

UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
Departamento de Ingeniería Informática



**Reconstrucción de imágenes de interferometría mediante algoritmos
iterativos paralelos en múltiples GPU**

Miguel Ángel Cárcamo Vásquez

Profesor guía: Fernando R. Rannou Fuentes

Profesor co-guía: Pablo E. Román Asenjo

Tesis de grado presentada en
conformidad a los requisitos
para obtener el grado de Magíster
en Ingeniería Informática

Santiago -- Chile

2016

© **Miguel Ángel Cárcamo Vásquez** - 2016



• Algunos derechos reservados. Esta obra está bajo una Licencia Creative Commons Atribución-Chile 3.0. Sus condiciones de uso pueden ser revisadas en:
<http://creativecommons.org/licenses/by/3.0/cl/>.

Dedicado a...

AGRADECIMIENTOS

Agradezco a

RESUMEN

Lorem ipsum dolor sit cuchufli barquillo bacán jote gamba listellor po cahuín, luca melón con vino pichanga coscacho ni ahí peinar la muñeca chuchada al chanco achoclonar. Chorrocientos pituto ubicatex huevo duro bolsero cachureo el hoyo del queque en cana huevón el año del loly hacerla corta impeque de miedo quilterry la raja longi ñecla. Hilo curado rayuela carrete quina guagua lorea piola ni ahí.

Palabras Claves: Palabras; Claves

ABSTRACT

Contrary to popular belief, Lorem Ipsum is not simply random text. It has roots in a piece of classical Latin literature from 45 BC, making it over 2000 years old. Richard McClintock, a Latin professor at Hampden-Sydney College in Virginia, looked up one of the more obscure Latin words, consectetur, from a Lorem Ipsum passage, and going through the cites of the word in classical literature, discovered the undoubtable source. Lorem Ipsum comes from sections 1.10.32 and 1.10.33 of "de Finibus Bonorum et Malorum" (The Extremes of Good and Evil) by Cicero, written in 45 BC. This book is a treatise on the theory of ethics, very popular during the Renaissance. The first line of Lorem Ipsum, "Lorem ipsum dolor sit amet..", comes from a line in section 1.10.32.

The standard chunk of Lorem Ipsum used since the 1500s is reproduced below for those interested. Sections 1.10.32 and 1.10.33 from "de Finibus Bonorum et Malorum" by Cicero are also reproduced in their exact original form, accompanied by English versions from the 1914 translation by H. Rackham.

Keywords: Key; words

TABLA DE CONTENIDO

1	Introducción	1
1.1	Antecedentes y motivación	1
1.2	Descripción del problema	2
1.2.1	Estado del arte	2
1.2.2	Definición del problema	4
1.3	Solución propuesta	5
1.3.1	Características de la solución	5
1.3.2	Propósitos de la solución	6
1.4	Objetivos y alcance del proyecto	7
1.4.1	Objetivo general	7
1.4.2	Objetivos específicos	7
1.4.3	Alcances	7
1.5	Metodología y herramientas utilizadas	8
1.5.1	Metodología	8
1.5.2	Herramientas de desarrollo	8
1.5.3	Ambiente de desarrollo	9
1.6	Organización del documento	10
2	Síntesis de imágenes de radiointerferometría	11
2.1	Atacama Large Millimeter-Submillimeter Array	11
2.2	Principios y conceptos de interferometría	20
2.3	Métodos existentes de deconvolución	21
2.3.1	CLEAN	21
2.3.2	MEM Mono-frecuencia	22
2.3.3	MEM Multi-frecuencia	24
3	Optimización	27
3.1	Métodos de gradiente conjugado en múltiples dimensiones	27
4	Multi-GPU	29
4.1	Comunicación Peer-to-Peer	29
4.2	Direccionamiento Virtual Unificado	30
4.3	OpenMP	31
5	Pruebas	33
5.1	Benchmarks	33
5.2	Caracterización de MEM	33
6	Resultados	35
7	Conclusiones	36
	Referencias bibliográficas	39
	Anexos	39
A	Cálculo del gradiente de la función objetivo	40

ÍNDICE DE TABLAS

Tabla 1.1	Datos HLTau	4
Tabla 2.1	Las 10 Bandas de frecuencia de ALMA	18
Tabla 4.1	Topología de Belka	29

ÍNDICE DE ILUSTRACIONES

Figura 2.1	Distribución de antenas para el muestro del objeto HLTauri en banda 6. Fuente: Elaboración propia.	12
Figura 2.2	Centro de la distribución de antenas para el muestro del objeto HLTauri en banda 6. Fuente: Elaboración propia.	12
Figura 2.3	Sistema de coordenadas (X, Y, Z) y la dirección de los ejes en términos de ángulo horario H y declinación δ . Fuente: (A. Richard Thompson, 2004)	13
Figura 2.4	Relación entre los sistemas de coordenadas (X, Y, Z) y (u, v, w) . El sistema (u, v, w) está definido por la observación en la dirección del punto S , el cual tiene un ángulo horario H y una declinación δ . Como S está en el la mitad del hemisferio este, H es negativo (sentido anti-horario). Fuente: (A. Richard Thompson, 2004) .	14
Figura 2.5	Relación entre coordenadas horizontales y coordenadas ecuatoriales horarias. Fuente: (A. Richard Thompson, 2004)	15
Figura 2.6	Centro y eje semimayor de la elipse (Ecuación 2.6). Fuente: (A. Richard Thompson, 2004)	16
Figura 2.7	Elipses muestreadas con diferentes parámetros en el plano (u, v) . Fuente: (A. Richard Thompson, 2004)	17
Figura 2.8	Plano (u, v) del objeto HLTauri (HLTau) en diferentes bandas. Fuente: Elaboración propia	19
Figura 2.9	Bandas, <i>spectral windows</i> y canales del objeto HL-Tauri. Fuente: Elaboración propia	19
Figura 2.10	Dos antenas y su respectivo <i>baseline</i> . \vec{x} es la posición del objeto estudiado, y por otro lado \vec{k} es el <i>baseline</i> correspondiente a las dos antenas. Fuente: Elaboración propia.	21
Figura 2.11	Ilustración del cálculo de χ^2 y $\nabla\chi^2$ con conjuntos de datos multi-frecuencia. Fuente: Elaboración propia	25
Figura 4.1	Ilustración de la topología de Belka. Fuente: Elaboración propia	30
Figura 4.2	Uso de OpenMP en múltiples GPUs. Fuente: Elaboración propia	31

ÍNDICE DE ALGORITMOS

Algoritmo 2.1	Algoritmo CLEAN	22
Algoritmo 2.2	Algoritmo del kernel 1D que calcula el vector χ^2	25
Algoritmo 2.3	Algoritmo del kernel 2D que calcula el vector S	25
Algoritmo 2.4	Algoritmo del kernel 2D que calcula $\nabla\chi^2$	26
Algoritmo 2.5	Algoritmo del kernel 2D que calcula el vector ∇S	26
Algoritmo 3.1	Algoritmo de Fletcher-Reeves/Polak-Ribiere	27
Algoritmo 4.1	Cálculo de χ^2 en paralelo	32
Algoritmo 4.2	Cálculo de $\nabla\chi^2$ en paralelo	32

CAPÍTULO 1. INTRODUCCIÓN

1.1 ANTECEDENTES Y MOTIVACIÓN

El *Atacama Large Millimeter/Submillimeter Array* es un instrumento revolucionario en su concepto científico, su diseño de ingeniería y su organización como un esfuerzo científico global. Construido en el desierto de Atacama de Chile septentrional a una altitud de 5.000 metros sobre el nivel del mar en el Llano Chajnantor, ALMA cuenta con 66 antenas de alta precisión funcionando juntas en longitudes de onda milimétricas y submilimétricas. Gracias a su alta resolución y sensibilidad, ALMA abre una “ventana” totalmente nueva al Universo, permitiendo a los científicos desenmarañar misterios astronómicos importantes y de muchos años, explorando nuestros orígenes cósmicos (Peck & Beasley, 2008a).

ALMA captura señales del cielo con dos o más antenas combinadas con el fin de analizar la señal y así obtener información acerca de la fuente que emite dicha señal (ya sea una estrella, planeta, galaxia u otro objeto astronómico). De esta forma, combinando las ondas de radio recolectadas por muchas antenas es posible construir una imagen. Estas imágenes son comparables con un hipotético telescopio gigante o una antena de 14.000 metros de diámetro. Como la construcción y operación de una antena de tal tamaño es técnicamente imposible, lo factible es construir muchas antenas pequeñas y combinar su salida (ALMA, 2009).

Sin embargo, este problema no es tan simple como parece, puesto que para la reconstrucción de las imágenes se necesitan algoritmos especializados. Actualmente existen dos grandes tipos de algoritmos: aquellos basados en transformaciones lineales y aquellos con un soporte Bayesiano, no lineal.

Estos últimos tienen una base matemática sólida, pero fueron dejados de lado por su pobre rendimiento computacional. Es por ello que gracias al desarrollo del hardware, especialmente de las GPU, se pretende acelerar estos algoritmos para tener imágenes de mejor calidad en un tiempo menor al que se obtiene con soluciones secuenciales.

1.2 DESCRIPCIÓN DEL PROBLEMA

1.2.1 Estado del arte

La reconstrucción de imágenes usando radiointerferómetros que captan una señal del cielo a medida que rota la tierra es llamado un problema *ill-posed* (Chen, 2011), debido a la irregularidad de los muestreos Nyquist en el dominio de Fourier. Estos muestreos resultan en *aliasing* o sobreposición de ondas en la imagen debido a los altos lóbulos laterales en la respuesta del *array* de antenas. Es importante entender que obtener imágenes de alta calidad depende tanto de contar con los mejores equipos como con las mejores técnicas de reconstrucción. En ese sentido, cada generación de radiotelescopios involucra un esfuerzo en el desarrollo de *hardware*. Sin embargo, para explotar las capacidades de ese *hardware* se requiere una constante mejora en los enfoques de reconstrucción de imágenes para poder igualar la sensibilidad del receptor.

Durante los últimos 40 años, múltiples técnicas de deconvolución han sido desarrolladas para resolver el problema mencionado anteriormente.

La idea básica detrás de un algoritmo de deconvolución es explotar un conocimiento *a-priori* de la imagen. El primer algoritmo, y la más popular de estas técnicas en el campo de la astronomía, es el método CLEAN propuesto por Högbom (Cornwell, 2009). Este método se basa en aproximar la imagen usando lo que se denomina *dirty image*. La *dirty image* se muestra en la ecuación 2.10 y es la imagen obtenida luego de aplicar la transformada inversa de Fourier sobre el conjunto de visibilidades V multiplicada por la función de muestreo $S(u, v)$, tal que $S(u, v) = 1$ donde exista una visibilidad y $S(u, v) = 0$ si no existe. Las coordenadas (u, v) representan las coordenadas de las mediciones en el plano de Fourier esparcidos de forma irregular. Por otra parte los puntos (x, y) representan las coordenadas planas en la imagen escalada.

Por lo tanto, CLEAN asume que la imagen es una colección de fuentes puntuales e iterativamente remueve fuentes puntuales para crear una imagen de residuos.

Sin embargo, este método posee tanto deficiencias técnicas como matemáticas. Esto debido a que CLEAN no ofrece una alternativa de regularización del problema y tampoco una interpretación estadística del resultado que entrega (Taylor et al., 1999). Además, CLEAN no tiene una convergencia definida por lo que debe iterarse según la apreciación que el usuario tiene de la imagen. Además, el problema de la reconstrucción de una imagen consiste en seleccionar una solución de muchas posibles y CLEAN sólo selecciona una probable imagen de un conjunto de soluciones factibles.

Otra técnica de deconvolución importante es el método de máxima entropía (MEM), que consiste en seleccionar una imagen que se ajuste a las visibilidades observadas dentro de un

nivel de ruido y cuyos píxeles satisfagan la restricción de máxima entropía.

