Accelerated coplanar facet radio synthesis imaging

Dissertation presented in fulfillment of the requirements for the degree of
MASTER OF SCIENCE
in the Department of Computer Science
University of Cape Town

Author:
Benjamin Hugo
Department of Computer Science,
University of Cape Town

Supervisors:
James Gain
Department of Computer Science,
University of Cape Town
Oleg Smirnov
Cyril Tasse
Department of Physics and Electronics,
Rhodes University

February 2016

Abstract

Imaging in radio astronomy entails the Fourier inversion of the relation between the sampled spatial coherence of an electromagnetic field and the intensity of its emitting source. This inversion is normally computed by performing a convolutional resampling step and applying the Inverse Fast Fourier Transform, because this leads to computational savings. Unfortunately, the resulting planar approximation of the sky is only valid over small regions. When imaging over wider fields of view, and in particular using telescope arrays with long non-East-West components, significant distortions are introduced in the computed image. We propose a coplanar faceting algorithm, where the sky is split up into many smaller images. Each of these narrow-field images are further corrected using a phase-correcting technique known as w-projection. This eliminates the projection error along the edges of the facets and ensures approximate coplanarity. The combination of faceting and w-projection approaches alleviates the memory constraints of previous w-projection implementations.

We compared the scaling performance of both single and double precision resampled images in both an optimized multi-threaded CPU implementation and a GPU implementation that uses a memory-access-limiting work distribution strategy. We found that such a w-faceting approach scales slightly better than a traditional w-projection approach on GPUs. We also found that double precision resampling on GPUs is about 71% slower than its single precision counterpart, making double precision resampling on GPUs less power efficient than CPU-based double precision resampling. Lastly, we have seen that employing only single precision in the resampling summations produces significant error in continuum images for a MeerKAT-sized array over long observations, especially when employing the large convolution filters necessary to create large images.

Acknowledgements

I would like to thank my supervisors James Gain, Oleg Smirnov and Cyril Tasse for their very insightful discussions, as well as their guidance over the period of February 2014 to February 2016, without which this work would have taken significantly longer to complete. James Gain and the Computer Science Department of the University of Cape Town provided the necessary equipment funding for a development machine, as well as work space for both years. Thanks go to the SKA South Africa and Oleg Smirnov for funding trips to Rhodes University to meet up and work with my Rhodes colleagues as well as work space at the SKA offices.

Secondly I would like to thank my peers in the Computer Science Department, the Radio Astronomy Techniques and Technologies group of the Department of Physics and Electronics at Rhodes University and the Cape Town branch of the Square Kilometre Array South Africa for their insight into some of the problems I ran into while developing the imaging software. Thanks also go to imaging experts Richard Perley and Bill Cotton from the National Radio Astronomy Observatory (NRAO, US) who provided valuable insights into facet imaging.

I would be in error to neglect to thank my friends, family and my especially my parents for their support over the last couple of years, especially in the latter half of 2015.

We acknowledge and thank the National Research Foundation (NRF) of the Republic of South Africa for providing the necessary funding to support this research. Use was made of the University of Cape Town Information and Communication Technology Services' high performance HEX cluster. In particular I would like to thank Andrew Lewis for his assistance in setting up the necessary software packages and dependencies of our software package on the cluster.

Plagiarism

I acknowledge that plagiarism is wrong and hereby declare that the work contained in this document and in the supporting software is my own, save for that which is properly acknowledged.

Benjamin Vorster Hugo

The source code for the imaging software package (along with its full development history) accompanying this document is publicly available on the Rhodes University Radio Astronomy Techniques and Technologies software repository at `https://www.github.com/ratt-ru/bullseye`. As such no part of the source code will be printed in this document.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 The synthesis imaging wide-field problem

In this work we investigate accelerating a computationally expensive resampling component used in the process that synthesizes sky images from the measurements made by radio telescopes arrays. The resampling computational costs arise, primarily, due to the costs of removing distortions introduced when creating images several degrees in size, known as *wide-field* distortions. This work compares the computational performance between parallel CPU- and GPU-based resampling implementations.

There is a well-known Fourier relationship between the sky brightness distribution and the measurements taken by antenna arrays [57, Lecture 1]. To synthesize an image of the radio sky, measurements taken by these radio arrays are normally inverted by resampling the points onto a regularly-spaced grid and performing an inverse Fast Fourier Transform [11, 60]. The synthesized images are convolved with the inverse transform of the sampling pattern of the array, requiring that the inversion step be called upon multiple times in an iterative deconvolution strategy such as Cotton-Schwab CLEAN [59, ch 11].

When synthesizing wide-field images using an array of telescopes with non-East-West antenna pairs, the Fourier relationship between the sky brightness distribution and array measurements breaks down. This occurs because of a combination of two factors: the Fast Fourier Transform only approximates the sky by a plane, and secondly the measurements taken by the telescope do not remain coplanar over the course of longer observations. Due to this "tilting" of the sampling plane the projected position of sources far away from the centre of the synthesized fields do not remain constant and the brightness of the sources is therefore smeared out over time. These two sources of error are collectively known as the problem of *non-coplanar wide-field imaging* [13].

One strategy for resolving the sources affected by this smearing is to split the sky up into small narrow-field ("facet") images, tiling the sky in a polyhedron-like fashion [13]. The synthesized images produced by such a non-coplanar faceting approach is hard to jointly deconvolve and require reprojection and intensity correction in overlapping areas. Another strategy uses convolution to introduce a correcting phase shift into each of the individual measurements taken by the instrument over time and is known as w-projection [12]. This approach relates the non-coplanar measurements to a single plane, eliminating the phase delay introduced by the tilted interferometer baselines.

In w-projection, the resampling cost can be related to the size of the produced images, whereas in

traditional faceting the cost rise with the number of facets. Unlike w-projection, faceting is less memory intensive, especially for large images and arrays with very long baseline components. The work in this thesis concentrates on creating coplanar facet images by combining facet imaging and w-projection in what we call *w-faceting*. Due to the nature of imaging using interferometers it is desirable to have baselines as long as possible to improve the resolving capability of the telescope, enough baselines in-between to create uniform sampling coverage in the measurement domain and as much collecting area as possible to improve the sensitivity of the instrument.

Instruments such as the Square Kilometre Array [9] and its pathfinders MeerKAT [6] and ASKAP [32] have significantly more baselines than some of their predecessors, such as the Jansky Very Large Array [44]. The number of baselines grow roughly quadratically with number of antennas, and improved spatial resolving capability decreases the integration time for each measurement made per baseline. Therefore, the data rates from these telescopes provide a tough computational problem for image synthesis. Luckily due to the linearity of the underlying relationship between the sky and the measurements, and the dimensions of the input data and output products, the synthesis pipeline lends itself to the parallel and distributed architecture of modern processing equipment. The work in this thesis focuses on investigating resampling scalability within shared-memory environments, comparing CPU- and GPU-based w-facet imaging. A fully distributed implementation across a cluster of machines is not within the scope of this work, although the shared-memory w-facet technique can be expanded to include multiple processing cluster nodes due to its data-parallel nature.

1.2 Research questions and aims

We have identified the following research questions:

1. Is GPU-based w-faceting more scalable than a CPU-based parallel and vectorized implementation?
2. Does single precision gridding introduce any significant error in the synthesized images?
3. What effect does double precision resampling have on GPU scalability?

The aim of this work is to build a scalable w-facet imager. At present a full accelerated deconvolution pipeline implementation is out of scope, and we focus solely on the "backward" synthesis step, which includes the computationally expensive convolutional resampling necessary to take the Inverse Fast Fourier Transform. Our imager will also support the ability to target regions of the sky in what is appropriately called *targeted faceting*.

1.3 Software approach

To enable our investigation into scalability we have built our own targeted facet imaging framework called *Bullseye*. The software package and source code is publicly available at the Rhodes University Radio Astronomy Techniques and Technologies Group repository `https://www.github.com/ratt-ru/bullseye` under the GNU General Public License. Our framework includes a command-line utility with options similar to those provided by other imagers such as WSClean [40] and CASA [31, 35]. The package reads radio measurement information from the widely-used Measurement Set [63, 62] format standard and writes the synthesized orthogonally-projected "dirty" facet images out in the Flexible

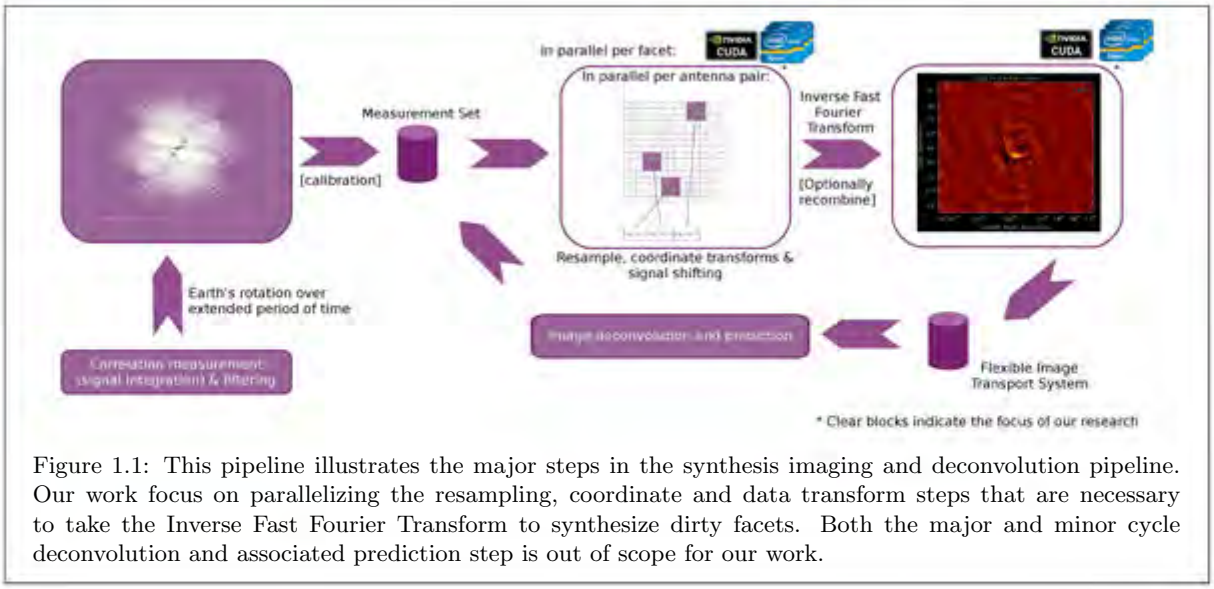


Figure 1.1: This pipeline illustrates the major steps in the synthesis imaging and deconvolution pipeline. Our work focus on parallelizing the resampling, coordinate and data transform steps that are necessary to take the Inverse Fast Fourier Transform to synthesize dirty facets. Both the major and minor cycle deconvolution and associated prediction step is out of scope for our work.

Image Transport System [43, 8] standard. Figure 1.1 shows the focus of our work within the synthesis imaging pipeline.

1.4 Outline

This document is divided into the following sections:

- **Chapter 2: Review of multi- and many-core processing models** gives readers not familiar with recent trends in CPU and GPU design an architectural overview of each, and a discussion on the processing terminology used in later chapters.
- **Chapter 3: The Radio Interferometric Measurement Equation** is aimed at readers coming from an engineering background or those unfamiliar with radio interferometry. The chapter gives a basic overview of radio interferometry, associated coordinate systems and derives the formal mathematical model for the relationship between the sky and the measurement domain.
- **Chapter 4: Wide-field image synthesis** discusses the inversion of the Radio Interferometric Measurement Equation and the associated wide-field problems. This technical discussion leads on directly to the imager design chapter.
- **Chapter 5: Bullseye: A parallel targeted facet imager** discusses the architectural design, processing algorithms used in our imager and our validation strategy.
- **Chapter 6: Performance analysis** presents imager performance in terms scalability and accuracy.
- **Chapter 7: Conclusion and future work** highlights the key findings made in the analysis and outlines the key areas where future work and investigation are necessary.

Chapter 2

Review of multi- and many-core processing models

In this chapter we focus on two shared-memory architectures widely used in parallel computing: multi-core CPU architectures and many-core GPU architectures. Recent trends in computing have favored the growth in throughput-driven parallel architectures, of which these two are currently the most common. Our imaging software will compare implementations on both these platforms and therefore we summarize the necessary background for the underlying concepts, as well as the terminology used in later chapters, before moving onto the problem context, description and solution. Distributed solutions to the imaging problem across parallel clusters of machines are out of scope for our current implementation and therefore we focus solely on shared-memory architectures in this discussion.

