aire seco $H_{air} \approx 8km$. Por este motivo los radio telescopios son usualmente ubicados por sobre los 3.000 m de altura. El telescopio ALMA se encuentra ubicado a 5.000 m de altura y tiene su centro de operaciones a 3.000 m de altura.

En la *Figura 1.1* se muestra un modelo utilizado para predecir la atenuación generada por el H_2O , el O_2 y otros elementos presentes en la atmósfera sobre las medidas obtenidas. Con este ejemplo se puede ver la influencia de la atmósfera sobre las distintas longitudes de onda obtenidas. En la *Figura 1.1* se muestra en la parte superior la transmisión total de la data, seguida de la transmisión de la data cuando hay solo O_2 presente y en la parte de abajo se muestra la transmisión de data cuando hay solo vapor de H_2O presente. Como se puede observar en la *Figura 1.1* la transmisión de la data a mayores frecuencias sufre una mayor atenuación por lo que lógicamente la gran mayoría de estudios se encuentra enfocado en estas frecuencias. Observaciones por sobre estas frecuencias, donde la transmisión de la información se hace más complejo es posible lograrlo mediante la ayuda de nuevos modelos que se encuentran en desarrollo.

Si se considera que la consistencia física de la atmósfera no depende de su posición en un vecindario aproximado de 100 km de diámetro, sino que todas las variaciones sobre la atmósfera como la presión, densidad y mezcla de gases dependen de la altura se torna más fácil el estudio de atenuación sobre las medidas.

La atmósfera también produce cierta refracción sobre los rayos obtenidos generando curvaturas sobre estos, un factor no menor a la hora de calibrar.

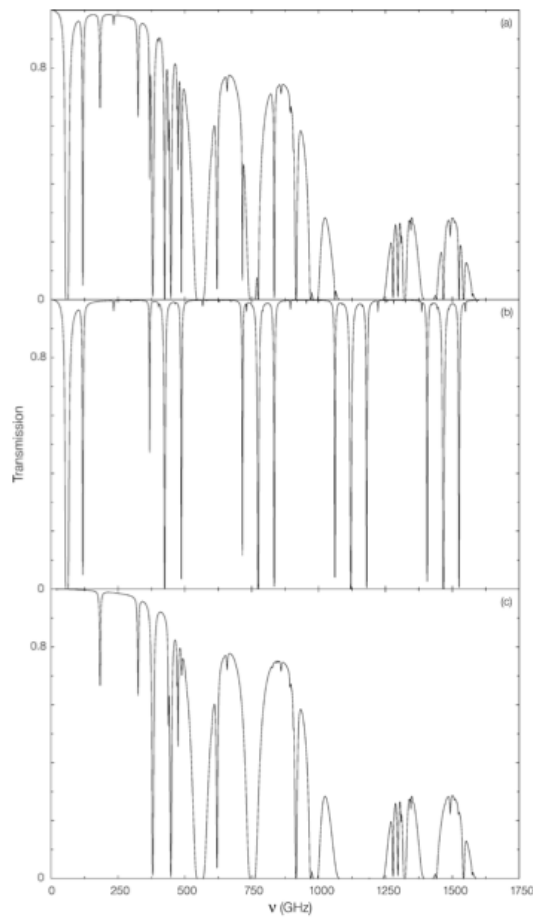


Figura 1.1: Transmisión de la data, con diferentes elementos presentes.

Por lo general el comportamiento de la atmósfera no varía notoriamente a lo largo de las horas, lo cuál facilita la estimación de la atmósfera pudiendo obtener parámetros de calibración en varios interválos para obtener una calibración óptima.

1.2.2. Parámetros de la Antena

Por lo general los sistemas de antenas son analizadas usando detallados modelos numéricos como GRASP¹, especialmente cuando se trata de antenas de alta ganancia como las utilizadas en la radio astronomía. Dentro de esta sección se darán a conocer las características de las antenas con el fin de luego poder entender su calibración.

¹Modelo numérico utilizado para optimización combinatoria.

El patrón de poder Por lo general el patrón de poder normalizado se conoce como:

$$P_n(\vartheta, \varphi) = \frac{1}{P_{max}} P(\vartheta, \varphi) \quad (1.3)$$

El teorema de reciprocidad provee un método para medir esta cantidad. La fuente de radiación puede ser reemplazada por una fuente de radio pequeña. La densidad de flujo de esas fuentes es determinada mediante las medidas obtenidas por la antena, en longitudes de onda de centímetros y milímetros. A longitudes de onda cortas se usan planetas o lunas de planetas, donde la temperatura de la superficie es captada por infrarojo.