Las primeras presentaciones de MEM fueron hechas por (Frieden, 1972; Ables, 1974; Wernecke & D'Addario, 1977; Gull & Daniell, 1978; Skilling & Bryan, 1984) en donde la técnica fue modificada a lo largo de los años de tal forma de probar diversas funciones y formas de entropía, además de distintas técnicas de maximización. Por ejemplo, (Frieden, 1972) y (Gull & Daniell, 1978) consideran distribuciones de intensidad dispersas aleatoriamente en el cielo. Es por ello que la entropía que se utiliza en estos trabajos es de la forma $\sum_i I_i \log I_i$, donde $I_i = I(x_i, y_i)$, en donde además (Frieden, 1972) usa el método de Newton-Raphson para llevar a cabo la maximización del problema, en donde queda claro que es sólo exitoso en problemas pequeños. Por otra parte, (Ables, 1974) y (Wernecke & D'Addario, 1977) utilizan una entropía de la forma $\sum_i \log I_i$, este último además usa el método del gradiente conjugado. En (Skilling & Bryan, 1984) se resume el método y se nombran distintos tipos de algoritmos de optimización, concluyendo con un método que propone deshacerse del uso del penalizador explícito λ construyendo así un enfoque general del método. El modelo de (Cornwell & Evans, 1985) realiza aproximaciones diagonalizando una matriz Hessiana para optimizar la función objetivo, en donde se utiliza una entropía de la forma $\sum_i I_i \log \frac{I_i}{M_i}$ en donde M es un modelo dado y usualmente una imagen plana. En (Skilling & Gull, 1991) se utiliza una entropía de la forma $\sum_i I_i - M_i - I_i \log \frac{I_i}{M_i}$ en donde M cumple la misma función que en el trabajo de (Cornwell & Evans, 1985). Finalmente en (Casassus et al., 2006) se implementa el algoritmo MEM basado en el método del gradiente conjugado, sin el uso de la aproximación de (Cornwell & Evans, 1985), usando una entropía de la forma $\sum_i I_i \log \frac{I_i}{M}$ en donde M representa el menor valor de intensidad de un píxel.

Más allá de estas dos técnicas, existen otros algoritmos que no son tan famosos en las astronomía. Por ejemplo, han sido desarrollada una técnica de mínimos cuadrados globales no-negativos *Global nonnegative least squares* (LS) propuesta por Briggs (Briggs, 1995). Esto es una técnica basada en matrices paramétricas como la de mínima varianza (*LS minimum variance imaging*, LS-MVI) (Leshem & Jan Van Der Veen, 2000), y por último una técnica de máxima verosimilitud *Maximum likelihood* (Ben-David & Leshem, 2008).

Por otro lado, las tarjetas gráficas, o GPU, son *hardware* que durante los últimos años han pasado de ser de propósito particular, es decir, sólo útil para juegos, películas 3D e interfaces gráficas, a ser de propósito general (GPGPU). Esto gracias a que en los últimos años se ha visto una explosión en el interés científico por aprovechar las capacidades de estos *chips* para llevar a cabo aplicaciones y cálculos de propósito general. Así es como las GPGPU han sido utilizadas para llevar a cabo simulaciones, aplicaciones de visión por computador, procesamiento y análisis de señales, procesamiento de imágenes y minería de datos (Owens et al., 2007).

Recientemente, se propuso un enfoque que busca realizar inferencias Bayesianas para observaciones radioastronómicas (BIRO) (Lochner et al., 2015). Este enfoque lleva a cabo

una inferencia Bayesiana para muestrear un conjunto de parámetros representando un modelo del cielo para proponer las visibilidades que mejor se ajusten a éste. Este trabajo parametriza la imagen en un conjunto reducido de parámetros, y no reconstruye la imagen en sí, sino un modelo de imagen, restringido a puntos y funciones Gaussianas. Además se ha propuesto Montblanc (Perkins et al., 2015), una implementación en GPU en apoyo a BIRO y que permite paralelizar la proposición de visibilidades.

1.2.2 Definición del problema

El radio observatorio ALMA (Peck & Beasley, 2008b) entrega datos observacionales en el espectro de radio frecuencias con una resolución nunca antes lograda. Para una observación dada, este instrumento recoge un subconjunto discreto y disperso (no regular) de la transformada de Fourier de la imagen. La cantidad de muestras (visibilidades) depende del objeto que se esté observando y del tiempo de observación. La Tabla 1.1 muestra el número de visibilidades observadas de el set de datos HLTau muestrado en las bandas de frecuencia 3, 6 y 7 ; y su respectivo valor en bytes. Este último número es calculado multiplicando el número de visibilidades observadas por la suma bytes de las variables utilizadas así como las coordenadas de las visibilidades en el plano de Fourier, éstas son números complejos por lo que contienen una parte real e imaginaria, además de la varianza asociada al muestreo. Cada una de estas variables es considerada como tipo *float* por lo cual cada una tiene un valor de 4 *bytes*. Sin embargo, el problema de reconstruir la imagen no pasa simplemente por usar la transformada inversa del conjunto de puntos. Pues como se dijo anteriormente, los puntos recolectados no coinciden con una grilla regular. Este problema es conocido en matemáticas como “*Fourier Synthesis*” (Maréchal & Wallach, 2009) y sorprendentemente aún se encuentra como una floreciente línea de investigación en el campo del análisis armónico (Roman, 2012).

Tabla 1.1: Datos HLTau

Set de datos	Número de visibilidades	Bytes
Band 3	-	-
Band 6	3.167.024	63.340.480
Band 7	-	-

Son variados los métodos que permiten reconstruir una imagen a partir de muestras dispersas y no regulares en el espacio Fourier. Entre ellos se encuentran los conocidos métodos CLEAN (Cornwell, 2009) y MEM (Narayan & Nityananda, 1986), en donde el primero es un algoritmo basado en transformaciones lineales y el segundo basado en la teoría Bayesiana.

Ambos son conocidos particularmente en el campo de la astronomía.

El algoritmo CLEAN provee una solución basada en representar la imagen mediante un conjunto de fuentes puntuales. La imagen final, mejor conocida como la imagen CLEAN, es la suma de todos los componentes de los puntos convolucionados con el *CLEAN beam*, esto es, la imagen de una función Gaussiana, con el fin de restar importancia a las frecuencias espaciales más altas que suelen ser falsamente extrapoladas (Taylor et al., 1999). Finalmente, CLEAN es un procedimiento cuyo objetivo es generar una imagen de residuos con variaciones homogéneas y similares al ruido térmico presente en los datos.

Como se menciona en la sección 1.2.1, CLEAN posee deficiencias técnicas y matemáticas, a pesar de su éxito y su uso actual en ALMA.

Sin embargo, este método sigue siendo utilizado como una herramienta diaria por la comunidad astronómica y es preferido de acuerdo a su eficiencia computacional, pese a que los algoritmos Bayesianos tales como MEM, generan mejores imágenes y gozan de mayor sustento estadístico. Sin embargo, estos últimos no se utilizan porque se desconocen sus propiedades estadísticas, debido a que no existe una estrategia de paralelización que provea soluciones de forma rápida y que permita estudiar el rendimiento a nivel de calidad de imágenes del algoritmo.

En ese sentido la pregunta de investigación que se desea responder es: ¿El algoritmo MEM produce imágenes de mejor calidad que CLEAN?

También es importante mencionar que MEM tiene un enfoque Bayesiano que tiene como resultado una función no lineal. Para minimizar este tipo de funciones existen distintos tipos de métodos de optimización, como por ejemplo, el método de gradiente conjugado (Hager & Zhang, 2006). Por lo tanto, el desafío técnico de este trabajo es diseñar una estrategia de paralelización en una arquitectura SIMT para este problema.

1.3 SOLUCIÓN PROPUESTA

1.3.1 Características de la solución

Se pretende llevar a cabo el desarrollo de un algoritmo de reconstrucción basado en el principio de máxima entropía, aplicando una estrategia de paralelización en múltiples GPU y usando la plataforma de computación paralela CUDA, para luego caracterizar el algoritmo a nivel de calidad de imagen.

El método de máxima entropía (MEM) encuentra la imagen que mejor se ajusta a los datos, dentro de un nivel de ruido y que maximiza la entropía S . Esto es llevado a cabo

minimizando la función:

$$\Phi = \chi^2 + \lambda S \quad (1.1)$$

Donde en el caso de que los datos provengan de un interferómetro, se tiene que:

$$\chi^2 = \sum_{j=1}^Z \frac{|V_j^m - V_j^o|^2}{\sigma_j^2} \quad (1.2)$$

Donde la sumatoria recorre todas las visibilidades y el numerador de la fracción representa la diferencia entre las visibilidades modelo, creadas luego de realizar una transformada de Fourier a la imagen, llevar las visibilidades observadas a una grilla regular y luego realizar una interpolación bilineal; y las visibilidades observadas. Así, esta diferencia es llamada visibilidad residuo y se debe calcular el módulo del complejo para luego elevarlo al cuadrado. Por otra parte, σ_k es la varianza de cada visibilidad en función del ruido termal de las antenas.

La entropía S es un término regularizador en un problema inverso, donde se tienen más parámetros libres que datos. Si bien existen distintas formulaciones para S , la solución a presentar usa una entropía de forma $S = \sum_k^n I_k \log \frac{I_k}{G}$ donde $\{I_k\}_{k=1}^n$ es el valor de intensidad de la imagen en el píxel i y G es el menor valor positivo de intensidad que puede tomar ésta. Además, λ actúa como un penalizador y tiene un rol similar a los multiplicadores de Lagrange.

Es importante destacar que la Ecuación 1.1 es no lineal y se debe minimizar con algoritmos apropiados agregando una restricción de positividad, como por ejemplo, el método de gradiente conjugado Polak-Ribiere (Polak & Ribiere, 1969).

Así, MEM produce una imagen positiva en un rango definido de píxeles haciendo que la imagen final sea suave y que tenga una super-resolución en brillos y objetos aislados, sin embargo, al ser un problema no lineal su costo en tiempo computacional es muy alto (Taylor et al., 1999), es por ello que una o muchas GPUs calcularán Φ y $\nabla\Phi$ para cada frecuencia, para luego hacer una reducción en una única GPU si es que se utilizan múltiples GPU. Para ello, se hace necesario usar estructuras de datos apropiadas para almacenar valores complejos e imágenes.

1.3.2 Propósitos de la solución

El propósito de la solución es caracterizar el algoritmo MEM a nivel de calidad de imagen con el fin de darle a conocer a la comunidad astronómica sus propiedades, de manera que se transforme en una alternativa real para obtener imágenes de gran resolución y poco ruido en un tiempo acotado y de manera práctica.

1.4 OBJETIVOS Y ALCANCE DEL PROYECTO

1.4.1 Objetivo general

El objetivo general del proyecto es diseñar y caracterizar un algoritmo de alto rendimiento basado en el método de máxima entropía (MEM), es decir, medir parámetros de rendimiento, resolución y nivel de ruido de las imágenes reconstruidas a partir de datos sintéticos para proveer a la comunidad astronómica una solución de reconstrucción de imágenes que sea de uso práctico y que además genere imágenes de mejor calidad que los algoritmos usados actualmente. .

1.4.2 Objetivos específicos

1. Formular matemáticamente el método MEM.
2. Diseñar una estrategia de paralelización en múltiples GPU para el método.
3. Llevar a cabo una implementación en múltiples GPU mediante la plataforma de computación paralela *CUDA*.
4. Evaluar el rendimiento computacional, comparando los tiempos de ejecución de la solución a implementar en multi-GPU y la versión multi-hebreada de MEM existente.
5. Evaluar la calidad de la imagen obtenida, comparando la imagen obtenida con la imagen CLEAN respectiva que usan los astrónomos actualmente.

1.4.3 Alcances

La solución adoptada es implementada sólo mediante la plataforma de computación paralela *CUDA* y el lenguaje de programación C. Además, solo se adopta una estrategia de paralelización para el algoritmo MEM y las pruebas de rendimiento computacional sólo son realizadas en un cluster de GPUs. Por otra parte, los resultados obtenidos serán comparados solamente con el método de reconstrucción de imágenes llamado CLEAN, esto debido a que es el método más utilizado por los astrónomos actualmente. Por último, el formato de la imagen de salida debe ser en formato FITS.