2.1 Multi-core CPU architectures

In this section we give the reader a brief overview of the historic development of CPUs and outline key design considerations in modern CPU architecture. The discussion is drawn mostly from Patterson and Hennessy [42, ch. 1, 4, 5 and 7], and Akhter and Roberts [1, ch. 1, 3 and 6].

2.1.1 Switch to MIMD processing paradigm

Historically software development was geared towards Single Instruction Single Data (SISD) processing architectures, where a single set of sequentially listed instructions is executed at a time and the appearance of task-level concurrency is left to the operating system scheduler, which is responsible for dividing processor time between many processes. For a long period this approach worked well; the growth in the number of transistors in processors and associated advancements in cache, hardware scheduling and execution logic meant that software executed faster without the need for any serious modifications. Processor clock speeds continued to increase dramatically throughout the 1980s and into the early 2000s (see Figure 2.1), driving an enormous growth in the capabilities of both desktop and server computers. However, since processor power usage is a function of the clock rate, the significant increase in processor clock rates also brought about increased power consumption and associated heat dispersion problems. Hardware manufacturers could no longer continue to drop the input voltage to processors to overcome

this power requirement and CPU clockrates stagnated. This brought about a sea change in the software industry: future software can no longer rely on significant improvements stemming solely from hardware improvements. Since 2002 the response time of programs has slowed from a 50% decrease to less than a 20% decrease per year (see Figure 2.2) [42].

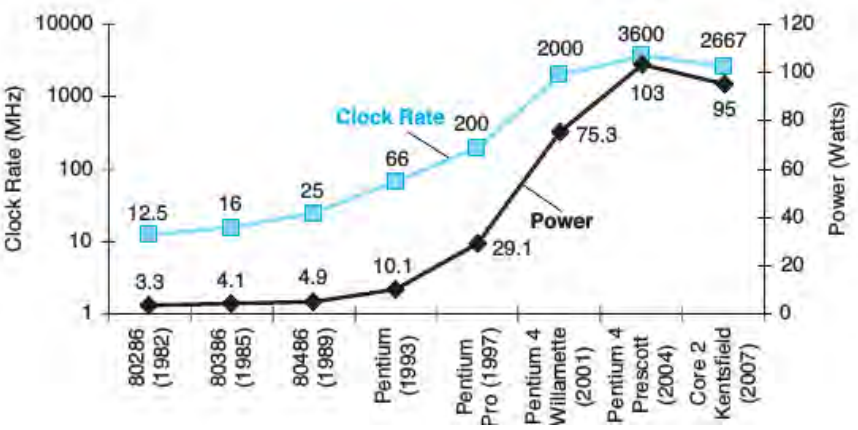


Figure 2.1: The growth in CPU clock rates for 8 generations of x86 Intel processors and associated power consumption. By reducing input voltages to the processors, engineers were able to keep power requirements low while increasing clock frequencies throughout the 1980s and 1990s. Limitations in semiconductor technology has, however, limited further increases in clock cycle rate. Taken from Patterson and Hennessy [42, ch. 1].

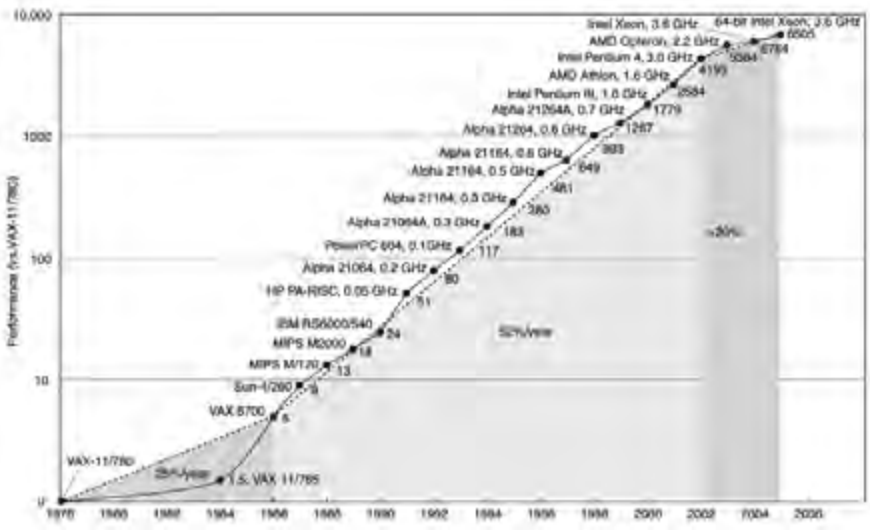


Figure 2.2: Here processor performance is plotted relative to the VAX-11/780 measured by the SPECint benchmarks. The substantial growth seen after the mid-1980s are primarily due to increasingly advanced processor architectures. Since 2002 the growth has been slowed to below 25% primarily due to the power wall and high memory latencies. Taken from Patterson and Hennessy [42, ch. 1].

Since the latter half of the 2000s most desktop processors subscribe to the Multiple Instruction Multiple Data (MIMD) paradigm, where several processes (or *threads*[1] of execution within a single process)

[1]Here a thread of execution is taken to be the basic unit of execution, with its own program counter and stack space, but sharing the address space with other threads started by the same process

are executed simultaneously within multiple *microprocessors* on the same processor die (also known as processing cores). See Figure 2.3 for an example of such a multi-core layout. Modern processors are thus geared towards attaining higher throughput by better exploiting the abundance of task-level parallelism in modern day computer usage, instead of increasing the response rate of individual processes. Whereas once only a select number of compute programs warranted nodes with multiple processors (physical chips) per compute node, today any desktop software has to be able to use multiple compute streams ("threads") in order to fully exploit the compute capacity of modern hardware [42] [1].

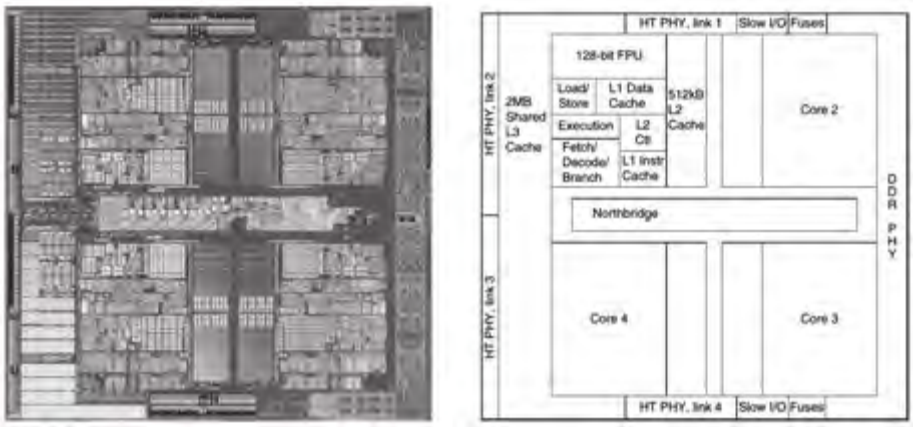


Figure 2.3: Microphotograph and layout of the quad-core AMD Barcelona generation processor. Taken from Patterson and Hennessy [42, ch. 1].

2.1.2 Fine- and course-grained hardware parallelism

Due to the shift towards the MIMD paradigm, most modern processors have both fine- and course-grained hardware parallelism. Well before the advent of multi-core processors, instruction-level parallelism existed. The original Pentium processor was able to pipeline instructions so that more than one instruction could be executed in a single clock cycle, by having two execution units. Keeping the execution units busy most of the time has since become one of the key focuses of pipeline architectures. As such, instructions without dependencies between them are no longer executed in sequence (starting with Pentium Pro processor [1995]). Later processors, such as the Pentium 4 (2002), have multiple units dealing with interrupt logic and processor state and are able to execute two logical threads concurrently per processor (and later processor core). This is known as *simultaneous multithreading* [22, 1]. See Figure 2.4 for an overview of the latest Intel Skylake generation microarchitecture.

These advanced multiple-issue out-of-order execution engines parallelize instruction execution at a hardware level, hidden from the application programmer. However, the ALUs of modern CPUs also support short SIMD vector operations. Figure 2.5 illustrates a typical SIMD computation. These instructions extend the original x86 instruction set with vector equivalents for many common operations. Originally the MMX instruction set allowed only integer arithmetic operations on packed byte, word and double-word registers and added eight additional 64-bit registers to the processor. Later extensions included various versions of Streaming SIMD Extensions (SSE) and Advanced Vector eXtensions (AVX), adding floating point and integer arithmetic on packed 128-bit and 256-bit registers, respectively. These SIMD vector operations improves the performance of tasks like 3D graphics and signal/image processing, that

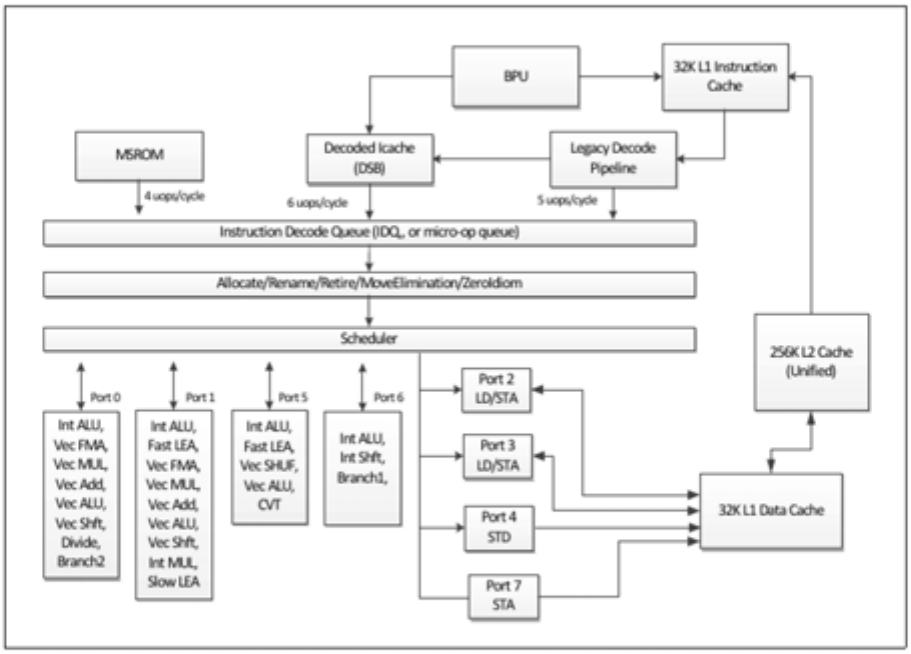


Figure 2.4: Architecture of the 6^{th} generation (Skylake) Intel Core processors (2015). As with many modern processors the front-end instruction fetch and decode components (apart from the branch prediction unit) and back-end instruction retire components are all in-order components. Each logical core has access to an out-of-order execution engine with many arthmetic units in order to pipeline, issue and execute multiple independent instructions simultaneously at hardware-level. Like most modern AMD and Intel processors, each core has access to a level 1 cache to store instructions and data, as well as a larger L2 cache (32KB and 256 KB, respectively, for Skylake processors). All of the cores share a large L3 cache (up to 2MB per core). Picture and specifications taken from the Intel Architecture Optimization Guide [15].

are inherently parallel, have localized recurring operations on data streams and have data-independent control flow. Most compilers (including the GNU compiler suite) vectorize code automatically when optimizations are enabled, but it may be necessary to write SIMD directives by hand in cases where the compiler fails to do so (deeply nested loops is one such situation) [15].

Both instruction-level parallelism and SIMD vector operations are fine-grained parallel processes handled by hardware. Since most modern processors have multiple processing cores, operating systems can extend beyond concurrent thread execution (where the execution of multiple threads are interleaved on physical hardware), to truly parallel execution of threads on multiple processor cores. One threading API that has found widespread cross-platform use is the OpenMP standard [4], which is currently implemented by the major Fortran, C and C++ compilers, including the Microsoft C++, Intel and GNU compiler suites. OpenMP provides parallelization options for both task- and data-parallel problems, is based on the lightweight fork-join threading pattern and takes care of thread instantiation and termination automatically. Most sections and loops can therefore be parallelized by simply specifying a single compiler directive in front of the block of code to be parallelized. OpenMP also support static and dynamic scheduling options that are useful in problems that require load-balancing [4, 1].