El patrón de poder también puede ser medido mediante transmisores artificiales, donde se mandan señales de una antena a otra. Con este método hay que tener en consideración que la separación de las antenas al ser tan larga y donde las fuentes son tan puntuales y pequeñas (comparadas con el tamaño de un planeta) la radiación que emite la antena A sobre la B puede ser considerada pequeña y lejana. Para esto se requiere que la curvatura del frente de onda emitido por la antena B es mucho menor que una longitud de onda de A, en dimensiones geométricas. Por lo que debe cumplir con $k \ll 2D^2/\lambda$, donde λ es la longitud de onda y D es el diámetro de la antena.

Considerando el patrón de poder de una antena usada como transmisor. Si el poder total espectral, P_v en $[WHz^{-1}]$ es alimentado con una antena isotrópica sin pérdida, esto transmitirá P unidades de poder por ángulo sólido¹ por Hertz. Entonces el poder total de radiación a una frecuencia ν es $4\pi P_\nu$. En términos reales y con una antena sin pérdida el poder $P(\vartheta, \varphi)$ por unidad de ángulo sólido es radiado en la dirección (ϑ, φ) . Si se define la ganancia directiva $G(\vartheta, \varphi)$ como en la ecuación 1.4

¹ Ángulo espacial que abarca un objeto visto desde un punto de vista dado, que se corresponde con la zona del espacio limitada por una superficie cónica.

o en la 1.5 :

$$P(\vartheta, \varphi) = G(\vartheta, \varphi)P \quad (1.4)$$

$$G(\vartheta, \varphi) = \frac{4\pi P(\vartheta, \varphi)}{\int \int P(\vartheta, \varphi) d\Omega} \quad (1.5)$$

Se tiene que la ganancia o directividad es también un patrón de poder normalizado similar al de la ecuación 1.3, pero con la diferencia que el factor normalizador es $\int P(\vartheta, \varphi) d\Omega / 4\pi$. Esta es la ganancia relativa a una fuente isotrópica sin pérdida. Dado que esta fuente no puede darse en la realidad una cantidad medible es la ganancia relativa hacia alguna antena estándar como una antena dipolo de media onda que su directividad es usada para consideraciones teóricas.

Haz principal de ángulo sólido El haz principal de ángulo sólido conocido como Ω_A de una antena, está dado por:

$$\Omega_A = \int \int_{4\pi} P_n(\vartheta, \varphi) d\Omega = \int_0^{2\pi} \int_0^\pi P_n(\vartheta, \varphi) \sin \vartheta d\vartheta d\varphi \quad (1.6)$$

La unidad de medida del haz principal es de *steradians*² o *sr*, la integración de datos se hace a lo largo de toda la esfera (4π) y al igual que el cálculo del poder la ecuación 1.3 también se expresa en términos ideales donde el tipo de antena a modelar no existe, dado que su comportamiento debiera ser de $P_n = 1$ en todos los Ω_A , de lo contrario $P_n = 0$. La mayoría de las antenas tienen el mismo comportamiento, donde el patrón de poder normalizado tiene un valor considerablemente mayor a un cierto rango de ϑ y φ que para el resto de la esfera. Este rango se denomina haz principal o lóbulo principal de la antena y lo demás se denomina lóbulos laterales o traseros. En la *Figura 1.2* se muestra este comportamiento, donde se observa la gran diferencia que hay entre el poder en el haz principal y a lo largo del resto de la esfera.

En situaciones actuales las propiedades parecen bien definidas hasta en las lon-

²Unidad de medida utilizada para cuantificar ángulos en un plano.

gitudes de onda más cortas. En las longitudes de onda más cortas existe el haz principal pero mucho del poder entra también por los lados de este haz. Es decir la eficiencia del haz principal puede variar notoriamente con el aumento de la altura, por lo que calibrar el radio telescopio a muy cortas longitudes de onda puede ser muy tedioso.

Análogamente a la ecuación 1.6 se define la medida del haz principal de luz (sólo el haz principal a diferencia de Ω_A que abarca los lóbulos laterales también) se conoce como Ω_{MB} .

$$\Omega_{MB} = \int_{mainlobe} \int P_n(\vartheta, \varphi) d\Omega \quad (1.7)$$

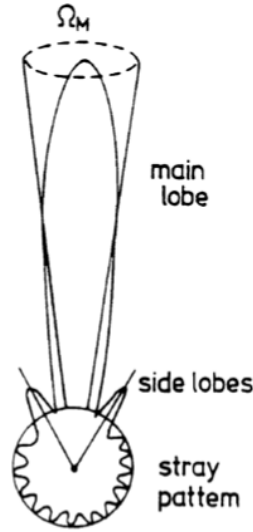


Figura 1.2: Haz principal con sus lóbulos laterales lejanos y cercanos.