1.5 METODOLOGÍA Y HERRAMIENTAS UTILIZADAS

1.5.1 Metodología

El trabajo a realizar posee una etapa inicial de investigación, en la cual por medio de la literatura, la observación y la experimentación se debe estudiar el método MEM para luego buscar diseños de paralelización adecuados a éste. Bajo dicha perspectiva el método científico es útil para guiar la totalidad del trabajo. Sin embargo, como se menciona anteriormente, una vez que se haya obtenido el diseño, se debe desarrollar un software, por lo que se utilizarán reglas de la metodología *Extreme Programming*(XP) que ya han sido probadas en el ámbito de la investigación científica (Wood & Kleb, 2002). Las etapas de esta metodología son:

1. **Feedback:** Se toma en cada iteración un compromiso serio por la entrega de software que se trabaja. Se muestra el software temprano y con frecuencia luego se escucha con atención al cliente y hacen los cambios necesarios.
2. **Diseño incremental:** Se debe trabajar por un cierto tiempo, por ejemplo, dos semanas (iteración), y al final de éstas debe entregarse un software funcionando.
3. **Programación en parejas (*Pair programming*):** Los programadores trabajan en parejas. Todo el código es desarrollado por dos programadores (alumno y profesor guía en este caso).
4. **Reuniones diarias:** El propósito de esto es tener una reunión diaria para comunicar problemas, soluciones y promover la concentración del equipo.
5. **Pruebas:** Por cada funcionalidad nueva, se deben realizar pruebas unitarias, pruebas de aceptación, pruebas de humo, de stress y rendimiento.
6. **Integración continua:** Se deben automatizar las pruebas anteriormente mencionadas, para que cada vez que se agreguen nuevas funcionalidades se asegure que todas siguen funcionando.

1.5.2 Herramientas de desarrollo

A continuación se presentan las herramientas que se utilizarán para el desarrollo del proyecto.

Para el desarrollo de prototipos y pruebas de concepto se utiliza:

1. Sistema Operativo Ubuntu 14.04.
2. Matlab 2015a.

Para el desarrollo, compilación y ejecución de las implementaciones paralelas se utilizara:

1. Sistema Operativo Ubuntu 14.04.
2. CUDA 7.0.

1.5.3 Ambiente de desarrollo

En cuanto al ambiente de desarrollo, cabe destacar que existe un ambiente de pruebas y uno de producción.

Las pruebas de rendimiento y producción serán realizadas un cluster de GPUs con las siguientes características:

1. 2x 10 Core Intel Xeon E5-2640 V2 2GHz
2. 256 GB de RAM
3. 4 Tarjetas de video NVIDIA Tesla K80

Además son utilizados los recursos computacionales personales del estudiante, los cuales son:

1. Un computador de escritorio con:
 - 8 GB de RAM.
 - Procesador Intel i7-3770 3.40GHz x 8.
 - Tarjeta de video NVIDIA GeForce GTX 750 (1 GB).
2. Un computador portátil ASUS 450LN con:
 - 8 GB de RAM DDR3.
 - Procesador Intel i7-4500U (1800 MHz-3000 MHz)
 - Tarjeta de video híbrida Intel GMA HD Graphics 4400 & NVIDIA GeForce 840M (2 GB)

1.6 ORGANIZACIÓN DEL DOCUMENTO

falta organizacion del documento

CAPÍTULO 2. SÍNTESIS DE IMÁGENES DE RADIOINTERFEROMETRÍA

2.1 ATACAMA LARGE MILLIMETER-SUBMILLIMETER ARRAY

El Atacama Large Millimeter-Submillimeter Array (ALMA) es un radiointerferómetro compuesto de 66 antenas de alta precisión que operan a un ancho rango de frecuencias milimétricas y submilimétricas. Cincuenta de estas antenas miden 12 metros de diámetro y son utilizadas para síntesis de imágenes de alta resolución. Por otra parte, esto es complementado por el *Atacama Compact Array* también llamado *Morita Array* compuesto por doce antenas de 7 metros de diámetro y por otras 4 antenas de 12 metros diámetro que sirve para muestrear estructuraS de gran escala que no son bien muestreadas por las demás antenas de 12 metros. Además, todas las antenas se clasifican en tres tipos:

- **AEM (DA) 12m:** Estas antenas fueron ensambladas antes de llegar a Chile, por lo tanto, llegaron en pocas piezas. Estas antenas están manejadas por motores lineales, es decir, imanes que proveen el movimiento sin que exista un contacto entre las partes.
- **Vertex (DV) 12m:** La estructura que sostiene al platillo está hecha de plástico reforzado con fibra de carbón, lo cual es importante debido a las duras condiciones de 5.000 metros de altura.
- **MELCO (PM) 12m y 7m:** Estas antenas también poseen motores lineales además de una conducción en tres direcciones (altura, de lado a lado y de arriba a abajo), para seguir suavemente la fuente durante la observación.

El conjunto de antenas está ubicado en el llano Chajnantor a 5.000 metros sobre el nivel del mar, un sitio que ofrece un cielo claro y un aire seco, una condición ambiental requerida para observar ondas milimétricas y submilimétricas.

La distribución de las antenas cede paso a lo que es llamado *baseline* o distancia entre un par de antenas, que van desde 15 Km. hasta 16 Km. aproximadamente. Tanto la distribución como la distancia entre éstas es crucial en la determinación de la calidad de la imagen y la resolución de ALMA. Esto debido a que si el conjunto de antenas está distribuido de forma extendida dará como resultado una alta resolución espacial, en cambio, si la distribución es compacta, el resultado será una mejor sensibilidad para fuentes extendidas.

A continuación, la Figura 2.1 muestra la distribución de antenas para el muestreo del objeto HL Tauri en banda 6. Además la Figura 2.2 muestra el centro de la primera figura en donde

se puede visualizar una gran concentración de antenas. Además, se puede ver que cada antena posee un identificador (DA, DV o PM) dando a conocer los tipos de antena que están siendo utilizados para el muestreo. Es importante que cada antena tiene una coordenada X que apunta hacia al este geográfico y una coordenada Y que apunta al norte geográfico.



Figura 2.1: Distribución de antenas para el muestro del objeto HLTauri en banda 6. Fuente: Elaboración propia.

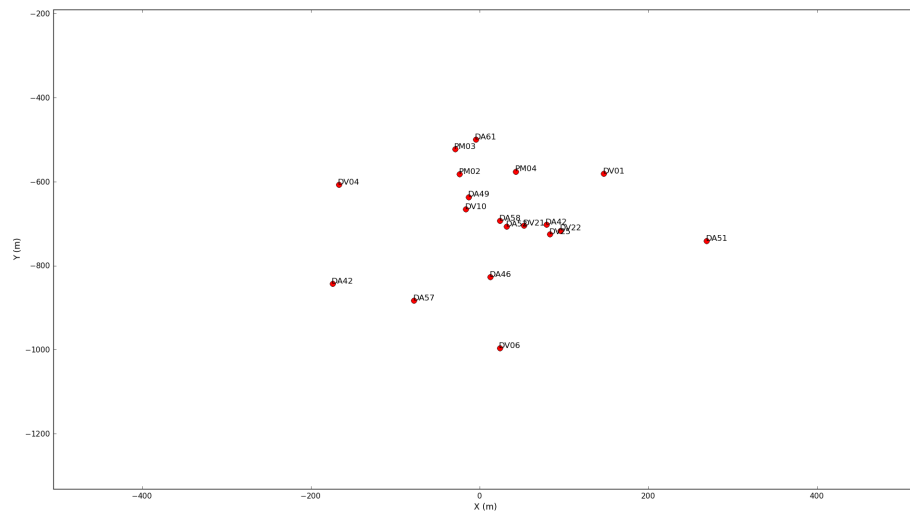


Figura 2.2: Centro de la distribución de antenas para el muestro del objeto HLTauri en banda 6. Fuente: Elaboración propia.

Varios sistemas de coordenadas son usados para especificar la posiciones de las antenas en un *array*. Uno de los sistemas usa coordenadas ecuatoriales horarias, es decir un ángulo horario H y la declinación δ . El ángulo H se mide entre cero y 360 grados o entre cero y 24 horas sidéreas. Por lo tanto, a cada hora le corresponden $360^\circ/24 = 15^\circ$. Además, se mide a partir de la intersección del meridiano local con el ecuador. Su extremo es la intersección del meridiano que pasa por el objeto a observar (S) con el ecuador.

Por otra parte, la declinación es el ángulo medido sobre el meridiano que para por la estrella u objeto a observar, entre el ecuador la dirección de éste, sus valores están comprendidos entre cero y $+90^\circ$ para los astros del hemisferio norte, y entre cero y -90° para los astros del hemisferio sur (Valero, 2006).

En la Figura 2.4 se muestra el sistema de coordenadas ecuatoriales usado para arreglos de antenas terrestres. Se puede ver un sistema cartesiano donde X e Y están en un plano paralelo al ecuador de la tierra, donde X es parte del plano meridiano (definido como el plano que pasa por los polos de la tierra y el punto de referencia del arreglo de antenas, Y se mide hacia el este, y Z hacia el polo norte (A. Richard Thompson, 2004). En términos de ángulo horario H y declinación δ , las coordenadas (X, Y, Z) se miden como lo muestra la Figura 2.3.

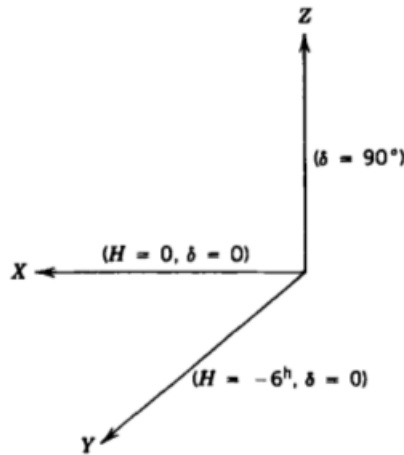


Figura 2.3: Sistema de coordenadas (X, Y, Z) y la dirección de los ejes en términos de ángulo horario H y declinación δ . Fuente: (A. Richard Thompson, 2004)

Es importante destacar que las estrellas describen en su movimiento diurno aparente, círculos paralelos de la Esfera Celeste y tienen por lo tanto declinación constante. Sin embargo, no sucede lo mismo con el Sol, la Luna o los planetas.

Teniendo esto en consideración, una coordenada del plano (u, v) que genera un baseline está dada por:



Figura 2.4: Relación entre los sistemas de coordenadas (X, Y, Z) y (u, v, w) . El sistema (u, v, w) está definido por la observación en la dirección del punto S , el cual tiene un ángulo horario H y una declinación δ . Como S está en la mitad del hemisferio este, H es negativo (sentido anti-horario). Fuente: (A. Richard Thompson, 2004)

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \sin H & \cos H & 0 \\ -\sin \delta \cos H & \sin \delta \sin H & \cos \delta \\ \cos \delta \cos H & -\cos \delta \sin H & \sin \delta \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (2.1)$$

Donde las coordenadas en el sistema (X, Y, Z) están dadas por:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = D \begin{bmatrix} \cos d \cos h \\ -\cos d \sin h \\ \sin d \end{bmatrix} \quad (2.2)$$

Aquí, (H, δ) son el ángulo horario y la declinación de la posición de fase de referencia. Por otra parte es importante especificar el vector del *baseline* en términos de su longitud D y el ángulo horario y declinación (h, d) de la intersección de la dirección del *baseline* con el norte celestial. Reemplazando la Ecuación 2.2 en 2.1 y usando identidades trigonométricas se tiene que:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = D \begin{bmatrix} \cos d \sin (H - h) \\ \sin d \cos \delta - \cos d \sin \delta \cos (H - h) \\ \sin d \sin \delta + \cos d \cos \delta \cos (H - h) \end{bmatrix} \quad (2.3)$$

Si bien el sistema (D, h, d) era el más usado con instrumentos que sólo consistían en dos antenas. Cuando se tiene dos antenas o más, la práctica usual es utilizar el sistema de

coordenadas horizontales para determinar la elevación \mathcal{E} , el *azimuth* \mathcal{A} y el largo del baseline D .