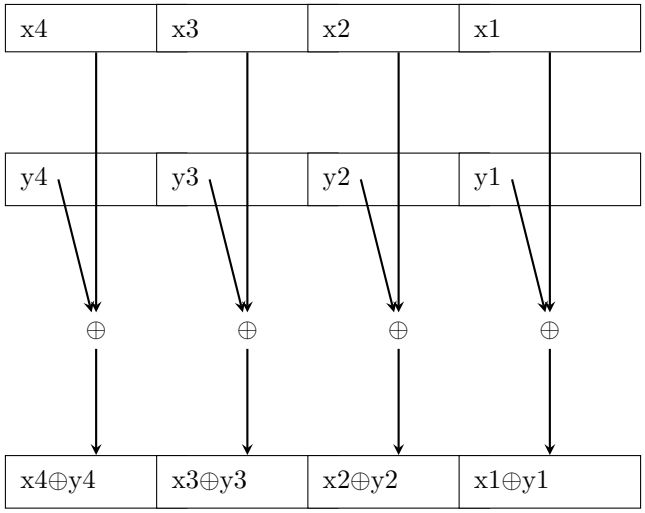


Figure 2.5: A typical Single Instruction Multiple Data (SIMD) vector operation. Here $\oplus$ is any binary arithmetic or logical operator. This diagram illustrates four operands packed into one of the special-length registers provided by the instruction set extensions. The SIMD operation then applies the same operation element-wise to all the packed operands simultaneously. Some variants on this instruction layout exist, but they all perform the same operation on multiple data elements packed into extended registers

2.2 Many-core GPU architectures

Much of the discussion in this chapter is drawn from Kirk and Hwu [33, ch. 1-3], Owens et al. [41] and the Compute Uniform Device Architecture (CUDA) programming reference [39]. Our presentation is focused towards implementation in NVIDIA CUDA. We start by giving the reader an historical overview and then highlight the key differences in architecture that sets GPUs apart from CPUs.

2.2.1 Historical development

Today's modern programmable GPU devices have evolved from the fixed-pipeline graphics hardware of the 1980's and 1990's. Driven by the demand for high resolution graphics in the video game industry, modern devices must be able to render billions of pixels per second (72 giga pixels per second for the latest Maxwell GTX980 devices [17]). From inception these devices favored achieving high throughput over higher operational latencies compared to traditional CPU-based computing. The steps involved with transforming primitives (typically triangles) in world space to rasterized images rendered by a display take thousands of compute cycles from start to finish. However, the coordinate, lighting and per-pixel shading operations are data-parallel operations. In addition the stages within the graphics pipeline can be computed in parallel; while new primitives enter the pipeline the rasterization and fragment processing of primitives previously transformed is completed, enabling both data- and task-parallelism on GPUs. The major steps in the graphics pipeline are shown in Figure 2.6 for reference.

The requirement of transforming, rasterizing and shading possibly millions of triangles clearly diverges from the requirements behind the design of traditional CPUs. CPU development is driven by the need to process large sequential programs per CPU core, each containing complex branching and diverse memory access patterns, whereas GPU development is driven by the need to apply the same set of basic operations to many elements (or the Single Instruction Multiple Data [SIMD]) paradigm that graphics processing

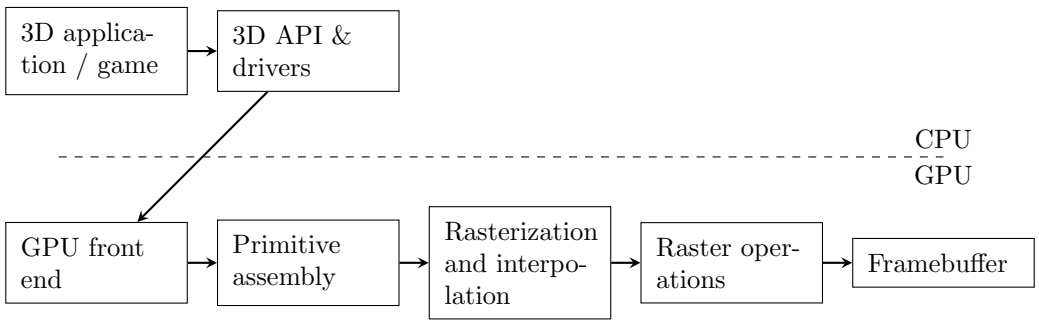


Figure 2.6: The classical graphics pipeline. At the front end of the pipeline verticies are transformed, colours assigned, texture coordinates and normals calculated. The rasterization step interpolate per-primitive data, such as colour, across all the pixels touched by the triangle formed between vertex triples. Raster operations perform final blending and anti-aliasing operations, before the images are written out to the display frame buffer.

subscribes to. Whereas CPUs use several tiers of large caches to hide the latencies of memory access, GPUs have relatively little on-chip cache memory per basic compute ("Streaming Processor" [SP]) unit. Instead of caching, GPUs rely mostly on the amount of parallel work available to each SP unit; a fast context switching / work scheduling mechanism ensures that each of the SPs are occupied with work while memory transactions are completed for threads stalled by load and store operations.

Although the GPU hardware platform could potentially be employed for applications other than 3D graphics it was not until the early 2000's that programmable graphics hardware became widely available. For instance, the NVIDIA GeForce 3 exposed the internal instruction set of the vertex processor to the application developer. Soon the ATI Radeon 9700 and GeForce FX made the fragment shading process, normally part of the rasterization and interpolation step, reprogrammable. At this point the vertex and fragment shading units were run on separate hardware. The XBox 360 (2005) introduced an early unified vertex and fragment shader processor.

Even though the hardware now supported extending the traditional graphics pipeline to more complex operations, it was still impractical to use the highly parallel hardware for computational processes other than small research projects. Computational problems had to be mapped to the standard graphics operands: verticies and textures. Other problems such as the lack of scattered memory writes limited the applicability of the platform to a small number of problems.

This was addressed by a series of software and hardware developments that aimed to give application developers access to the processors without having to call on Graphics APIs. Of those developments, BrookGPU [7] was an early abstraction away from graphics primitives. It recast computation in terms of small programs ("kernels") operating on "streams" of input data elements (arrays of values that can be operated on in parallel). This paradigm sets GPUs apart from ordinary vector processors that load a series of values from global memory, perform a simple mathematical operation on each of the elements and write the results back to global memory. Instead stream processors can load values from local register memory, performing multiple operations on each of these values before storing the results (possibly to local memory). The paradigm allows for greater arithmetic intensity (a single memory operation is followed by many computations) and is critical to hiding the high latencies of memory accesses experienced on GPUs.

More recently, NVIDIA CUDA [39] and the OpenCL standard [26] have become widely used in program-

ming GPUs to perform general scientific computing in a variety of fields, including the signal processing domain. Both lend themselves to the streaming processor paradigm and facilitate the implementation of the following generally-used parallel computing primitives:

- Scatter/gather: The addresses used in memory accesses (both load and store) can be computed.
- Map: an operation is applied to every element in the stream. Typically, many threads will be launched each reading an element from the stream, performing the operation on that element and writing the value back to memory afterwards.
- Reduce: By applying a binary associative operation repeatedly, an array of values is reduced to a single value. Examples include ordinary summation, minimums/maximums, variance, etc. These operations typically split the data into subsets, performing many operations in parallel and repeating the process on the set of results until a single value is obtained.
- Scans: Scan and prefix scan operations are widely used in parallel programming (for instance index calculations, as used in our work). In the case of addition, a scan of an array produces a new array containing the running accumulations of all the elements in the input array up to the index being computed, ie. $\text{Scan}(A)[i] = A[0] \oplus A[1] \oplus ... \oplus A[i]$, for any binary associative operator $\oplus$.

2.2.2 Modern programmable GPU architecture

The difference in design philosophy is reflected in the substantial differences in the architectures of modern GPUs and CPUs. CPUs dedicate significant die area to large caches and complex control logic dealing with branch prediction and scheduling. GPUs on the other hand dedicate more die area to arithmetic and other execution units and rely on having enough arithmetic work to occupy most of those units most of the time. Figure 2.7 illustrates the proportions of both CPU and GPU die area spent on arithmetic.

Figure 2.7: The proportion of transistors allocated to arithmetic processing in CPUs and GPUs. Taken from the NVIDIA programming reference [39]

Modern GPUs works best in problem contexts where a significant proportion of the computation is data-parallel. In CUDA nomenclature (similar concepts exist in OpenCL) the parallel work is broken up into a *grid* of separate thread *blocks* (see Figure 2.8 for an illustration). In each of these blocks, on-chip memory resources are shared, limiting the size of the individual blocks. On current NVIDIA GPUs there is a hard limit (1024) for the number of threads within a block. However, the amount of special memory (including register memory) available to each of these blocks may further limit the number threads in

the block that can physically execute simultaneously. Using streaming processor terminology each of the blocks would therefore process a portion of stream memory. Blocks are in turn subdivided into *warps* of threads (currently 32 threads form a warp) that execute instructions in lockstep. This means that when one thread inside a warp is stalled (for instance, for memory access or at a synchronization barrier) the entire warp is stalled. Unlike the instruction scheduling system of CPUs the schedulers in GPUs do not contain complex branch prediction logic; when some threads in a warp require the execution of one of the directions in a branch, while the rest take another direction, both sides of the branch are evaluated and the results are simply masked out for those threads that are unaffected by the branch direction. The requirement of lockstep execution places a hefty penalty on branch divergence within the instruction kernels, as well as memory accesses that do not adhere to the alignment specifications of the GPU. Note that this description of how work is specified in a high level programming language is independent of specifics of the device the work is to be processed on. The individual blocks can be mapped onto the targeted device in any order and in any quantity, depending on the resources of the device. Each block of work should therefore perform its computation in isolation of the remaining blocks. Although intra-block communication between threads and synchronization is possible, inter-block communication is only possible through accesses to off-chip memory and should be avoided if at all possible

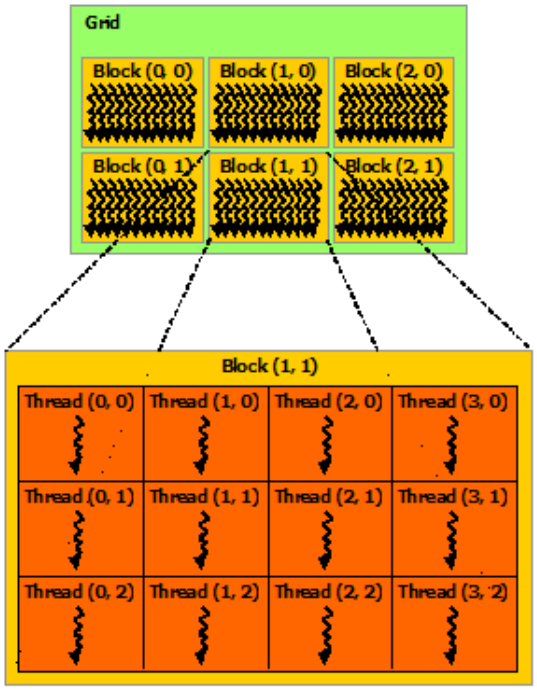


Figure 2.8: The layout of work threads in CUDA. Taken from the NVIDIA programming reference [39]

At a hardware level, GPUs comprise multiple *Streaming Multiprocessors*, each containing many *Stream Processors* capable of performing arithmetic operations (predominantly IEEE 754 single precisision floating point) along with several special function units, warp-schedulers and memory load/store units. Figure 2.9 shows the layout of Kepler-generation NVIDIA GPUs. The exact number of Streaming Multiprocessors and Stream Processors varies between generations of GPUs, but the total number of Stream Processors per GPU tend to double every 2 years. Depending on the resource constraints of each of the thread blocks several blocks may be mapped to a single Streaming Multiprocessor for simultaneous

execution. The warp schedulers schedule warps that have instructions (along with the required operand data) ready for execution onto sets of Stream Processors, dispatching multiple independent instructions onto individual Stream Processors per clock cycle (depending on the number of dispatch units available per Multiprocessor). Warps that are stalled (for example waiting on a load/store operation) are switched out of context and replaced with warps that have operands ready for processing, thereby hiding memory access latencies. Ideal kernels should therefore:

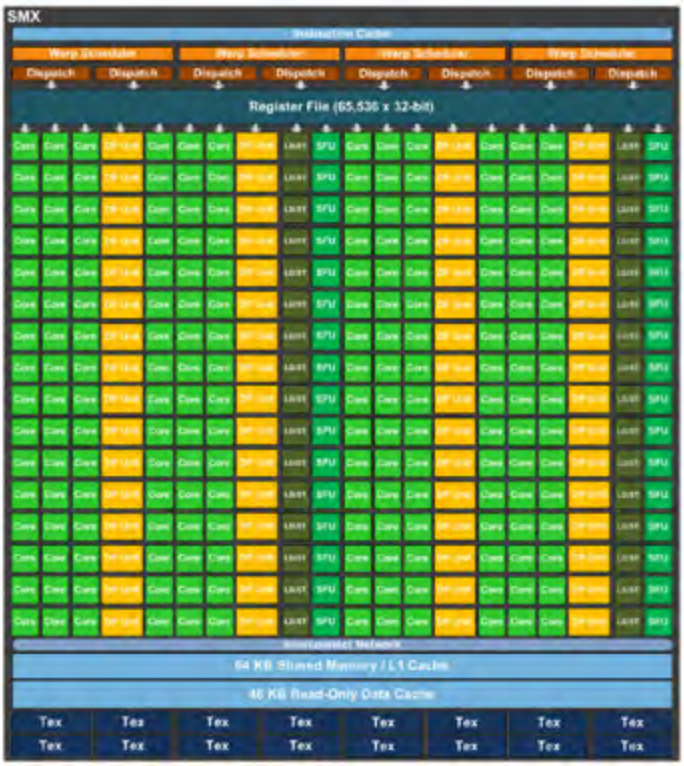
- Contain sufficient independent arithmetic instructions to occupy all the Stream Processors during any given clock cycle.
- Access to both on- and especially off-chip memory should be kept to a minimum and subscribe to coalesced access patterns, especially considering that the number of load/store units are far fewer than the number of single precision units and peak memory bandwidth is on the order of 60x slower (Kepler generation) than the peak single precision compute throughput of the device. In other words, the arithmetic intensity should be high, as is the case in typical graphics shading operations, for instance.
- There must be enough warps of work scheduled to the GPU to keep the Multiprocessors occupied most of the time.
- Special memory resources (including registers) should be used sparingly to ensure the Streaming Multiprocessor is not starved of resources, as this lowers the number of warps that can be executed at any given point in time ("effective occupancy").

2.2.3 GPU memory layout

The GPU memory hirarchy is substantially different from those found on CPUs. As was pointed out earlier, the total cache memory per Multiprocessor is divided between many Stream Processors. The following special memory systems exist on the GPU:

- Shared instruction cache. GPUs are SIMD devices by nature and thus require the same set of instructions to be executed by many Stream Processors. This means that the instruction cache can be shared between many Stream Processors, unlike with CPUs where each core has its own instruction cache.
- Local memory: Each Stream Processor has access to its own private local memory space in which register memory resides. The total number of registers is divided amongst all threads and is determined by the maximum number of registers needed to execute the instructions contained in a kernel, setting an upper limit on how many threads can be executed on the Multiprocessor at any point in time.
- L1 data cache: The L1 cache on GPUs is split into a local data cache and a shared memory cache. On GPUs the local cache stores register spills from local memory. Depending on the hardware generation, L1 memory also caches accesses to global memory (2.x by default), but this is not necessarily supported by all devices. Some 3.x Kepler devices can opt in to this behavior by specifying compile time options.
- Shared memory: As suggested shared memory is a cache memory shared between all threads executing in a block. The split of the L1 cache into shared and data caches is reconfigurable at runtime. It is also important to note that the exact amount of shared memory requested by each kernel

(a)



(b)

Figure 2.9: Kepler die architecture. (a) shows the overall die layout, containing 15 Streaming Multiprocessors. (b) shows the layout of each multiprocessor, containing one double precision unit (yellow) for every 3 single precision units (green), 1 load/store and 1 special function unit for every 6 single precision units. The total number of registers, local and shared memory is split between the number of threads per block, determining how many blocks can be executed simultaneously. Taken from the Kepler whitepaper [16]

at launch sets the limit on how many blocks can be executed simultaneously. Shared memory is used for both communication between threads as well as storing values that are commonly accessed

(and/or modified) by multiple threads. Importantly for performance, however, shared memory is divided in banks; each consecutive word falls into a different bank. An out-of-sequential-order or strided reads between two or more threads can result in accesses to the same bank simultaneously, generating a bank conflict. Special rules apply for sub-word and multiple-word accesses, for which the reader can refer to the CUDA API [39, Section G: Compute Capabilities].

- Constant cache: Constant memory is read-only memory that resides in off-chip memory, but is cached on chip. CUDA uses this mechanism to broadcast a single read to a half-warp of threads thereby saving 15/16 memory accesses that would be encountered when the same read pattern is made to global memory without this mechanism. Consecutive accesses to the same memory encounter no extra cost.

- Read-only data (texture) cache: In graphics processing one of the most common operations performed by the GPU is to map textures onto triangle primitives. Memory reads from textures (stored off-chip) are highly regular and spatially coherent, meaning that a group of work units will likely read values from the same area of texture memory. The caching mechanism is designed to optimize this access pattern. The texture cache residing in each Multiprocessor is not kept up to date with changes made to textures in global memory and can become stale. Loads from texture memory can also be made with hardware-based interpolation between neighboring values enabled. As noted later in the discussions on implementation we use this memory to store precomputed filter values.

- Global memory: Global memory resides off chip, just like constant and texture memory. Global memory accesses are cached in the crossbar L2 cache that is shared between Multiprocessors and has a cache-line length of 32-bytes. In Fermi, accesses were further cached in the L1 data cache by default, where the cache lines were 128 bytes in length. With some compute 3.x hardware this caching option can be enabled. Some read-only accesses are cached in the read-only data cache in compute 3.x devices. Warp memory accesses (not essentially ordered on an intra-warp level) that are aligned with these boundaries maximizes available memory bandwidth.

Table 2.1 outlines some of the performance and capacity of these different memories for reference.

Tesla card	M2075	M2090	K10	K20	K20X
32-bit register file / multiprocessor	32768	32768	65536	65536	65536
L1 cache + shared memory size	64 KB.	64 KB.	64 KB.	64 KB.	64 KB.
Width of 32 shared memory banks	32 bits	32 bits	64 bits	64 bits	64 bits
SRAM clock frequency (same as GPU)	575 MHz	650 MHz	745 MHz	706 MHz	732 MHz
L1 and shared memory bandwidth	73.6 GB/s.	83.2 GB/s.	190.7 GB/s	180.7 GB/s	187.3 GB/s
L2 cache size	768 KB.	768 KB.	768 KB.	1.25 MB.	1.5 MB.
L2 cache bandwidth (bytes per cycle)	384	384	512	1024	1024
L2 on atomic ops. (shared address)	1/9 per clk	1/9 per clk	1 per clk	1 per clk	1 per clk
L2 on atomic ops. (indep. address)	24 per clk	24 per clk	64 per clk	64 per clk	64 per clk
DRAM memory width	384 bits	384 bits	256 bits	320 bits	384 bits
DRAM memory clock (MHz)	2x 1500	2x 1850	2x 2500	2x 2600	2x 2600
DRAM bandwidth (GB/s, ECC off)	144	177	160 (x2)	208	250
DRAM generation	GDDR5	GDDR5	GDDR5	GDDR5	GDDR5
DRAM memory size in Gigabytes	6	6	4 (x2)	5	6

Table 2.1: This table summarizes the features for several generations GPU hardware. Taken from a talk by Manual Ujaldon presented at the Department of Computer Science, UCT in 2013.

Chapter 3

The Radio Interferometric Measurement Equation

In this chapter we give the reader a "grand tour" [1] of how radio telescopes make measurements of the radio sky. A mathematical model known as the Radio Interferometric Measurement Equation that relates the radio sky to these measurements is derived. This chapter aims to gradually build up an understanding of the data products that are measured, defines the necessary coordinate systems used in the observation and how modern aperture synthesis telescopes are used in synthesis imaging observations. In the next chapter we will discuss the inversion of this model and some of the complications that arise when doing so.

3.1 The radio universe

Just as with visible light, radio waves are a form of electromagnetic radiation (consisting of waves with an electrical and perpendicular magnetic component), which propagates through free space at the speed of light. Recall from elementary physics that the frequency and wavelength of a wave are related:

$$\nu = \frac{v}{\lambda}$$

[1] It is important to stress that radio astronomy is a cross-section of many disciplines including astronomy, physics, electrical engineering and, increasingly, high performance and distributed computing. The following texts provide further insight for those with a computing background (we recommend reading the first three texts in order before moving to synthesis imaging):

- *Antennas: Fundamentals, Design, Measurement* [3] serves as a good introductory text on general communications radio antenna design from an electrical engineering perspective. Chapters 1 through 4 are very insightful.
- *A Scientist and Engineer's guide to Digital Signals Processing* [53]. Available freely at `http://www.dspguide.com/`. A must-read introduction to core digital signals processing techniques, which cover sampling theory, introductory filter design and a good starter on the practical uses of Fourier transforms.
- *Radio telescopes* [10] gives insight into the historic development of radio telescopes from the 1930s through to the 1960s, with a focus on telescope design, interferometry, measurement and a good overview of the field of radio astronomy from an engineering perspective. The book is in the public domain and freely available from `https://archive.org/details/Radiotelescopes`.
- *Synthesis Imaging In Radio Astronomy II* [57]. A very useful (and beginner-friendly) collection of lectures on synthesis imaging, covering the domain of radio astronomical imaging in its entirety.
- *Interferometry and Synthesis in Radio Astronomy* [59] covers the imaging pipeline in great detail from correlation through calibration, cleaning and beyond. A very valuable reference.

where: ν is the frequency, λ is the wavelength and v is the velocity of the propagation of the wave in some propagation medium.

In a vacuum and far away from obstacles (*free space*) electromagnetic waves propagate at a constant velocity, $c \approx 3\text{x}10^8\text{ms}^{-1}$. This velocity is only slightly reduced when propagating though most other media. We can conveniently measure these wavefronts with telescopes of various form at either ground-level or from planetary orbit.

Generally speaking, black bodies (sources that near-perfectly absorb all incoming electromagnetic radiation) will radiate this energy over a very wide band of the electromagnetic spectrum. See Figure 3.1.

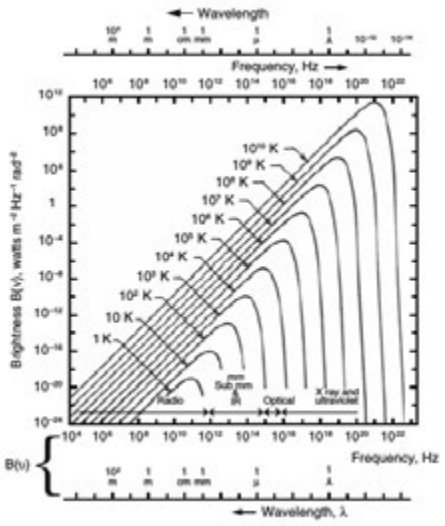


Figure 3.1: The radiation intensity distribution for a black body in equilibrium. The intensity distribution of black-body emission depends on both the equilibrium temperature of the body and the frequency of emission alluded to earlier. A black-body in equilibrium will, on average, have an emission intensity distribution that roughly follows the Plank distribution. Taken from *Tools of Radio Astronomy* [64]

Large bodies like planets and stars are generally considered to be of this type. Black bodies have to be very hot in order to be observed in radio if they are far outside our solar system. One would expect the radio sky to be quite empty if this was the only source of radio emission in nature. This placed a damper on radio observation until Karl Jansky's observation of radio electromagnetic radiation stemming from the centre of the galaxy in the early 1930s sparked renewed interest in this region of the spectrum.

We know now that electromagnetic energy can be emitted by both thermal and non-thermal sources. These thermal sources not only include black bodies, but can include, for example ionized gasses such as ionized hydrogen. On the other hand, a good example of non-thermal emission is the synchrotron emission by electrons accelerated radially through magnetic fields and the HI absorption line.

One would expect radio waves to have the same optical properties as visible light, since they too are a form of electromagnetic radiation. These are, respectively, reflection, refraction (bending as these waves propagate through media of different densities), diffraction and interference. The last two phenomena are of particular importance in our discussion on radio telescopes and can only be described using physical optics. Interference can either be constructive or destructive in nature. If the incoming waves are

perfectly in phase (their crests line up perfectly) the resulting wave will have the combined amplitude of the contributing waves. However, if they are out of phase the resulting wave may have significantly reduced amplitude. As for diffraction, Huygens' principle states that each point on an incoming wavefront (either planar or curved) acts as a point source on its own. The secondary waves produced by each of these point sources propagates forward radially and a new wavefront is formed where they experience maximum constructive interference. This explains why even planar waves can seemingly "bend" around obstacles.

In free space the total energy along the wavefront is conserved as it propagates forward, provided the wavefront is of reasonable extent (significantly longer than a wavelength). This also means that the energy density on each of these wavefronts will decay at a rate proportional to the square of the distance between the front and its emitting source. Hence the emitting source should be sufficiently far away from the observing telescope that the wavefront remains approximately planar over the entire area of the telescope.

When these waves propagate through some medium other than a vacuum the decrease in directional energy is not the only form of attenuation. In an atmosphere, depending on the wavelength, some particles such as oxygen and water vapor will absorb and scatter a significant portion of the incoming energy (especially at shorter wavelengths). At very long wavelengths the charged ionosphere is effectively opaque and acts as a good reflector. This may be ideal when trying to transmit signals very far beyond the horizon, but makes astronomical observation at such wavelengths impossible. Due to these additional attenuation factors ground-based observation is effectively limited to the spectrum of visible light and the, vastly wider, radio band. Most of the infrared spectrum in-between is only observable at high altitudes and under dry conditions, see Figure 3.2.

In addition to the optical properties of electromagnetic waves and their attenuation one has to consider the direction of propagation of each point on the incoming wavefront. If most of the energy of these points is strongly directional the wave is said to be *polarized*. For polarized emissions the path traced by each point (of either the electrical or magnetic components) will be highly regular; it will remain in a single plane (*linear* polarization), will spiral at a fixed diameter (circular polarization) or will spiral elliptically.

We can draw on an application of this property from an everyday context: in the visible spectrum sunglasses will filter out all light except vertically polarized light, in order to reduce glare. A single-feed radio antenna will similarly measure a single directional component, and will therefore only be useful in measuring strongly polarized sources and the total power received by the antenna is at most half the power that would have been collected by perpendicular feeds (a dipole is a simple example). Additionally two feeds are desirable because the measurements they take fully describes the polarization of the incoming wavefront.

In the last century significant progress has been made towards increasing the sensitivity and size of these radio telescopes. Next, we will explore how a single-element telescope works, before moving onto the topic of aperture synthesis with array-based telescopes.

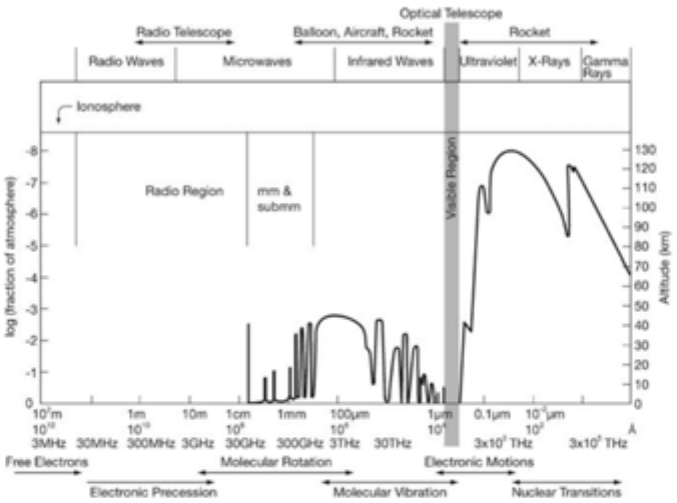


Figure 3.2: The radio window - Earth-based radio astronomy is bound to a range of wavelengths between $\lambda \approx 0.2mm$ and $\lambda \approx 20m$, by the molecular absorption bands of oxygen and water at shorter wavelengths and the ionosphere at longer wavelengths. The figure shows at what altitude the incoming electromagnetic radiation is attenuated by a factor of 0.5. Image taken from *Tools of Radio Astronomy* [64].

3.2 Single antenna telescopes

3.2.1 Overview

Maxwell's set of partial differential equations (1873) is one of the most elegant ways of explaining the relationship between electrical and magnetic fields, and how these propagate at the speed of light though free space. In summary they state that when current flows, a magnetic field is created in the surrounding space. When this magnetic field is varied an accompanying perpendicular electrical field is formed. This electrical field varies at the same frequency as the magnetic field, which in turn varies at the frequency at which the underlying current changes. Not only does this mean that antennae can generate electromagnetic fields and transmit signals, but in fact that any transmitting antenna can be used as a receiver and vice versa, assuming that it is capable of dealing with high voltages and is efficient enough for the particular application domain. Radio telescopes take the form of receivers and, as with terrestrial radio transmission, the extraterrestrial electromagnetic radiation will induce measurable current in the antenna.

To simplify our discussion we will only consider directional antennae (a simple parabolic reflector with an axial feed above the center of the parabola is one such choice). Here the parabolic reflector serves either to focus highly directional incoming energy to a single point, or conversely to focus the energy of the transmitter into a very narrow beam. Because of the wavelengths of radio waves geometrical optics, where waves can be considered as rays, are of very limited use. It is preferable to describe these antennae in terms of physical optics.

Consider, for the moment, that the telescope is suspended in free space with no obstructions in its vicinity (including the ground or supports). If we also discard the effects of the feed between this focus and measuring equipment, then the energy measured at the focus of the parabola should be the sum of contributions across the extent of the collected wavefront. However, instead of focusing all energy at

a single point, as one would expect when using geometrical optics, an interference pattern is formed. Here distinct beams are discernible (a close analogue to this is the interference pattern formed when light passes through a narrow slit). If a cutting plane were to be placed horizontally at the focus of the parabola a single "primary" beam of maximum constructive interference would be noticed, along with several minor lobes to either side of that primary beam. The lobes right next to the primary beam are appropriately termed "side" lobes (refer to Figure 3.3). A highly directional antenna limits the maximum amplitude of these lobes considerably. It is important to note that an isotropic antenna will not have a single primary lobe, but may have several main lobes of equal amplitude.

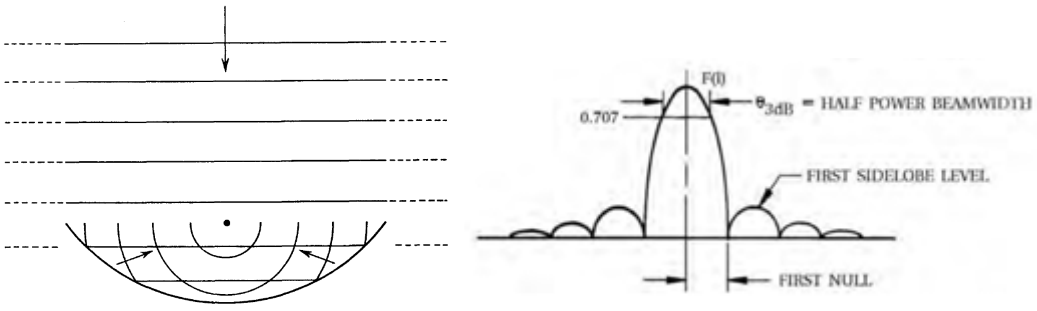


Figure 3.3: Here the wave pattern of a parabolic aperture is illustrated with a single cutting plane. The interference pattern created at the focus (illustrated on the right) is analogous to the interference pattern of visible light when passing through a single narrow slit. Directional antennae form a discernible primary lobe, with several minor lobes of decreasing magnitude to either side. The beam width of the primary lobe is usually defined as being at the level where the electric intensity is halved (around $1/\sqrt{2}$ of the maximum response). At approximately $\frac{\lambda}{D}$ the "first null" appears (D is the diameter of the aperture). This is the first point of complete destructive interference. When synthesizing an image of a source which extends beyond the primary beam, one will notice a severe drop in the intensity of its edges. Images taken from *Radiotelescopes* [10] and *Synthesis Imaging II* [57] (left and right, respectively).

When placing the antenna back into a more realistic context: relatively close to the ground and taking the resistance of the feed connecting the antenna to measuring equipment into account, this radiation pattern changes considerably to a so-called "absolute" beam pattern. As expected the ground and any large nearby object will act as a reflector. Although the intensity of the reflection varies depending on how level the surrounding terrain is and its conductivity (dry or wet conditions) it cannot simply be ignored. Because the reflected wave may be out of phase depending on the height of the antenna and its elevation angle, portions of the primary beam may experience significant destructive interference.

The gain, G, of the antenna is defined as the ratio of radiation intensity collected in a given direction (as when emitted by a highly directional source), to the intensity that would have been obtained when receiving radiation isotropically. This definition includes feed losses due to ordinary resistance and is obviously direction dependent. For our discussion on radio telescopes, however it is much more useful to think in terms of the effective area of the antenna. The effective area is defined in terms of the gain of the antenna as:

$$A_e = \frac{G\lambda^2}{4\pi}$$

Assuming absolutely no losses occur in the measured electric density and no phase errors are introduced, this effective area will approach the geometric area of telescope in the direction of maximum constructive interference. In practical applications this is never achieved. The effective collecting area of the telescope is limited by a variety of factors including, but not limited to, aperture surface deformation, blockage of the collecting area by support struts, and the radiating properties of the feed.

It is worth pointing out that these radiating patterns are measured in the far field of the antenna (at distances at least $\frac{2D^2}{\lambda}$ from the antenna where D is the diameter of the aperture) to avoid any near-field reactive effects. The gain of an antenna must therefore be seen as a far-field concept.

In addition to the effective collecting area just defined, all antennae are only considered effective for a limited spectrum of frequencies (or "bandwidth"). This may be either a narrow or a wide band of frequencies. Size is one of the factors governing effectiveness when it comes to bandwidth. Although there are no hard and fast rules about the size of antennae, a general rule of thumb is that antennae smaller than 0.5λ are considered electrically small and antennae more than about 100λ are electrically large, and are capable of attaining much higher gains in the case of directional antennae.

For astronomy purposes it is very important to have an antenna much larger than λ in order to increase both sensitivity and resolution. Although large objects may easily be resolved if they are smaller than the primary beam in the response pattern of the antenna, the resolution of finer details depends squarely on the angular resolution of the observing telescope:

$$\text{angular resolution} \propto \frac{\lambda}{D}$$

where D is the aperture diameter.

Small telescopes will smear finer detail, and point sources right next to each other may not be discernible. The angular resolution is an indication of the minimum distance at which two point sources can be separated and still be discernible.

3.2.2 Measurement

For the sake of discussion and to simplify the mathematics we introduce the following simplifying assumptions about radio emission:

1. Most radio sources emit their radiation outward uniformly in all directions (they are *isotropic*),
2. The emission from any two astronomical sources (or any two points on a single source) is incoherent.
3. That the distances over which these waves travel are far enough to consider them planar by the time they reach the observing telescope.

The energy available at the output terminals of a single aperture antenna will be roughly proportional to the electrical intensity per unit area per unit frequency on the collected planar wavefront. This is the electrical flux density, S, measured in units $\mathrm{Wm^{-2}Hz^{-1}}$.

A single antenna telescope measures its power over a limited bandwidth and effective area A_e, as explained earlier. Assuming no attenuation or delays are introduced by the atmosphere or equipment, and that the antenna measures two complementary polarizations, this means the total power measured by the telescope when pointed in the direction of a single point source is:

$$W = A_e.S.\Delta\nu, \text{ where } \Delta\nu \text{ is the observed bandwidth}$$

Using this total power relation it is easy to see the advantage of averaging several bands together to observe sources of *continuous emission*, although observation of spectral line emission (such as the

absorption line of abundant neutral hydrogen at 21cm, along with other common elements) is also very important when tracking the evolution of the universe.

Before deriving a more formal mathematical framework to describe the measurements taken of radio sources it is necessary to introduce a cartesian coordinate system centered on the focus of the antenna, tilted such that x and y are the orthogonal horizontal and vertical axis respectively, and z is orthogonal to both. The direction of a point on the sky dome (a fictitious unit sphere) is given by the spherical direction cosines, measured relative to this local frame. These cosines are denoted by l, m and n respectively (note that $l^2 + m^2 + n^2 = 1$), and can be measured in terms of the azimuth and elevation/zenithal angles to the source along the local horizon. Zenith is the position directly upwards from the telescope. Refer to Figure 3.4.