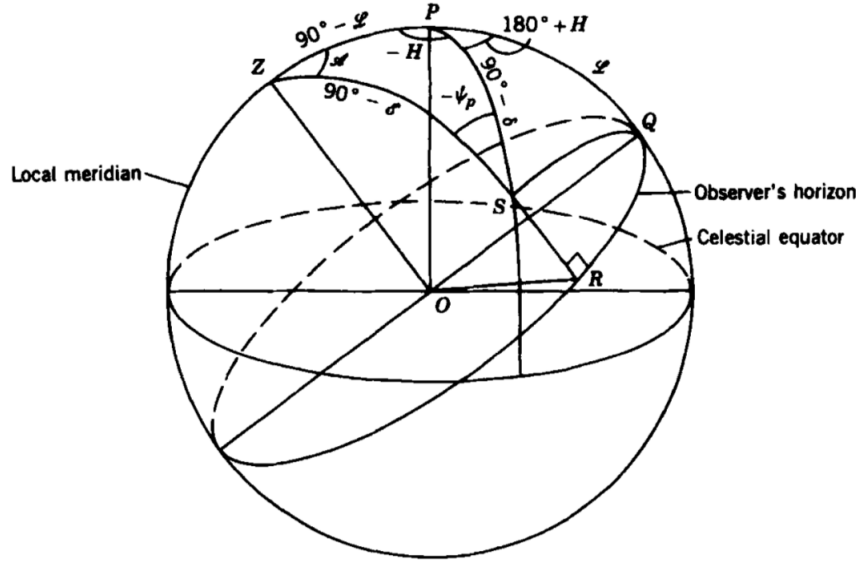


Figura 2.5: Relación entre coordenadas horizontales y coordenadas ecuatoriales horarias. Fuente: (A. Richard Thompson, 2004)

El plano de referencia es el horizonte, perpendicular a la vertical del lugar de observación que pasa por el centro de la Tierra. Z es el *zenith* del observador y es la parte superior (más próximo al polo norte P) del plano vertical perpendicular al plano de observación. Por otra parte, el meridiano del lugar que pasa por el *zenith*, corta al horizonte en una recta llamada meridiana que determina la dirección Norte-Sur. En este caso Q indica la dirección Norte, y luego mirando hacia éste, el Este queda a la derecha y el Oeste a la izquierda. Se llama *azimuth* al ángulo \mathcal{A} medido de Norte a Este en sentido anti-horario. Además, se llama elevación al ángulo \mathcal{E} formado por la dirección de la fuente S con el plano del horizonte.

Las fórmulas de conversión de coordenadas derivan de la aplicación de las reglas de seno y coseno para triángulos esféricos en la Figura 2.5. Por lo tanto para un observador a una latitud \mathcal{L} se tiene que:

$$\begin{aligned}\sin d &= \sin \mathcal{L} \sin \mathcal{E} + \cos \mathcal{L} \cos \mathcal{E} \cos \mathcal{A} \\ \cos d \cos h &= \cos \mathcal{L} \sin \mathcal{E} - \sin \mathcal{L} \cos \mathcal{E} \cos \mathcal{A} \\ \cos d \sin h &= -\cos \mathcal{E} \sin \mathcal{A}\end{aligned}\tag{2.4}$$

Reemplazando en la Ecuación 2.2 se tiene:

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = D \begin{bmatrix} \cos \mathcal{L} \sin \mathcal{E} - \sin \mathcal{L} \cos \mathcal{E} \cos \mathcal{A} \\ \cos \mathcal{E} \sin \mathcal{A} \\ \sin \mathcal{L} \sin \mathcal{E} + \cos \mathcal{L} \cos \mathcal{E} \cos \mathcal{A} \end{bmatrix} \quad (2.5)$$

Por otra parte, si se quisiera proyectar el plano (u, v) en la Figura 2.4 bastaría que hicieramos $H = 0$ en la ecuación 2.1 o 2.3, con esto se forma una ecuación de la elipse en el plano (u, v) de la forma:

$$\frac{u^2}{(\sqrt{X^2 + Y^2})^2} + \frac{(v - Z \cos \delta)^2}{(\sin \delta \sqrt{X^2 + Y^2})^2} = 1 \quad (2.6)$$

Si se analiza la ecuación 2.6 es posible notar que su centro está en $(u, v) = (0, Z \cos \delta)$, su eje semimayor se encuentra en $\sqrt{X^2 + Y^2}$ y su eje semimenor en $\sin \delta \sqrt{X^2 + Y^2}$. Este arco trazado durante la observación depende del *azimuth*, la elevación y la latitud del *baseline*; la declinación de la fuente, y el rango de ángulo horario que se cubre como se muestra en la Figura 2.7. Además, como una de las propiedades de la transformada de Fourier es la simetría Hermitiana $V(u, v) = V(-u, -v)$, el correlacionador devuelve como salida el valor de la visibilidad en dos puntos en el plano (u, v) como se muestra en la Figura 2.6.

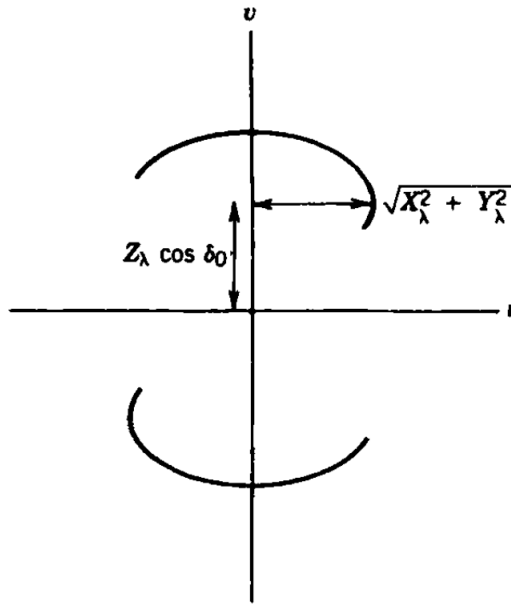


Figura 2.6: Centro y eje semimayor de la elipse (Ecuación 2.6). Fuente: (A. Richard Thompson, 2004)

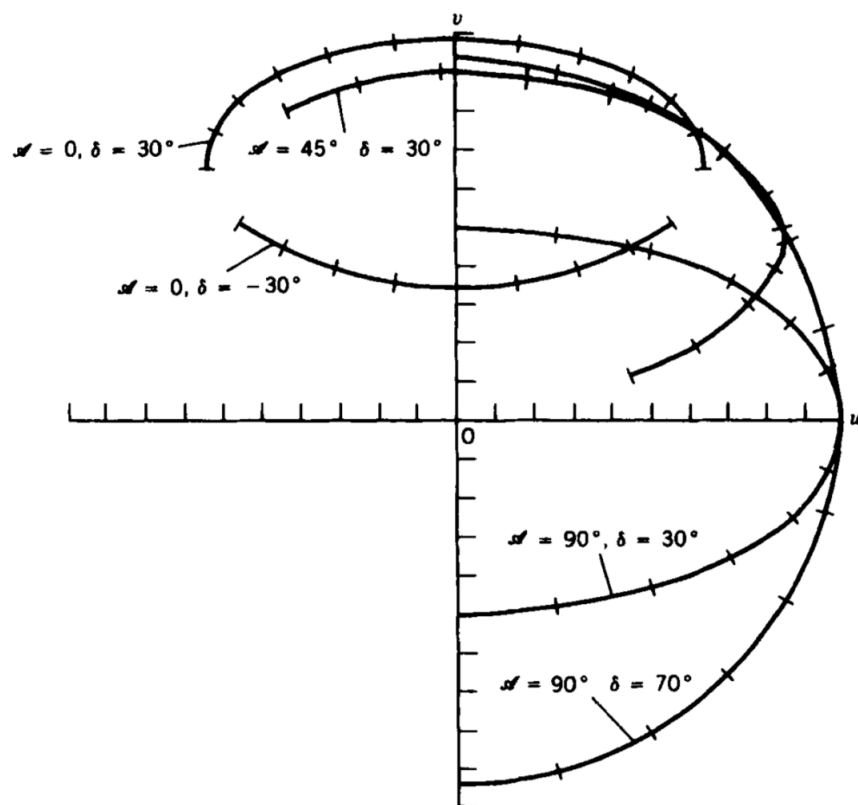


Figura 2.7: Elipses muestreadas con diferentes parámetros en el plano (u, v) . Fuente: (A. Richard Thompson, 2004)

Por otro lado, cada antena contiene un receptor (también llamado *Front-end*) que permite que las éstas capten información en diez bandas de frecuencia diferentes. Para ello cada antena está equipada con un criostato y su criorefrigerador adjunto. Estos criostatos contienen receptores que están montados y pueden ser reemplazados de forma fácil. En la tabla 2.1 se adjuntan los datos de las distintas bandas de frecuencia que ALMA puede cubrir gracias a su front-end.

Banda	Rango de frecuencia (GHz)	Temperatura (K)
1	31 - 45	26
2	67 - 90	47
3	84 - 116	60
4	125 - 163	82
5	162 - 211	105
6	211 - 275	136
7	275 - 373	219
8	385 - 500	292
9	602 - 720	261
10	787 - 950	344

Tabla 2.1: Las 10 Bandas de frecuencia de ALMA

Por ejemplo, la estrella HLTauri ha sido muestreada en las bandas 3, 6 y 7. En la Figura 2.8 es posible apreciar cada uno de los planos (u, v) formados en cada muestreo.

Es importante destacar que los datos muestreados en una banda se separan en *spectral windows* que son espectros contiguos cuyas frecuencia están uniformemente espaciadas con anchos de banda desde 58.6 MHz hasta 1.875 GHz (Mangum & Warmels, 2015). Asimismo, cada *spectral window* se divide uniformemente en canales. Es posible visualizar esto en la Figura 2.9, en donde se ha puesto como ejemplo el muestreo del objeto HL-Tauri.

Luego de que las ondas milimétricas y submilimétricas son recibidas por las antenas, éstas deben ser discretizadas y procesadas en uno de los dos correlacionadores. El correlacionador *64-input* es usado principalmente por los conjuntos de antena de 12 metros, mientras que el correlacionador ACA es usado por las antenas de 7 metros que componen el *Atacama Compact Array*. Así, ambos correlacionadores funcionan simultáneamente e independientemente, por lo que mientras las antenas de 12 metros pueden estar observando un objeto y usan el correlacionador *64-input*, las antenas de 7 metros pueden estar observando el mismo y otro objeto, y además, usando el correlacionador ACA (Mangum & Warmels, 2015).

Es importante destacar que los correlacionadores reciben señales de voltaje desde cada antena, calculan la correlación cruzada y autocorrelación para éstas por cada par de antenas (*baselines*), y producen las visibilidades de valores complejos que los astrónomos reciben



Figura 2.8: Plano (u, v) del objeto HLTauri (HLTau) en diferentes bandas. Fuente: Elaboración propia

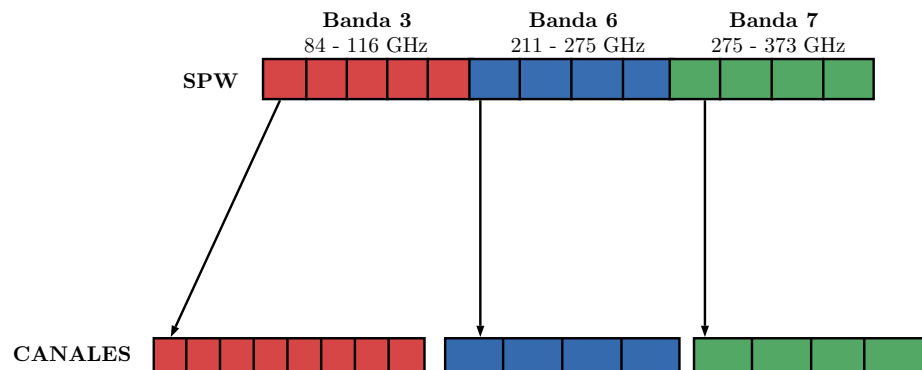


Figura 2.9: Bandas, *spectral windows* y canales del objeto HL-Tauri. Fuente: Elaboración propia

para sintetizar imágenes. Para entender de mejor forma qué son las visibilidades y cual es la implicancia que tienen en la síntesis de imágenes es necesario conocer los conceptos esenciales de interferometría.