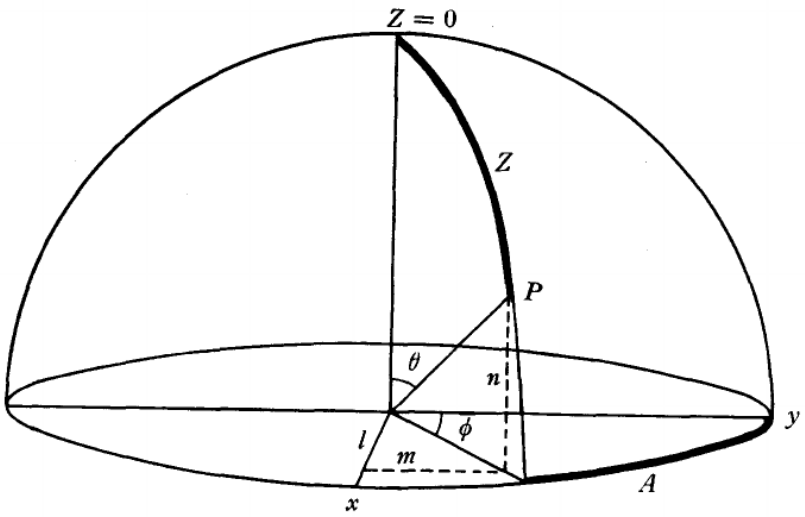


Figure 3.4: Source coordinates measured according to the local horizon. The sky dome is represented by a fictitious unit sphere called the celestial sphere in order to simplify the mathematics. The Azimuth angle is measured from the south while the zenith angle (directly above the observer) is measured along the great circle connecting the north-south meridian, the source and the horizon. Note that the arclengths are given by $Z = 1\theta$ and $A = 1\phi$. Figure taken from [10].

$$l = \sin Z \sin A$$
$$m = \sin Z \cos A$$
$$n = \cos Z$$

Now let Ω be a *small* solid angle subtended by a very small area. This solid angle is measured in square radians (steradians or "sr"). A visualization of the measured flux density falling on an infinitesimal area of telescope is given in Figure 3.5. The flux density measured per steradian over the entire solid angle per unit frequency must then be

$$p \approx \int_{\Omega} A(l,m).I(l,m) \cos\theta d\Omega \tag{3.1}$$

where
p is traditionally measured in terms of Janskys ($1\text{Jy} := 10^{-26}\text{Wm}^{-2}\text{Hz}^{-1}$) per steradian

I is the flux density measured per unit steradian: $I(l,m) := \frac{\Delta S(l,m)}{\Delta\Omega}$
A is the directional modulating beam of the antenna

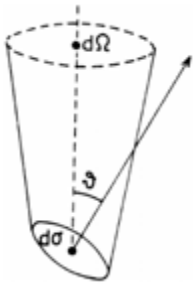


Figure 3.5: Flux density collected by a small area on the antenna. Image taken from *Tools of radio astronomy*[64].

In reality multiple sources will be contributing to this integral, each at a separate angle to the pointing center. The integral should therefore be taken over all contributing sources. It is assumed that these radiate independently of each other. Furthermore, the effects of the atmosphere, specifically frequency dependent delays and phase rotation, as well as tropospheric effects have been downplayed. The pointing accuracy and radiative properties of the antenna including its internal temperature, polarization leakage, have also been downplayed. Some of these effects can be corrected for through direct calibration and model-fitting techniques, but the exact details are beyond the scope of this introductory discussion. Typically calibration and preprocessing can be broken up into three areas:

- **Flagging**. This preprocessing is completed before imaging and used to mark broad portions of the data that is known to be invalid, due to, for example, technical malfunction or terrestrial radio interference. Modern radio astronomy software, for example the CASA reduction suite, have tools available for automatic flagging.
- **Calibration through known calibrator sources and a priori information**
- **Self-calibration processes** are iterative processes which typically involve model fitting algorithms and techniques.

For more details the reader is referred to Synthesis Imaging in Radio Astronomy II [57, Lectures 3, 5, 8 and 10]. In the measurement section of the array-based telescope observations we will refer to a general model, which may be used to describe both atmospheric and antenna terms and can be used for self-calibration processes.

3.3 Aperture synthesis with array telescopes

3.3.1 Overview

As pointed out in the previous section the angular resolution is constrained by the diameter of the telescope. For longer wavelengths very large directional antennae are needed to achieve good angular resolution. Unfortunately, there are material constraints, increased maintenance costs and difficulties in steering associated with large telescopes. Luckily, it is not essential to build a filled aperture in order

to create a reasonably efficient directional antenna. It is also possible to leave out large portions of the antenna aperture to create a directional antennae of significantly reduced weight (which is much cheaper to build). The obvious effect is a decrease in effective collecting area and therefore decreased sensitivity.

Array telescopes can be thought of as a special case of these unfilled apertures. The very simplest way of creating such a telescope is to add the signals from all the receivers together before measurement, in order to form a basic "total power" telescope. This is only possible if the signals are reasonably *coherent*. If they are significantly out of phase the signals will experience destructive interference and the telescope will be rendered useless. For the sake of discussion, however, it is assumed that the distances between antennae are fully accounted for: the wavefront collected at different locations will simultaneously arrive at the measurement equipment shortly thereafter, and the increased impedance associated with longer transmission lines is duly considered.

Using an array it is possible to "synthesize" a single aperture telescope that encompasses the entire array. This basic idea is known as *aperture synthesis*. It does, however, pose a serious conundrum: only some areas will be well covered, whereas large areas may not be covered at all. As we see later the Fourier Transform of this pattern is convolved into the synthesized images and is a topic of much discussion by itself.

The resolving power obtained from such a synthesized aperture depends on the distance between the furthest separated antennae, and can be expressed as:

$$\text{angular resolution} \propto \frac{\lambda}{B} \text{ if } D \ll B$$

where D is the diameter of the largest aperture in the array and B is the length of the longest *baseline* (the vector defined between the positions, P, of pairs of antennae p and q: $\vec{b}_{pq} = P_p - P_q$). λ is the observed wavelength.

Since Martin Ryle and his collaborators made their pioneering observations using array telescopes there has been a significant drive towards building arrays of ever-increasing size. In applications where angular resolution is the one of the dominant factors it is even possible to connect up telescopes located on different continents, or even orbiting satellites. This is appropriately called *Very Long Baseline Interferometry*. For more details refer to the survey conducted by Middelberg et al. [36] for an introductory overview of VLBI. In the next section we discuss what is meant by an interferometer and why array telescopes are referred to as radio interferometers

3.3.2 Measurement

It is also possible to think of arrays using a more physical model. These telescopes measure what is commonly referred to as the *spacial coherence* of a source. A simple analogy for how array antennae work from a physical perspective would be to think of a point disturbance in a bowl of water. If the amplitude is measured by two calibrated sensors at the same location, the readings obtained should be exactly equal at any point in time. As the sensors are spaced further and further away from each other, but remain equidistant to the source, one would expect the readings to still be the same - since the waves propagate outward at the same speed and with the same amplitude. The degree to which the two measurements correspond at any point in time will tell the observer how coherently the waves are

propagating outward. If one of the sensors is moved slightly further away from the disturbance, the delay between the measurement taken by the first and second sensors will tell the observer something about the position of the disturbance.

If this bowl of water analogy is scaled up to monstrous proportions, and the water is replaced with free space, it resembles an array telescope. Since the source is assumed to be sufficiently far away, each electromagnetic wave crest will be measured by two (or more) directional antennae at exactly the same time, provided the wavefront is parallel to the baseline vector between the two antennae and they point in the same direction. Just as with the bowl of water analogy if the source is slightly offset from the pointing centre of the telescope, the phase delay between the arrival of the wave front at two separate antennae will tell the observer something about its offset from the pointing centre. Refer to Figure 3.6 for an illustration.

Figure 3.6: In this simplified illustration of a two antenna array the incoming wavefront being measured is produced by a single source in the direction of $\vec{s}$, offset at an angle θ from the pointing center,$\vec{s_0}$, of the telescope. Here, it is assumed that the pointing center is also the center of maximum response and where the delay between incoming signals is exactly zero. The *correlator* measures the degree to which the signals measured at the two antennae correspond (both in phase and amplitude), by measuring the degree of spacial coherence of the incoming wavefront. For now we can assume the entire radio interferometer is on level ground, and importantly that the measured wavefronts are planar. As already mentioned the time delay, τ between the arrival of the wavefront between at the two antennae corresponds to a phase delay in the measured signal. c is the speed of light.

These differences in phase, which depend on the angle of the incoming wavefront with respect to the pointing center, produces an interference phase pattern that is analogous to Young's double slit experiment, where an interference pattern is formed on a screen when light passes through two small slits (separated by a distance L) somewhere in front of the screen. Here the size of the slits are small in comparison to both λ and L. The first null point occurs at a distance proportional to $\frac{\lambda}{L}$. The only major difference in context is that the antennae should be viewed as narrow slits. Waves from directions $\theta = n\pi, n \in \mathbb{Z}$ will undergo constructive interference, but those from directions $\theta = (n + \frac{1}{2})\pi$ will be invisible to the interferometer. Such an interference pattern is not ideal when resolving large structures.

One solution is to take an additional delayed measurement at a phase delay of $\tau_{delay} = \frac{\pi}{2}$ radians and combine this "sine interference pattern" with the original "cosine interference pattern". In theory this improves the signal to noise ratio by a factor of roughly $\sqrt{2}$ and will produce a response pattern that is sensitive over a wider range of angles. Hereafter whenever referring to the correlator we will be considering the *complex* correlator. It is convenient to think of the measurements made by such a complex correlator in terms of the polar form of complex numbers. Recall Euler's identity:

$$e^{i\theta} = \cos\theta + i\sin\theta$$

As the reader may suspect the mathematical treatment of the measurements taken by these array telescopes is analogous to that of a single aperture telescope, except that in the single antenna case the telescope measures the cosine modulated flux density directly. Here, instead, the signals are combined and measured by the external correlator. The mathematical discussion is very similar to that of a single antenna telescope, but will be that of a complex measurement, instead of the real domain.

Up to this point it has been assumed that the signal will not vary over time, and that the measurement equipment is perfect; it does not delay or attenuate the collected signal (commonly referred to as the *instrumental gain*). In reality, however, such perfection is never attained. The sensitivity of the complex interferometer depends firmly on the total collecting area, the resistivity (and associated temperature) of the equipment, including the antenna and amplifying electronics (T_{sys}), the bandwidth being integrated and the integration time:

$$\sigma_{\text{noise}} \propto \frac{T_{\text{sys}}}{\sqrt{\Delta\nu\Delta\tau_{\text{integration}}}} \tag{3.2}$$

At a physical level, sources do vary slightly over time, although we expect their average intensity to be centered around some mean. The instrumentation used for measuring will also contribute some variation in noise level that has be corrected for on a regular basis. With this in mind it is important to make yet another assumption about the physical processes being measured: sources *do not* vary significantly from their mean intensity on a day to day basis. The obvious exceptions to this are the fast transient sources like pulsars. These sources must be observed over very short periods of time, and by implication must be relatively bright. Fainter sources have to be observed for much longer to be discernible from noise. This brings up the problem of tracking stellar sources.

For this discussion refer to Figure 3.7 for a visual reference. Due to the rotation axis of the earth sources on the celestial sphere will appear to move both in azimuth and elevation, steadily moving from east to west and completing a full rotation in just under 24 hours. In order to track sources over a prolonged period it is very useful to convert their coordinates to a reference frame where the path of a source is determined only by an hour angle. However, for this to work astronomers need a coordinate system that measures the Earth's rotation with respect to the stars and not the sun.

It may not be immediately apparent that the rising and setting positions of the sun with respect to the local horizon do not remain fixed throughout the duration of a year. If one were to point a camera due east to monitor the sunrise over the course of one year, the rising position would appear to oscillate around true east. Only at two points during the year will the sun rise due east - these are known as the spring and autumn equinoxes. At these points the plane containing the earth's equator intersects the sun. This is the reason why the sun appears to move along different star constellations throughout the year.

Instead astronomers use the local sidereal time as the hour angle to a source. A sidereal day is slightly shorter than the solar day and the reference point for the angle is the vernal equinox - which remains valid on the scale of decades. In this reference frame the equator of the earth is projected out onto the unit celestial sphere mentioned previously. The declination of a source is the distance along the great circle connecting the North Celestial Pole (NCP, the projection of the magnetic north pole onto the unit sphere), the source and the projected equator as shown in Figure 3.7.

Next we need to amend the local coordinate system defined previously for a single antenna telescope to be relative to some fixed location in order to derive a mathematical model for an interferometer. All the antennae in an array will be measured relative to this new coordinate frame. We assume a single antenna is picked as the reference point and all the other antennae positions are measured relative to this "reference antenna".

This local Euclidian frame has the regular components X, Y, Z that have its origin at any arbitrary

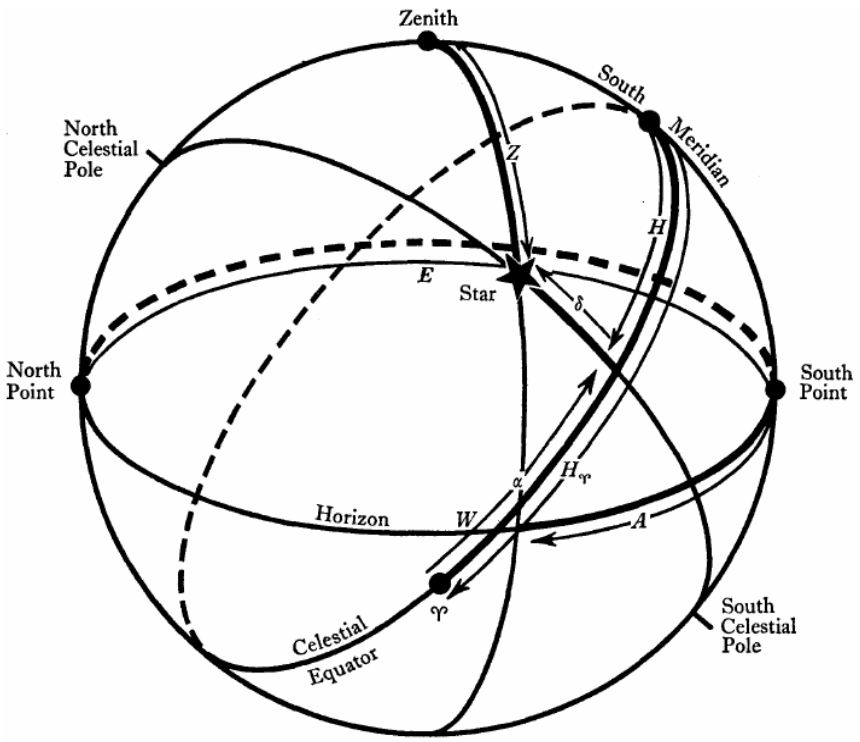


Figure 3.7: The angular coordinates of a point in the sky can be measured in terms of declination (δ) and either right-ascension (α or R.A.) or the local sidereal time (hour angle) at the position where observations are taken. The first point of Aries (Υ) is the position of sunrise at the March Equinox. The celestial equator is simply Earth's equator projected onto the celestial sphere. There is normally a slow precession in the Earth's axis and this has to be corrected for, based on some reference point in time (an epoch). The last epoch at the time of writing was at noon January 1, 2000. Zenith here refers to the point exactly above the observer's head. See Radiotelescopes [10, Appendix 4] for more information.

point, usually one of the antennas. If we use a right-handed system where X points in the direction of the 0^h hour hand, at a declination of $0°$, Y lies on the same plane as X, but points to 18^h and Z points straight up towards the North Celestial Pole (NCP), then the baseline is given by the following relation, where H_0 and δ_0 are the respective hour angle and declination of the phase reference center and λ is the wavelength corresponding to the center frequency of the receiver. Here u, v, w are given by:

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} \sin H_0 & \cos H_0 & 0 \\ -\sin\delta_0 \cos H_0 & \sin\delta_0 \sin H_0 & \cos\delta_0 \\ \cos\delta_0 \cos H_0 & -\cos\delta_0 \sin H_0 & \sin\delta_0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{3.3}$$

The direction cosines towards a point on the celestial sphere are still given in relation to this relative u, v, w coordinate frame. A similar left-handed convention can just as easily be defined (α_0 is the right ascension of the pointing center):

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \frac{1}{\lambda} \begin{bmatrix} -\sin\alpha_0 & \cos\alpha_0 & 0 \\ -\sin\delta_0 \cos\alpha_0 & -\sin\delta_0 \sin\alpha_0 & \cos\delta_0 \\ \cos\delta_0 \cos\alpha_0 & \cos\delta_0 \sin\alpha_0 & \sin\delta_0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{3.4}$$

Ignoring the effects of polarization, the directional antenna gain (the beam, as before) and any atmo-

spheric effects for the moment, the correlator will measure power values, P, proportional to:

$$P(u,v,w) \approx \left\langle \int_{sources} \int_{sources} E_p(\vec{s}) E_q^*(\vec{s'}) e^{2\pi i \vec{p}\cdot\vec{s}} e^{-2\pi i \vec{q}\cdot\vec{s'}} d\Omega_p d\Omega_q \right\rangle$$

Here E is the flux density over a small area (stemming directly from equation 3.1). The assumption that two sources at differing locations are not strongly correlated (in other words spatially coherent) and that the sources are very far away is essential in order to simplify this relationship. Still assuming both source direction vectors ($\vec{s}$ and $\vec{s'}$ from antennas p and q respectively) are unit vectors pointing to positions on the unit celestial sphere, this assumption implies that $< S_p(\vec{s}) S_q^*(\vec{s'}) > \neq 0$ only when $\vec{s} = \vec{s'}$. Also note that, since complex fluxes are being measured by the correlator, it is necessary to take short time averages of the complex conjugate (denoted *) of one of the antennae in order to measure the total flux contributed by both components - otherwise the difference between them will be measured!

$$P(u,v,w) \approx \int_{sources} \left\langle E_p(\vec{s}) E_q^*(\vec{s}) \right\rangle e^{-2\pi i (\vec{q}-\vec{p})\cdot\vec{s}} d\Omega$$
$$P(u,v,w) \approx \int_{sources} B(\vec{s}) e^{-2\pi i (\vec{b}\cdot(\vec{s}))} d\Omega$$

Notice that the expression for p is relative and depends only on the direction of each contributing source. Of course this is all relative to the pointing centre of the telescope. We may multiply through by a complex exponential to add the appropriate delay, which ensures measurements are taken relative to the pointing center. It may help to think of this as measuring the incoming (complex) flux from direction θ as shown in Figure 3.6:

$$\begin{aligned} P(u,v,w) &\approx \int_{sources} B(\vec{s}) e^{-2\pi i (\vec{b}\cdot(\vec{s}-\vec{s_0}))} d\Omega \\ &\approx \int\int_{sources} B(l,m,n) e^{-2\pi i (ul+vm+w(\sqrt{1-l^2-m^2}-1))} \frac{dldm}{\sqrt{1-l^2-m^2}} \end{aligned} \tag{3.5}$$

By no means should this model be taken as a rigorous derivation, but we will use it nonetheless (and slightly amend it along the way). A more rigorous derivation is provided by Romney in [57][ch. 4].

In reality averaging the flux over a small band of frequencies and, additionally, over short periods of time will result in smeared measurements. It is important to emphasize that the baselines between pairs of antennae are *always* measured in terms of wavelength (and by relation frequency) as evident from Equation 3.3. If one were to integrate over such limited bandwidth a problem becomes apparent: the correlator response is modulated by $\Delta\nu \operatorname{sinc} \pi\Delta\nu\tau$ (sinc is defined as $\operatorname{sinc} x = \frac{\sin x}{x}$). As apparent from Figure 3.6, τ depends on the orientation of the baseline with respect to a source, as well as the length of the baseline. The correlator response will be maximum only when the source vector is normal to the baseline. Therefore the correlator must continuously compensate for this delay at the tracking centre, as well as continuously shift the fringe modulation function to the phase centre of the observed field.

Up to this point we have, for simplicity, only discussed correlator output between single feed antennae. Just as in the case of a single antenna telescope, the response from such a correlator will only be sensitive to highly directional sources. The correlator may therefore additionally correlate all four combinations between the two orthogonal feeds of each antenna to form a 2x2 matrix of short time averages. Without any loss of generality the flux density average (B) in equation 3.5 can be replaced by four averages. The exponential term then becomes a scalar 2x2 matrix.

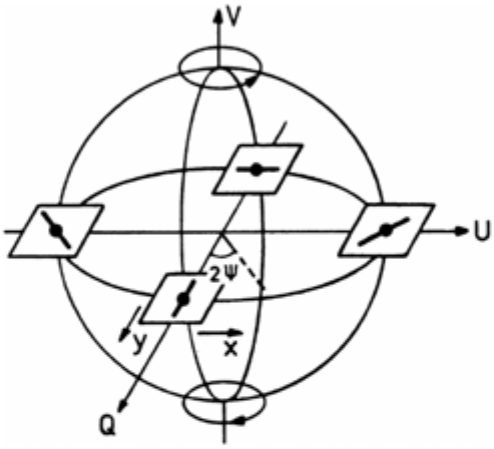


Figure 3.8: The Poincaré Sphere gives a visualization of the different polarizations in an electromagnetic field. The angles 2ψ (QU plane) and 2χ (QV plane) are the angles in a polar coordinate system, with each point on the sphere corresponding to a unique polarization. At $2\chi = 0$ (the equator) the polarizations are either linear (Q) or orthogonal (U). The northern latitudes ($2\chi > 0$) contain right-handed circular polarization, while the southern hemisphere contains the left-handed circular polarizations. I is not linearly independent and describes the total flux of the electromagnetic wave: $I = E_1^2 + E_2^2$ [64]

The four Stokes parameters are an easy way to describe the polarization of the measured electromagnetic field. These are non-physical quantities, but they conveniently express the degree to which the field is linearly or circularly polarized (and anything in between). Each of the parameters (Q, U and V) are linearly independent and describe a position on the Poincaré Sphere (figure 3.8). I is the total measured intensity. The coordinates of Q, U and V are those given by:

$$\begin{aligned} Q &= I\cos 2\chi \cos 2\psi \\ U &= I\cos 2\chi \sin 2\psi \\ V &= I\sin 2\chi \end{aligned} \tag{3.6}$$

In instances where all of these components are available the observer will be able to compute the polarization of the observed radiation. If the feeds of both antennae are orthogonal circular feeds the relation to the the four Stokes parameters is given by

$$\begin{bmatrix} < e_{R_p} e^*_{R_q} > & < e_{R_p} e^*_{L_q} > \\ < e_{L_p} e^*_{R_q} > & < e_{L_p} e^*_{L_q} > \end{bmatrix} \approx \begin{bmatrix} I+V & Q+iU \\ Q-iU & I-V \end{bmatrix} = B \tag{3.7}$$

For linear orthogonal feeds the relationship is slightly different [49]:

$$\begin{bmatrix} < e_{X_p} e^*_{X_q} > & < e_{X_p} e^*_{Y_q} > \\ < e_{Y_p} e^*_{X_q} > & < e_{Y_p} e^*_{Y_q} > \end{bmatrix} \approx \begin{bmatrix} I+Q & U+iV \\ U-iV & I-Q \end{bmatrix} = B \tag{3.8}$$

Equation 3.5 may thus be rewritten to take multiple polarizations, equipment gains and directional effects into account through a *Jones* matrix formulation, presented by Oleg Smirnov in a series of papers [49, 50, 51, 52]:

$$V_{pq} = G_p(t,\nu)\left(\int\int_{sources} D_p(l,m,t,\nu)BK_{pq}D_q(l,m,t,\nu)^H \frac{dldm}{\sqrt{1-l^2-m^2}}\right)G_q(t,\nu)^H \tag{3.9}$$

where $K = e^{-2\pi i(u_{pq}l+v_{pq}m+w_{pq}(n-1))}\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, the G terms are the direction-independent effects and the D terms are dependent on the direction of the source in l,m space. $J^H := J^{*^T}$ indicates the Hermitian transpose (the transpose of the complex conjugate matrix of J)

We shall henceforth refer to this relation as the *Radio Interferometric Measurement Equation (RIME).* The fundamental assumption here is that all effects on the measured intensities are linear and can therefore be written as two-dimensional matrices. Each antenna will then have its own set of corrections. They include corrections for Faraday rotation, ionospheric effects, tropospheric effects, the primary beam (and any rotation during observation) and pointing error. These corrections only account for amplitude scaling and phase shifts and can be combined in a layered approach by multiplying several Jones terms together (note that the terms may not necessarily commute, so the order of multiplication is important). We will come back to exactly how these terms are applied when discussing the inversion of the RIME.

Up to this point the formulation only accounts for two-element arrays. However, the linearity of the system allows us to simply combine multiple short term averages together to form a single observation using an entire array of antennae. In practice, we can correlate the signal between all possible pairs of antennae and reduce afterwards. This means that the data rates produced in correlation grows quadratically with the number of antennae (ie., the number of possible baselines). This includes the auto-correlated baselines (correlation of each antenna with itself).

$$\text{number of baselines } = \frac{n(n-1)}{2} + n \tag{3.10}$$

Even if many baselines are used to synthesize an aperture it will still be very sparsely sampled, especially when considering very large arrays. In fact, only a handful of small points will be sampled during a single integration period - hardly enough to represent a large continuous area! This means that the array telescope will be insensitive to very faint sources.