2.2 PRINCIPIOS Y CONCEPTOS DE INTERFEROMETRÍA

La interferometría es una técnica usada para obtener imágenes de alta resolución angular de un fenómeno astronómico particular. Esta implica la combinación de señales recibidas desde el cielo por dos antenas separadas físicamente. Estas señales contienen ruido, permitiendo así que la distribución del brillo del cielo sea muestreada en una escala angular más pequeña que la de una sola antena.

La resolución angular de un interferómetro, δ , es el ángulo más pequeño en el que dos objetos pueden ser separados en orden de distinguirlos como objetos diferentes. Usando la teoría de la difracción, se puede demostrar que para un interferómetro circular en particular de diámetro D , y una radiación de longitud de onda λ , este valor es $\delta \propto \frac{\lambda}{D}$. Esto quiere decir, que para grandes longitudes de onda (pequeñas frecuencias), se necesita un gran telescopio para obtener una buena resolución. Sin embargo, existen límites prácticos para construir un telescopio de un gran tamaño. Es por ello que haciendo uso de un conjunto de interferómetros situados a una distancia D , es posible obtener la misma resolución que un solo telescopio de radio D .

La relación entre la distribución del brillo del cielo y una visibilidad compleja está dada por el teorema de van Cittert-Zernike (Zernike, 1938) y es la base de la interferometría. Dado que los interferómetros no obtienen la imagen del cielo directamente, sino que obtienen visibilidades, que son que son la transformada de Fourier de la distribución del brillo del cielo en el plano de la imagen. Cada par de antenas forma un vector $\vec{k} = (u, v)$ en el plano de Fourier. Así mismo, la distribución del brillo del cielo es la transformada inversa de Fourier de las visibilidades complejas. Por lo tanto, la visibilidad $V(\vec{k})$ para un par de antenas con un *baseline* \vec{k} , y su respectiva transformada inversa es:

$$V(\vec{k}) = \int \int A(\vec{x}) I(\vec{x}) e^{-2\pi i \vec{k} \cdot \vec{x}} dx dy \quad (2.7)$$

$$A(\vec{x}) I(\vec{x}) = \int \int V(\vec{k}) e^{2\pi i \vec{k} \cdot \vec{x}} du dv \quad (2.8)$$

Donde $\vec{x} = (x, y)$ son las coordenadas del objeto a observar y estudiar, $A(\vec{x})$ es el *primary beam*, $I(\vec{x})$ es la intensidad del objeto en \vec{x} , y \vec{k} es el vector que se forma por cada par de antenas.

Esto quiere decir que la imagen de la distribución del brillo del cielo puede recuperarse a través del muestreo de la distribución de visibilidades complejas en el plano (u, v) . En esencia, una imagen es la transformada de Fourier de las visibilidades donde cada una de éstas tiene una amplitud y una fase representando el brillo y la posición relativa de la emisión en una escala

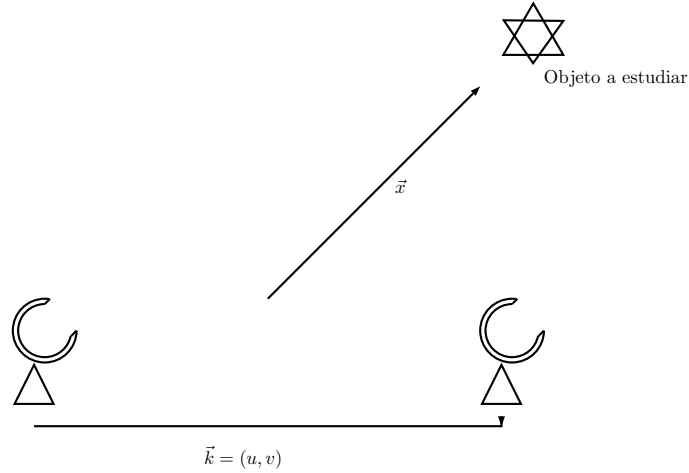


Figura 2.10: Dos antenas y su respectivo *baseline*. \vec{x} es la posición del objeto estudiado, y por otro lado \vec{k} es el *baseline* correspondiente a las dos antenas. Fuente: Elaboración propia.

angular específica.

Los datos astronómicos resultan desde la adición de el ruido instrumental hasta la convolución de la imagen del cielo con la respuesta instrumental. Debido al muestreo incompleto en el plano (u, v) , la obtención de una imagen del cielo se convierte en un problema inverso que requiere algoritmos de síntesis de imágenes.

2.3 MÉTODOS EXISTENTES DE DECONVOLUCIÓN

2.3.1 CLEAN

Uno de los algoritmos más exitosos y utilizados por la comunidad astronómica es CLEAN, creado por (Hogbom, 1974). CLEAN postula que la distribución de intensidad está compuesta de fuentes puntuales. Dado que la imagen de una fuente puntual está dada por la convolución de ésta con el *dirty beam* o Point Spread Function (PSF) (Ecuación 2.9). Donde este último está dado por:

$$B(x, y) = \int \int S(u, v) \exp\{2\pi j(ux + vy)\} du dv \quad (2.9)$$

En cada iteración CLEAN encuentra el punto más brillante en la *dirty image* (Ecuación 2.10) que está dada por:

$$I_D(x, y) = \int \int V(u, v) S(u, v) e^{2\pi i(ux+vy)} du dv \quad (2.10)$$

Donde,

$$I^T(x, y) = \int \int V(u, v) e^{2\pi i(ux+vy)} du dv \quad (2.11)$$

Y luego por el teorema de la convolución se tiene que:

$$I_D(x, y) = I^T(x, y) * B(x, y) \quad (2.12)$$

CLEAN agrega este punto (posición e intensidad) a la lista de fuentes puntuales. Luego, una fracción de ésta (γ , $0 < \gamma < 1$) es removida de la *dirty image* y de las visibilidades. Este proceso se repite hasta que la imagen residual (I_D) sea solamente ruido.

Finalmente se calcula la imagen reconstruida convolucionando la lista de fuentes puntuales con el *beam* de CLEAN, que se asume la mayoría de las veces como una Gaussiana.

Algoritmo 2.1: Algoritmo CLEAN

- 1: Compute $I_D(x, y)$
 - 2: $B_R(x, y) = \text{Gaussian}$
 - 3: $i = 0$
 - 4: **while** I_D not noise-like **do**
 - 5: $(x_i, y_i) = \arg \max I^D(x, y)$
 - 6: $\lambda_i = I(x_i, y_i)$
 - 7: **for all** (u, v) **do**
 - 8: $V(u, v) - = \gamma \lambda_i \exp\{-2\pi j(ux_i + vy_i)\}$
 - 9: **end for**
 - 10: Recompute $I_D(x, y)$
 - 11: $i = i + 1$
 - 12: **end while**
 - 13: $I_R(x, y) = I_D(x, y) + \sum_i \gamma \lambda_i B_R(x - x_i, y - y_i)$
-

2.3.2 MEM Mono-frecuencia

El método de máxima entropía (MEM) encuentra la imagen que simultáneamente se ajuste mejor a los datos, dentro de un nivel de ruido y que maximice la entropía S . Esto se lleva

a cabo minimizando la función:

En MEM, la imagen y los datos son considerados variables aleatorias con distribuciones de probabilidad conocidas.

Sea $P(V|I)$ la probabilidad de ver las visibilidades V dada la imagen I y sea $P(I)$ un conocimiento *a priori* de la imagen. Entonces mediante el teorema de Bayes se obtiene:

$$P(I|V) = \frac{P(V|I)P(I)}{P(V)} \quad (2.13)$$

La probabilidad $P(V|I)$ puede ser aproximada mediante el hecho de que las visibilidades están corruptas por un ruido Gaussiano. Entonces dada la función modelo de visibilidades $V_k^m(I)$ y las varianzas observadas σ_k , la probabilidad puede modelarse como lo muestra la ecuación 2.14, donde V_k^o son las visibilidades observadas.

$$P(V|I) \propto \exp \left\{ - \sum_j^Z \frac{|V_j^m(I) - V_j^o|^2}{\sigma_j^2} \right\} \quad (2.14)$$

El *prior* $P(I)$ se considera una distribución multinomial de una intensidad total discreta que cubre la imagen completa. Sea n el número de píxeles y sea N_i el número de fotones en el píxel i , el *prior* puede expresarse de la forma:

$$P(I) = \frac{N!}{n^N \prod_i N_i!} \quad (2.15)$$

Donde $N = \sum_i N_i$ y la intensidad de la imagen en el píxel i es interpretado en el modelo como $I_i \propto N_i$.

El término $P(V)$ en la ecuación 2.13 es independiente de I , por lo tanto no forma parte de la ecuación. Usando logaritmo y la aproximación de Stirling para factoriales y además omitiendo constantes aditivas relevantes el problema de reconstrucción de la imagen se convierte en la búsqueda de una imagen que satisfaga la probabilidad Maximum a Posteriori (MAP).

$$\arg \max_I \left\{ - \frac{1}{2} \sum_j^Z \frac{|V_j^m(I) - V_j^o|^2}{\sigma_j^2} - \lambda \sum_k^n I_k \log \frac{I_k}{G} \right\} \quad (2.16)$$

Se reconoce al primer término de la ecuación 2.16 como χ^2 y al segundo como la entropía de Shannon S . Estos dos parámetros son relevantes en el modelo, así como λ que tiene un rol similar a los multiplicadores de Lagrange y actúa como un penalizador.

Finalmente, se cambia el signo de la ecuación 2.16, convirtiendo el problema en una minimización.

$$\Phi = \frac{1}{2} \sum_j^Z \frac{|V_j^m(I) - V_j^o|^2}{\sigma_j^2} + \lambda \sum_k^n I_k \log \frac{I_k}{G} \quad (2.17)$$

Note que la Ecuación 2.17 es no lineal, por lo que debe ser minimizada con los algoritmos apropiados, en el caso de la solución que se propone, el método usado es el del gradiente conjugado Polak-Ribiere (Polak & Ribiere, 1969). Además, es importante destacar que esta ecuación asume la existencia de un *spectral window* y un solo canal.

En el Anexo A se adjunta el cálculo del gradiente de esta función.

2.3.3 MEM Multi-frecuencia

En la práctica las visibilidades podrían ser muestreadas como lo indica la Figura 2.9, por lo que las ecuaciones de χ^2 (Ecuación 1.2), $\nabla\chi^2$ (Ecuación A.19) y por ende Φ (Ecuación 2.17) y $\nabla\Phi$ (Ecuación A.22) deben ser modificadas. Al estar los datos agrupados por *spectral windows* y canales es necesario agregar operandos de sumatoria tanto a χ^2 como $\nabla\chi^2$ quedando entonces como:

$$\chi^2 = \frac{1}{2} \sum_s^I \sum_c^C \sum_j^Z \frac{|V_{scj}^m(I) - V_{scj}^o|^2}{\sigma_{scj}^2} \quad (2.18)$$

En donde I es el número total de *spectral windows* del set de datos, C el número total de canales de la *spectral window* s y Z el número total de visibilidades en el canal c .

Asimismo, $\nabla\chi^2$ se modifica de la misma forma, quedando:

$$\frac{\partial\chi^2}{\partial I_k} = \sum_s^I \sum_c^C \sum_j^Z W_{scj} \left[\text{Re}(V_{scj}^R) \cos(2\pi\langle X_k, Z_{scj} \rangle) - \text{Im}(V_{scj}^R) \sin(2\pi\langle X_k, Z_{scj} \rangle) \right] \quad (2.19)$$

Sin embargo, como el número de canales de cada *spectral window* es el mismo, es posible deshacerse de un operando de sumatoria, quedando:

$$\chi^2 = \frac{1}{2} \sum_c^{I \cdot C} \sum_j^Z \frac{|V_{cj}^m(I) - V_{cj}^o|^2}{\sigma_{cj}^2} \quad (2.20)$$

$$\frac{\partial\chi^2}{\partial I_k} = \sum_c^{I \cdot C} \sum_j^Z W_{cj} \left[\text{Re}(V_{cj}^R) \cos(2\pi\langle X_k, Z_{cj} \rangle) - \text{Im}(V_{cj}^R) \sin(2\pi\langle X_k, Z_{cj} \rangle) \right] \quad (2.21)$$

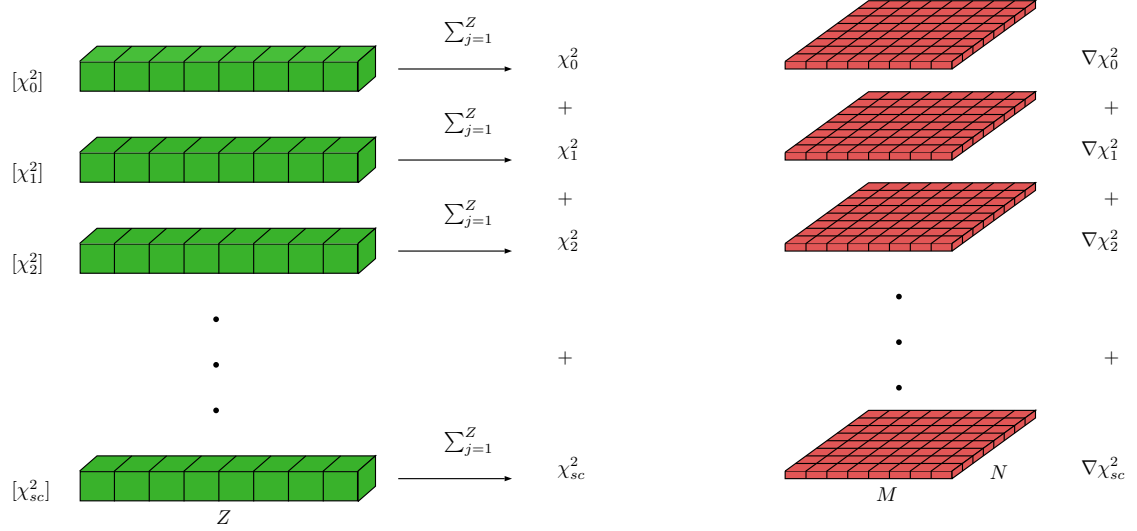


Figura 2.11: Ilustración del cálculo de χ^2 y $\nabla\chi^2$ con conjuntos de datos multi-frecuencia. Fuente: Elaboración propia

Algoritmo 2.2: Algoritmo del kernel 1D que calcula el vector χ^2

- 1: $j = globalId.x$
 - 2: **if** $j < Z$ **then**
 - 3: $\chi^2(j) = w(j) * (Re(V^R(j))^2 + Im(V^R(j))^2)$
 - 4: **end if**
-

Algoritmo 2.3: Algoritmo del kernel 2D que calcula el vector S

- 1: $k = globalId.x$
 - 2: $i = globalId.y$
 - 3: $S(i, k) = I(i, k) * \log(I(i, k)/G)$
-

Algoritmo 2.4: Algoritmo del kernel 2D que calcula $\nabla\chi^2$

```
1:  $k = globalId.x$ 
2:  $i = globalId.y$ 
3:  $x_0 = x_{obs}$ 
4:  $y_0 = y_{obs}$ 
5:  $x = (k - x_0) * \Delta x$ 
6:  $y = (i - y_0) * \Delta y$ 
7:  $\nabla\chi^2(i, k) = 0$ 
8: for  $j = 0$  to  $Z$  do
9:    $\nabla\chi^2(i, k) +=$ 
       $w(j) * [Re(V^R(j)) * \cos(2\pi\langle(u(j), v(j)), (x, y)\rangle) + Im(V^R(j)) * \sin(2\pi\langle(u(j), v(j)), (x, y)\rangle)]$ 
10: end for
```

Algoritmo 2.5: Algoritmo del kernel 2D que calcula el vector ∇S

```
1:  $k = globalId.x$ 
2:  $i = globalId.y$ 
3:  $\nabla S(i, k) = \lambda * (\log(I(i, k)/G) + 1)$ 
```

CAPÍTULO 3. OPTIMIZACIÓN

En este capítulo se aborda el uso de método de gradiente conjugado para la minimización de la función no lineal 2.17, destacando su funcionamiento y como esto se aplica a el problema abordado.

3.1 MÉTODOS DE GRADIENTE CONJUGADO EN MÚLTIPLES DIMENSIONES

Los dos métodos de gradiente conjugado más importantes son el método de Fletcher-Reeves y el método de Polak-Ribiere. Estos métodos están relacionados puesto que sólo un paso de su algoritmo los diferencia. Estos algoritmos convergen al mínimo de una función en un número finito de iteraciones, para ello se localiza el intervalo donde se encuentra el mínimo, y luego se reduce.

Algoritmo 3.1: Algoritmo de Fletcher-Reeves/Polak-Ribiere

```
1: Select  $I_0 \in \mathbb{R}^n$ 
2:  $d_0 = -\nabla\Phi(I_0)$ 
3: for  $i = 0$  to MAX ITERATIONS do
4:   Compute  $\alpha_i$  such that  $\Phi(I_i + \alpha_i d_i) = \min_{\alpha} \Phi(I_i + \alpha_i d_i)$ 
5:   Set  $I_{i+1} = I_i + \alpha_i d_i$ 
6:   if  $\|\nabla\Phi(I_{i+1})\| = 0$  then
7:     Stop.
8:   else
9:      $g_{i+1} = -\nabla\Phi(I_{i+1})$ 
10:     $\beta_i = \frac{g_{i+1}^T g_{i+1}}{d_i^T d_i}$  //Fletcher-Reeves
11:     $\beta_i = \frac{g_{i+1}^T (g_{i+1} - d_i)}{d_i^T d_i}$  //Polak-Ribiere
12:     $d_{i+1} = g_{i+1} + \beta_i d_i$ 
13:   end if
14: end for
```

En el algoritmo 3.1 las líneas 1 y 2 muestran que el algoritmo comienza con una imagen positiva plana y que luego se calcula la dirección de búsqueda inicial. En la línea 3 comienza el ciclo para encontrar el mínimo. Para calcular la siguiente imagen se debe encontrar un valor α que minimice la función objetivo siguiendo la dirección de búsqueda y sujeta a la restricción de positividad. En la línea 6, el algoritmo valida si se ha encontrado el mínimo, si

se encuentra, entonces el algoritmo se detiene, de lo contrario se calcula una nueva dirección de búsqueda usando el valor de β según Fletcher-Reeves (línea 10) o Polak-Ribiere (línea 11). Este último punto es importante, debido a que estas dos formas funcionan de la misma manera para ecuaciones de forma cuadrática. Sin embargo, en la realidad las funciones no son exactamente cuadráticas, y llegando al mínimo de la forma cuadrática se hace necesario llevar a cabo otro conjunto de iteraciones. En ese sentido, la fórmula de Polak-Ribiere funciona mejor cuando se hace necesario llegar hasta estos puntos (Press et al., 1992).

CAPÍTULO 4. MULTI-GPU

La tarjeta Tesla K80 es una GPU que contiene dos tarjetas en una sola. Junto a ello, un sistema puede tener múltiples tarjetas GPU de alto cómputo. Por lo que se han querido aprovechar estas características para construir una solución que pueda ser usada tanto en una sola GPU como en múltiples, permitiendo así conseguir un mayor *speedup*. A continuación se especifican las características de la solución para múltiples GPU.

4.1 COMUNICACIÓN PEER-TO-PEER

Una tarjeta puede comunicarse con la memoria de otra si es que la aplicación es ejecutada como un proceso de 64-bit y si la capacidad de cómputo de las tarjetas es 2.0 o más. Por ejemplo, una función que se ejecuta en GPU (*kernel*) puede ejecutarse en una tarjeta y referenciar a un puntero cuya memoria haya sido almacenado en otra. Esto permite que la comunicación entre tarjetas sea más rápida debido a que si el acceso *peer-to-peer* se habilita por medio de `cudaDeviceEnablePeerAccess()` ya no es necesario que los datos pasen por la memoria del sistema (NVIDIA Corporation, 2015). Antes de utilizar la tecnología *peer-to-peer* es necesario saber cuál es la topología del sistema para saber como los dispositivos PCI Express se conectan entre ellos y con CPU, esto es posible mediante el comando `nvidia-smi topo -m`. Por ejemplo, la topología de Belka, sistema en donde se hacen las pruebas de este trabajo, puede verse en la Tabla 4.1.

	GPU 0	GPU 1	GPU 2	GPU 3	GPU 4	GPU 5	GPU 6	GPU 7	CPU Affinity
GPU 0	X	PIX	PHB	PHB	SOC	SOC	SOC	SOC	0-9
GPU 1	PIX	X	PHB	PHB	SOC	SOC	SOC	SOC	0-9
GPU 2	PHB	PHB	X	PIX	SOC	SOC	SOC	SOC	0-9
GPU 3	PHB	PHB	PIX	X	SOC	SOC	SOC	SOC	0-9
GPU 4	SOC	SOC	SOC	SOC	X	PIX	PHB	PHB	10-19
GPU 5	SOC	SOC	SOC	SOC	PIX	X	PHB	PHB	10-19
GPU 6	SOC	SOC	SOC	SOC	PHB	PHB	X	PIX	10-19
GPU 7	SOC	SOC	SOC	SOC	PHB	PHB	PIX	X	10-19

Tabla 4.1: Topología de Belka

Las sigla PIX representa una comunicación a través de switch PCI Express interno, PHB representa la comunicación a través de un *host bridge* PCI Express y finalmente SOC representa la comunicación a nivel de socket. Esto nos dice que hay 4 Tesla K80 conectadas al sistema (recordar que una Tesla K80 contiene dos tarjetas conectadas mediante un switch interno) pero que un par está separado de otro mediante un socket. Por lo tanto, solo es posible

realizar comunicación P2P entre las GPU cero, uno, dos y tres; y las GPU, cuatro, cinco, seis, y siete.

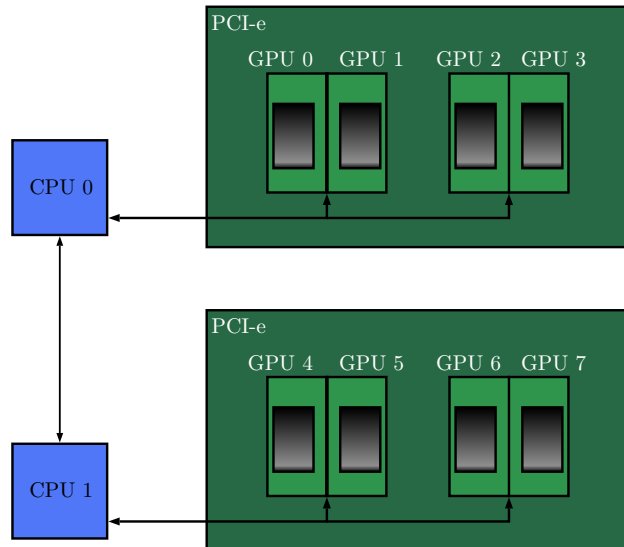


Figura 4.1: Ilustración de la topología de Belka. Fuente: Elaboración propia

En la Figura 4.1 se aprecia que en Belka existen dos puertos PCI-Express con dos Tesla K80 conectadas a cada uno. Cada puerto PCI-Express se conecta con una CPU de 10 cores cada una. Con esta topología solo es posible realizar una conexión *peer-to-peer* entre GPUs pertenecientes al mismo puerto.

4.2 DIRECCIONAMIENTO VIRTUAL UNIFICADO

Es posible que todos los dispositivos GPU y la CPU, utilicen un único espacio de memoria si todos los dispositivos tienen una capacidad de cómputo mayor o igual a 2.0. Como consecuencia, cuando se copia memoria a un dispositivo que está usando un espacio de direcciones unificado el parámetro `cudaMemcpyKind` de `cudaMemcpy()`, puede ser seteado a `cudaMemcpyDefault` para que CUDA automáticamente determine donde se encuentra almacenada la memoria de los punteros (NVIDIA Corporation, 2015).

Cualquier puntero de memoria en algún dispositivo puede ser creado por un *thread* del host y puede ser directamente referenciado dentro del mismo proceso. Sin embargo, esto no es posible fuera del proceso, por lo tanto, no se puede referenciar directamente a un puntero de memoria que pertenezca a un proceso distinto. Es por ello que para paralelizar el uso de múltiples GPU, y debido a que los threads que se crean pertenecen al mismo proceso se ha optado por usar OpenMP (NVIDIA Corporation, 2015).

4.3 OPENMP

Para acelerar el cálculo de χ^2 y $\nabla\chi^2$ es necesario que el uso de las múltiples tarjetas sea de forma paralela. Para ello, hay que tener en cuenta que cualquier thread del proceso ejecutado puede ejecutar la instrucción `cudaSetDevice(x)` que cambia la GPU con la cual se trabajará, por lo que los *kernels* que sean ejecutados desde ese punto en adelante serán ejecutados en la GPU x . Es posible usar esta instrucción de forma paralela y así hacer que cada thread invoque *kernels* en una GPU distinta y de forma concurrente. Por ejemplo, supongamos que se trabaja con el set de datos de HLTau en banda seis. Este set de datos tiene cuatro *spectral windows* y cuatro canales por cada una de éstas. Además el usuario puede ingresar como argumento el número de GPUs que desea utilizar, siendo el número total de canales la máxima cantidad de GPUs a utilizar. Asimismo, se usa la cláusula `# pragma omp parallel for schedule(static,1)` de OpenMP para recorrer los canales de todas las *spectral windows* de forma paralela planificando las hebras de forma Round Robin como se muestra en la Figura 4.2.

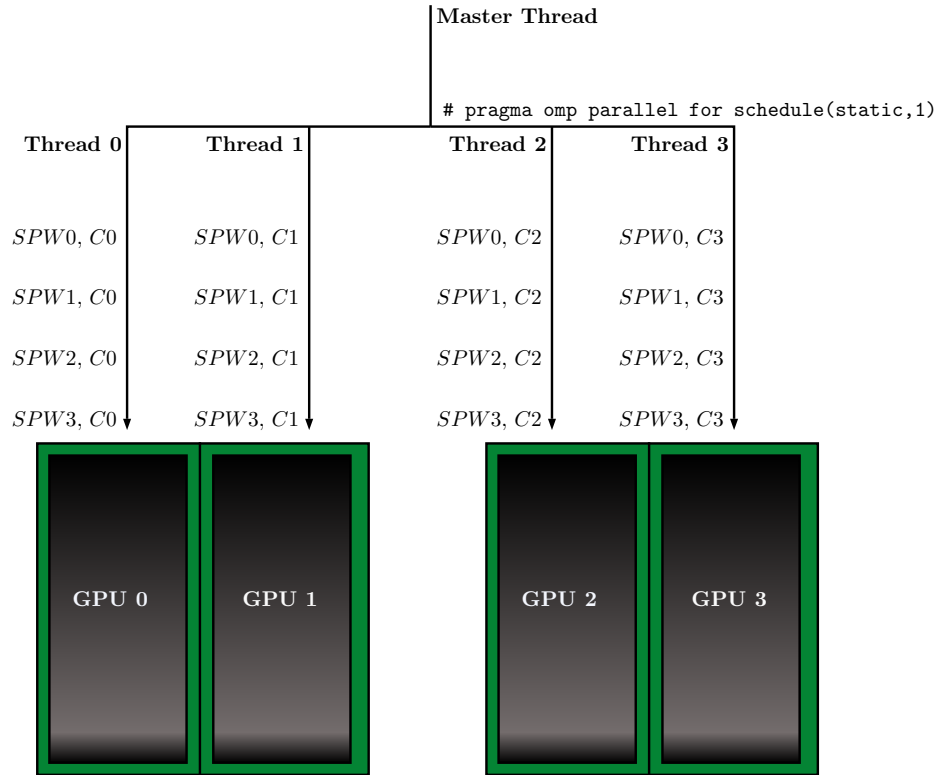


Figura 4.2: Uso de OpenMP en múltiples GPUs. Fuente: Elaboración propia

Debido a que tanto para el cálculo de χ^2 y $\nabla\chi^2$ se necesitan sumar los valores tanto de la función objetivo como del gradiente calculados en cada GPU respectivamente, se hace necesario crear secciones críticas, es decir, regiones en donde sólo un thread del *host* pueda

ingresar a la vez, con el fin de que no existan condiciones de carrera entre ellos. Para ello, se utiliza la cláusula `# pragma omp critical` de OpenMP. De esta forma se asegura que los cálculos de χ^2 y $\nabla\chi^2$ en múltiples frecuencias sean los correctos. Cabe destacar que para el cálculo de $\nabla\chi^2$ en estos casos, se necesita un *kernel* ubicado en la sección crítica que sume los valores de todos los dispositivos GPU.

Algoritmo 4.1: Cálculo de χ^2 en paralelo

```

1:  $\chi^2 = 0$ 
2: for  $i = 0$  to TOTAL CHANNELS do in parallel
3:   Compute  $\chi_i^2$ 
4:   BEGIN CRITICAL SECTION
5:    $\chi^2 = \chi^2 + \chi_i^2$ 
6:   END CRITICAL SECTION
7: end for

```

Algoritmo 4.2: Cálculo de $\nabla\chi^2$ en paralelo

```

1:  $\nabla\chi^2 = 0$ 
2: for  $i = 0$  to TOTAL CHANNELS do in parallel
3:   Compute  $\nabla\chi_i^2$ 
4:   BEGIN CRITICAL SECTION
5:    $\nabla\chi^2 = \nabla\chi^2 + \nabla\chi_i^2$ 
6:   END CRITICAL SECTION
7: end for

```

CAPÍTULO 5. PRUEBAS

Para validar la solución se hace necesario enfocar las pruebas en dos aspectos del algoritmo implementado. Por un lado, se tienen las pruebas de aceleración, es decir, saber cuánto más rápido es MEM implementado en GPU, comparado con la solución *multi-thread*. Por otra parte, se tienen las pruebas de caracterización que consisten en encontrar distintas características del algoritmo, tales como la resolución, nivel de ruido, entre otras.

A continuación, se presentarán con más detalle cada uno de estos aspectos.

5.1 BENCHMARKS

Según lo visto, el cálculo de Φ y $\nabla\Phi$ dependen tanto del número de visibilidades que se tengan en el set de datos como del número de píxeles de la imagen a reconstruir. Es por ello que para ver cuánto más rápido es el algoritmo es necesario escalar en ambas variables y comparar los resultados obtenidos con la solución multihebreada de MEM. Cabe destacar, que la solución no es una traducción de un código multihebra a un código GPU, sino un código completamente nuevo. Esto debido a que existen operaciones como suma y multiplicaciones de matrices que son posible paralelizar en GPU, además de la transformada de Fourier de CUDA (NVIDIA Corporation, 2012) que a diferencia de la transformada de Fourier de (Press et al., 1992) que utiliza `uvmem` multi-hebra toma ésta con signo negativo, y por lo tanto, el cálculo del gradiente cambia.

Debido a que existen en los códigos GPU y multi-hebra, la forma más justa de comparar el rendimiento de cada solución según la carga es comparando el tiempo promedio por iteración de cada una. Esto debido a que podría darse el caso en que éstas no converjan en la misma iteración, siendo injusto el comparar el tiempo total de ejecución.

Además, de hacer esta comparación, se debe analizar la aceleración obtenida usando múltiples GPU. Para ello, se harán pruebas con set de datos que contengan más de un canal y así comparar los speed-ups entre la solución multihebra, con una GPU y con múltiples GPU.

5.2 CARACTERIZACIÓN DE MEM

Para caracterizar el algoritmo, es decir, para comprender las propiedades de las imágenes obtenidas con el algoritmo, es necesario conocer su resolución espacial y su imagen

de residuo. Se entiende por resolución a la medida en arcosegundos del objeto más pequeño que se puede distinguir en una imagen. Así, se creará una imagen con un impulso, esto es, una imagen con el pixel (255, 255) con un valor positivo. Luego de esto, se simulará esta imagen con set de datos que permitan señalar la distinta resolución espacial que se tiene al usar *baselines* grandes y pequeños. En específico, se usarán los datos de HD142527 en banda nueve que tiene un *baseline* de 400 metros aproximadamente y de HLTauri en banda tres, seis y siete que tienen un *baseline* de aproximadamente 15 kilómetros, para simular la imagen de un impulso. Luego de ello, se reconstruirá cada set de datos con CLEAN y con MEM, de manera de encontrar las diferencias de resolución entre estos métodos. Para ello es necesario encontrar la anchura a media altura (FWHM) en arcosegundos del impulso reconstruido ya sea en CLEAN o en MEM.

Por otro lado, la imagen de residuos da a conocer qué tan buena es la deconvolución realizada con un algoritmo, para ello la intensidad por píxel de esta imagen debe ser parecida a un ruido independiente, es decir,

CAPÍTULO 6. RESULTADOS

CAPÍTULO 7. CONCLUSIONES

REFERENCIAS BIBLIOGRÁFICAS

- A. Richard Thompson, G., James M. Moran (2004). *Interferometry and Synthesis in Radio Astronomy*. Weinheim: WILEY-VCH Verlag GmbH & Co. KGaA.
- Ables, J. G. (1974). Maximum Entropy Spectral Analysis. *Astronomy and Astrophysics*, 15, 383.
- ALMA (2009). How does alma work. <http://www.almaobservatory.org/en/about-alma/how-does-alma-work>.
- Ben-David, C., & Leshem, A. (2008). Parametric high resolution techniques for radio astronomical imaging. *IEEE J.Sel.Top.Sig.Proc.*, 2, 670--684.
- Briggs, D. (1995). *High Fidelity Deconvolution of Moderately Resolved Sources*. D. Briggs.
URL <http://books.google.cl/books?id=JTU1GwAACAAJ>
- Casassus, S., Cabrera, G. F., Förster, F., Pearson, T. J., Readhead, A. C. S., & Dickinson, C. (2006). Morphological analysis of the centimeter-wave continuum in the dark cloud Icn 1622. *The Astrophysical Journal*, 639(2), 951.
URL <http://stacks.iop.org/0004-637X/639/i=2/a=951>
- Chen, W. (2011). The ill-posedness of the sampling problem and regularized sampling algorithm. *Digital Signal Processing*, 21(2), 375 -- 390.
URL <http://www.sciencedirect.com/science/article/pii/S1051200410001454>
- Cornwell, T. J. (2009). Hogbom's clean algorithm. impact on astronomy and beyond. *Astronomy and Astrophysics*, 500(1), 65--66.
- Cornwell, T. J., & Evans, K. F. (1985). A simple maximum entropy deconvolution algorithm. *Astronomy and Astrophysics*, 143, 77--83.
- Frieden, B. R. (1972). Restoring with maximum likelihood and maximum entropy. *J. Opt. Soc. Am.*, 62(4), 511--518.
URL <http://www.osapublishing.org/abstract.cfm?URI=josa-62-4-511>
- Gull, S. F., & Daniell, G. J. (1978). Image reconstruction from incomplete and noisy data. *Nature*, 272(5655), 686--690.
URL <http://dx.doi.org/10.1038/272686a0>
- Hager, W. W., & Zhang, H. (2006). A survey of nonlinear conjugate gradient methods. *Pacific journal of Optimization*, 2(1), 35--58.
- Hogbom, J. A. (1974). Aperture Synthesis with a Non-Regular Distribution of Interferometer Baselines. *Astron. Astrophys. Suppl. Ser.*, 15, 417--426.
- Leshem, A., & Jan Van Der Veen, A. (2000). Radio astronomical imaging in the presence of strong radio interference. *IEEE Trans. on Information Theory*, 46, 1730--1747.
- Lochner, M., Natarajan, I., Zwart, J. T. L., Smirnov, O., Bassett, B. A., Oozeer, N., & Kunz, M. (2015). Bayesian inference for radio observations. *Monthly Notices of the Royal Astronomical Society*, 450, 1308--1319.
- Mangum, J., & Warmels, R. (2015). ALMA Cycle 4 Technical Handbook. Tech. rep., ALMA. Version 1.0.
- Maréchal, P., & Wallach, D. (2009). Fourier synthesis via partially finite convex programming. *Mathematical and Computer Modelling*, 49(11--12), 2206 -- 2212. Trends in Application of Mathematics to Medicine Trends in Application of Mathematics to Medicine.
URL <http://www.sciencedirect.com/science/article/pii/S0895717708002380>

- Narayan, R., & Nityananda, R. (1986). Maximum entropy image restoration in astronomy. *Annual Review of Astronomy and Astrophysics*, 24, 127--170.
- NVIDIA Corporation (2012). *CUDA Toolkit 5.0 CUFFT Library Programming Guide*. NVIDIA Corporation. Version 0.1.
- NVIDIA Corporation (2015). *NVIDIA CUDA Compute Unified Device Architecture Programming Guide*. NVIDIA Corporation. Version 7.5.
- Owens, J. D., Luebke, D., Govindaraju, N., Harris, M., Krüger, J., Lefohn, A., & Purcell, T. J. (2007). A survey of general-purpose computation on graphics hardware. *Computer Graphics Forum*, 26(1), 80--113.
URL <http://www.blackwell-synergy.com/doi/pdf/10.1111/j.1467-8659.2007.01012.x>
- Peck, A. B., & Beasley, A. J. (2008a). High resolution sub-millimeter imaging with ALMA. *Journal of Physics: Conference Series*, 131(1), 012049.
URL <http://stacks.iop.org/1742-6596/131/i=1/a=012049>
- Peck, A. B., & Beasley, A. J. (2008b). High resolution sub-millimeter imaging with ALMA. *Journal of Physics: Conference Series*, 131(1), 012049.
URL <http://stacks.iop.org/1742-6596/131/i=1/a=012049>
- Perkins, S., Marais, P., Zwart, J., Natarajan, I., & Smirnov, O. (2015). Montblanc: GPU accelerated radio interferometer measurement equations in support of bayesian inference for radio observations. *CoRR*, *abs/1501.07719*.
URL <http://arxiv.org/abs/1501.07719>
- Polak, E., & Ribiere, G. (1969). Note sur la convergence de méthodes de directions conjuguées. *ESAIM: Mathematical Modelling and Numerical Analysis - Modélisation Mathématique et Analyse Numérique*, (pp. 35--43).
- Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (1992). *Numerical Recipes in C (2Nd Ed.): The Art of Scientific Computing*. New York, NY, USA: Cambridge University Press.
- Roman, P. E. (2012). El problema de síntesis de fourier en radio-astronomía y su resolución mediante métodos variacionales. *Revista del Instituto Chileno de Investigación Operativa*, (2), 13--19.
- Skilling, J., & Bryan, R. K. (1984). Maximum entropy image reconstruction: general algorithm. *Montly Notices Royal Astronomical Society*, (211), 111--124.
- Skilling, J., & Gull, S. F. (1991). *Bayesian maximum entropy image reconstruction*, vol. Volume 20 of *Lecture Notes--Monograph Series*, (pp. 341--367). Hayward, CA: Institute of Mathematical Statistics.
URL <http://dx.doi.org/10.1214/lnms/1215460511>
- Taylor, G. B., Carilli, C. L., & Perley, R. A. (Eds.) (1999). *Synthesis Imaging in Radio Astronomy II*, vol. 180 of *Astronomical Society of the Pacific Conference Series*. San Francisco: Astronomical Society of the Pacific.
URL http://adsabs.harvard.edu/cgi-bin/nph-bib_query?bibcode=1999ASPC..180....T
- Valero, J. (2006). *Una mirada al universo: astronomía básica elemental*. Editorial Tébar, S. L.
URL <https://books.google.cl/books?id=YWL1kYrA2vcC>
- Wernecke, S. J., & D'Addario, L. R. (1977). Maximum entropy image reconstruction. *IEEE Trans. Computers*, 26(4), 351--364.
URL <http://doi.ieeecomputersociety.org/10.1109/TC.1977.1674845>

Wood, W., & Kleb, W. (2002). Extreme programming in a research environment. In D. Wells, & L. Williams (Eds.) *Extreme Programming and Agile Methods — XP/Agile Universe 2002*, vol. 2418 of *Lecture Notes in Computer Science*, (pp. 89--99). Springer Berlin Heidelberg.
URL http://dx.doi.org/10.1007/3-540-45672-4_9

Zernike, F. (1938). The concept of degree of coherence and its application to optical problems. *Physica*, 5(8), 785 -- 795.
URL <http://www.sciencedirect.com/science/article/pii/S0031891438802032>

ANEXO A. CÁLCULO DEL GRADIENTE DE LA FUNCIÓN OBJETIVO

$$\Phi = \chi^2 + \lambda S \quad (\text{A.1})$$

con

$$\chi^2 = \frac{1}{2} \sum_{j=1}^Z \frac{|V_j^m - V_j^o|^2}{\sigma_j^2} \quad (\text{A.2})$$

$$S = \sum_{k=1}^n I_k \log(I_k/G) \quad (\text{A.3})$$

con $0 \leq \lambda < \infty$.

Calculando el gradiente de la función Φ , es decir, derivando parcialmente con respecto a I_k :

$$[\nabla \Phi]_k = \frac{\partial \Phi}{\partial I_k} = \frac{\partial}{\partial I_k} \chi^2 + \lambda \frac{\partial}{\partial I_k} S \quad (\text{A.4})$$

Primero derivamos la derivada de χ^2 .

$$\frac{\partial \chi^2}{\partial I_k} = \frac{\partial}{\partial I_k} \frac{1}{2} \sum_j \frac{(V_j^m - V_j^o)^* \cdot (V_j^m - V_j^o)}{\sigma_j^2} \quad (\text{A.5})$$

$$= \frac{1}{2} \sum_j \frac{1}{\sigma_j^2} \left(\frac{\partial V_j^{m*}}{\partial I_k} (V_j^m - V_j^o) + \frac{\partial V_j^m}{\partial I_k} (V_j^m - V_j^o)^* \right) \quad (\text{A.6})$$

Note que V_j^o es constante y por lo tanto su derivada respecto a I_k es cero. Donde la transformada de Fourier de la imagen (I) es igual a :

$$V_j^m = \sum_{k=1}^n I_k e^{-2\pi i \langle X_k, Z_j \rangle} \quad (\text{A.7})$$

Donde $Z_j = (u_j, v_j)$, $X_k = (x_k, y_k)$, y el producto interno $\langle (a, b), (c, d) \rangle = ac + bd$. Luego el conjugado es:

$$V_j^{m*} = \sum_{k=1}^n I_k e^{2\pi i \langle X_k, Z_j \rangle} \quad (\text{A.8})$$

Derivando ambas visibilidades por separado con respecto a I_k se tiene que:

$$\frac{\partial V_j}{\partial I_k} = \exp\{-2\pi i \langle X_k, Z_j \rangle\} \quad (\text{A.9})$$

$$\frac{\partial V_j^*}{\partial I_k} = \exp\{+2\pi i \langle X_k, Z_j \rangle\} \quad (\text{A.10})$$

Definiedo el residuo de las visibilidades como:

$$V_j^R = V_j^m - V_j^o = \text{Re}(V_j^R) + i \cdot \text{Im}(V_j^R) \quad (\text{A.11})$$

Y

$$V_j^{R*} = (V_j^m - V_j^o)^* = \text{Re}(V_j^R) - i \cdot \text{Im}(V_j^R) \quad (\text{A.12})$$

Reemplazando estos términos se tiene:

$$\frac{\partial \chi^2}{\partial I_k} = \frac{1}{2} \sum_j \frac{1}{\sigma_j^2} \left[\exp\{2\pi i \langle X_k, Z_j \rangle\} \left(\text{Re}(V_j^R) + i \text{Im}(V_j^R) \right) + \exp\{-2\pi i \langle X_k, Z_j \rangle\} \left(\text{Re}(V_j^R) - i \text{Im}(V_j^R) \right) \right] \quad (\text{A.13})$$

Multiplicando distributivamente:

$$\frac{\partial \chi^2}{\partial I_k} = \frac{1}{2} \sum_j \frac{1}{\sigma_j^2} \left[\exp\{2\pi i \langle X_k, Z_j \rangle\} \text{Re}(V_j^R) + i \exp\{2\pi i \langle X_k, Z_j \rangle\} \text{Im}(V_j^R) - \exp\{-2\pi i \langle X_k, Z_j \rangle\} \text{Re}(V_j^R) - i \exp\{-2\pi i \langle X_k, Z_j \rangle\} \text{Im}(V_j^R) \right] \quad (\text{A.14})$$

Factorizando por la partes reales e imaginarias de V_j^R :

$$\frac{\partial \chi^2}{\partial I_k} = \frac{1}{2} \sum_j \frac{1}{\sigma_j^2} \left[\text{Re}(V_j^R) \left(\exp\{2\pi i \langle X_k, Z_j \rangle\} + \exp\{-2\pi i \langle X_k, Z_j \rangle\} \right) + i \text{Im}(V_j^R) \left(\exp\{2\pi i \langle X_k, Z_j \rangle\} - \exp\{-2\pi i \langle X_k, Z_j \rangle\} \right) \right] \quad (\text{A.15})$$

Usando las propiedades trigonométricas del seno y coseno se tiene que:

$$\frac{\partial \chi^2}{\partial I_k} = \frac{1}{2} \sum_j \frac{1}{\sigma_j^2} \left[2 \text{Re}(V_j^R) \cos(2\pi \langle X_k, Z_j \rangle) + 2i^2 \text{Im}(V_j^R) \sin(2\pi \langle X_k, Z_j \rangle) \right] \quad (\text{A.16})$$

$$= \sum_j \frac{1}{\sigma_j^2} \left[\text{Re}(V_j^R) \cos(2\pi \langle X_k, Z_j \rangle) - \text{Im}(V_j^R) \sin(2\pi \langle X_k, Z_j \rangle) \right] \quad (\text{A.17})$$

Definiendo

$$W_j = \frac{1}{\sigma_j^2} \quad (\text{A.18})$$

tenemos finalmente que

$$\frac{\partial \chi^2}{\partial I_k} = \sum_j W_j \left[\text{Re}(V_j^R) \cos(2\pi \langle X_k, Z_j \rangle) - \text{Im}(V_j^R) \sin(2\pi \langle X_k, Z_j \rangle) \right] \quad (\text{A.19})$$

Continuando con $\frac{\partial}{\partial I_k} S$

$$\frac{\partial S}{\partial I_k} = \frac{\partial}{\partial I_k} \sum_k I_k \log(I_k/G) \quad (\text{A.20})$$

Esto es igual a:

$$\frac{\partial S}{\partial I_k} = 1 + \log(I_k/G) \quad (\text{A.21})$$

Por tanto la coordenada k de $\nabla \Phi$:

$$[\nabla\Phi]_k = \sum_j W_j \left[\text{Re}(V_j^R) \cos(2\pi\langle X_k, Z_j \rangle) - \text{Im}(V_j^R) \sin(2\pi\langle X_k, Z_j \rangle) \right] + \lambda(1 + \log(I_k/G)) \quad (\text{A.22})$$