The solution to this problem lies in the beauty of the underpinning model: the flux measurements are *relative*. Recall also the assumption that the intensity of the sources of electromagnetic radiation *do not vary on a day-to-day basis*. This means an astronomer may fix one end of the baseline, and move the other end. Some of the early radio telescope arrays did just this. Although many of the antennae were stationary a handful of antennae were placed on tracks. These antennae would be moved to different positions to take multiple observations of the sky.

Martin Ryle and his associates introduced another method for building up the coverage in the measurement domain by using Earth rotation synthesis; as the earth rotates with respect to a fixed point on the celestial dome multiple coherence measurements can be taken, each at a different angle to every baseline, and hence along different points in the spatial frequency domain. This may sound strange, but it follows naturally from the way the local coordinate frame was defined in Equation 3.3. If the array tracks some point on the sky at some fixed declination over an extended period of time, the change in the hour angle will rotate every baseline. This becomes more explicit once the equation is recast from parametric to implicit form.

We start by obtaining the individual expressions for u,v,w in equation 3.3. If the expressions for u and v are manipulated into the familiar form of an ellipse it is easy to show that the following relation holds:

$$u^2 + \left(\frac{v - (L_Z/\lambda)\cos\delta_0}{\sin\delta_0} \right)^2 = \frac{L_X^2 + L_Y^2}{\lambda^2}$$

This ellipse is shifted along the v axis by $\frac{L_Z \cos\delta_0}{\lambda}$. The major axis has a length of $\frac{2\sqrt{L_X^2+L_Y^2}}{\lambda}$. It also follows that at low declinations the baselines of a pure east-to-west array will be nearly parallel to the pointing direction of the telescope. Not only will the foremost antenna shadow the antennae behind it, but there will be little coverage of the u, v frame. One way of improving u,v coverage is to add baselines that are sufficiently perpendicular to the rotation of the Earth (as is done in the VLA). However, these baselines are rotated into the w-direction and present a serious problem in imaging. We will come back to this when discussing wide-field imaging.

To make this more concrete consider a fictitious observation with the Extended Very Large Array. The antenna positions for all 27 antennae, available baselines and projected tracks are shown in Figure 3.9. With increasing frequency these tracks expand outward. This means that it is possible to effectively increase coverage by observing multiple frequencies and averaging them together. However, this will cause some radial smearing when images are synthesized.

Now that the various measurement products, the RIME model relating the sky and these measurements, and observation coordinate systems have been discussed we can move on to the synthesis imaging process where the RIME is inverted to reconstruct the sky intensity distribution. In the next chapter we will outline the issues surrounding narrow field imaging and then move on to a discussion on the wide field effect.

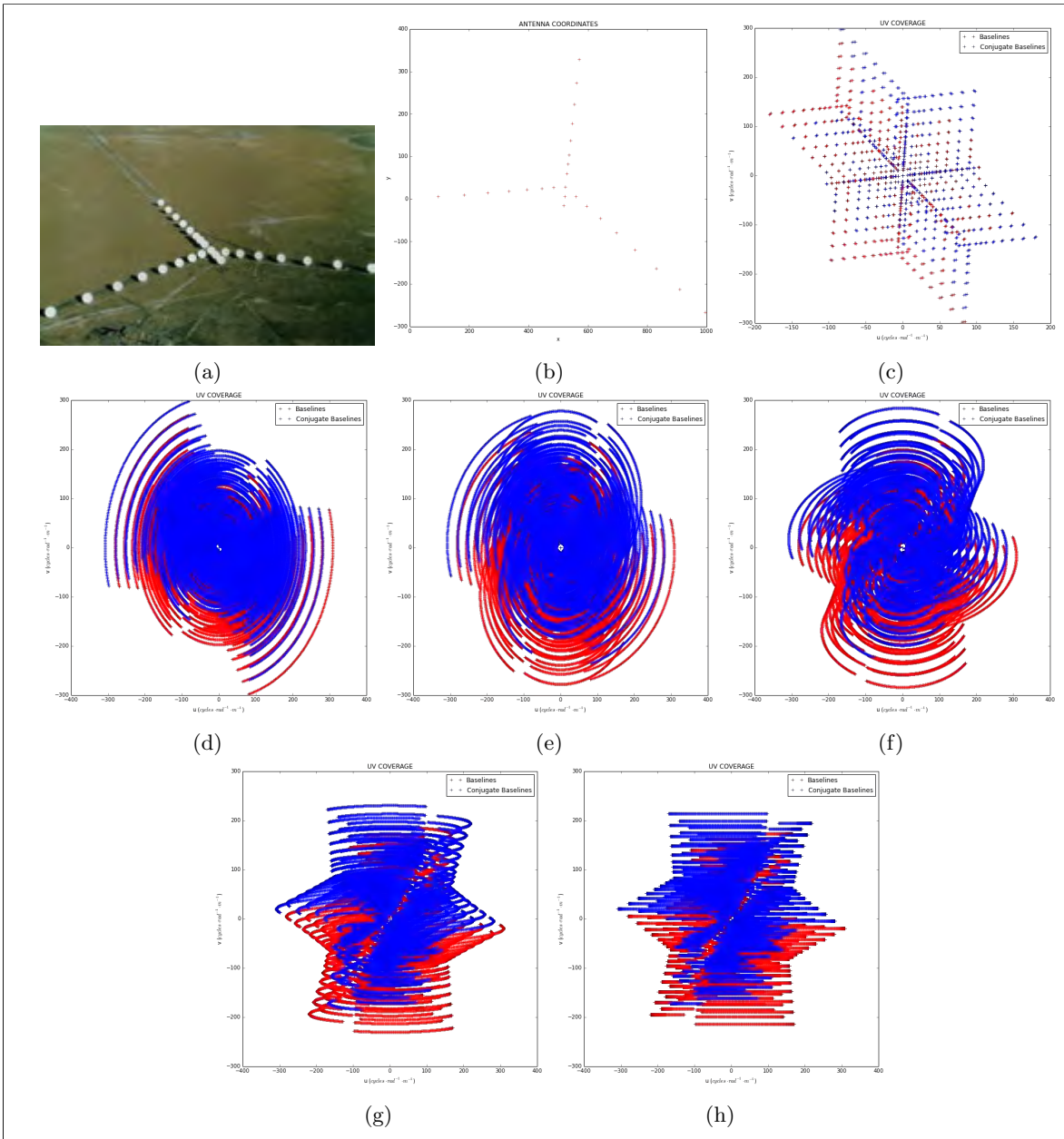


Figure 3.9: Here the eliptical tracks from all baselines of the 27 antennae of the reconfigurable Extended Very Large Array (National Radio Astronomy Observatory [New Mexico, US]) are shown at different declinations. In (a) a picture of the EVLA taken from `http://images.spaceref.com/news/2012/vla-625x412.jpg`. (b) shows the positions of the antennae when the array is in its compact "D" configuration. In (c) a short ("snapshot") 5min observation at NCP. In (d) a 6 hour observation at NCP. Figures (e)-(h) show 6 hour observations for $\delta_0 = 60°$, $30°$, $5°$ and $0°$ respectively. Notice that the conjugate baselines (baselines $\vec{b_{qp}}$ as opposed to $\vec{b_{pq}}$) measure the conjugates of the visibilities at the same spacial frequencies as their counterparts. These conjugate measurements adds no additional information about the sky and can be safely ignored.

Chapter 4

Wide-field image synthesis

In this chapter we discuss how the Radio Interferometric Measurement Equation (RIME) can be inverted to obtain a synthesized image of the radio sky. First an overview of the imaging pipeline is provided, before delving into the finer details of creating "dirty" images of the sky and correcting the wide field distortions introduced when breaking the assumptions made when the images are synthesized using a Fast Fourier Transform. Finally, we discuss previous literature on acceleration of the correcting process, before moving onto the design of our image in the next chapter. Before continuing readers not familiar with Fourier theory should refer to Appendix A for a refresher on basic signal processing concepts.

4.1 The calibration, imaging and deconvolution pipeline

Synthesizing images involve a cycle of computationally expensive steps as shown in the traditional major-minor cycle pipeline in Figure 4.1. Although this work is focussed solely on the "backwards" inversion step, we start by building up some context for readers not familiar with radio imaging. This context also serves to emphasize the importance of accelerating the resampling step, which by itself becomes computationally costly when synthesizing wide field images.

Both the backwards and forward predictive steps involve the RIME stated in Equation 3.9. The RIME can be thought of as predicting what measurements the telescope should make, given a model radio sky (*intensity distribution*), telescope behavior and environmental effects as inputs. If all the information is known, the inversion of this equation is as simple as taking an inverse Fourier Transform. To see why this is true consider that the continuous sky is simply an infinite sequence of shifted and scaled impulses (or delta functions), where each impulse is infinitesimally narrow, has an area of 1 and is infinitely high at the shifted position (zero everywhere else). The Fourier transform of each of these delta functions is then:

$$h(u,v) := \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \delta(l-l_0, m-m_0) e^{-2\pi i(u(l-l_0)+v(m-m_0))} dl dm \tag{4.1}$$

Observe that this is practically the same as the RIME (Equation 3.9). If the delta functions are replaced with scaled intensities over infinitesimally small areas of the sky as defined previously and the direction dependent and independent terms are added in the convolution integral, the two integrals are practically identical, apart from the $w(\sqrt{1-l^2-m^2}-1)$ term in the complex exponent and the $n = \sqrt{1-l^2-m^2}$ in the numerator. For narrow fields of view these terms are negligible.

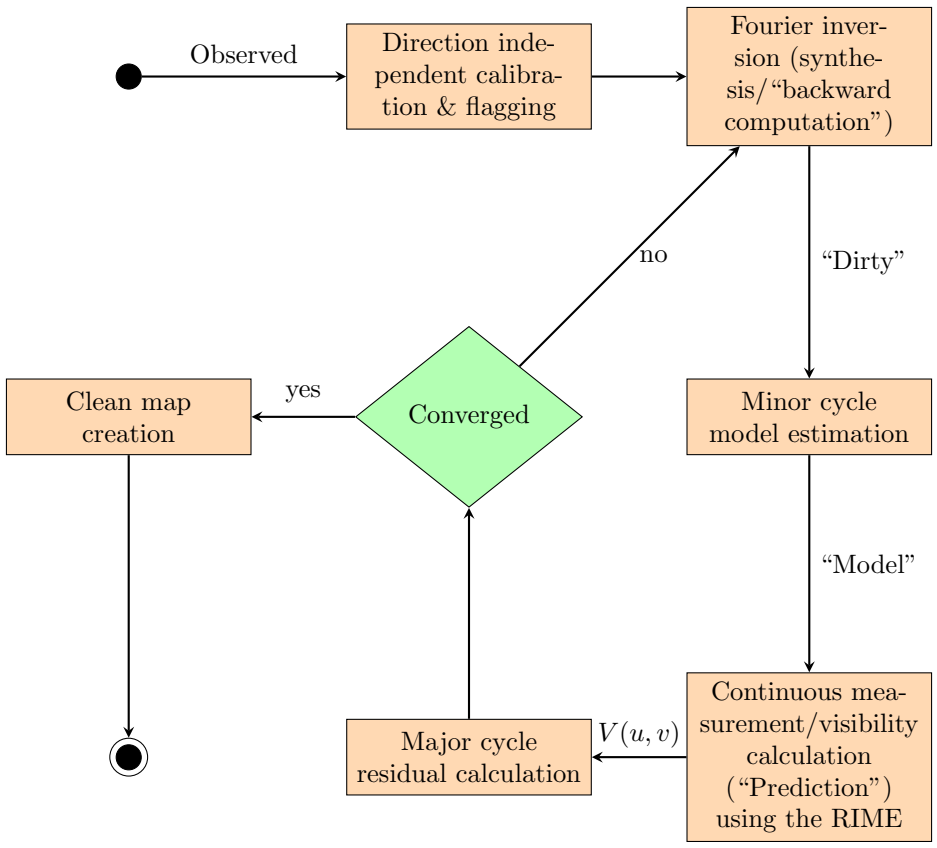


Figure 4.1: Here the traditional imaging pipeline is shown. The known directional-dependent calibration terms are taken into account during the "backwards" inversion step, while the unknown slowly-varying directional-dependent effects can be solved for if enough about the model sky is known.

In order to recover the *observed* intensity distribution, the measurements can simply be inverted using the inverse Fourier Transform. However, inversion is not all the synthesis imaging problem entails; a distinction has to be drawn between the *observed* and *true* radio sky: how much of the *true* radio sky can be recovered, given the imperfect measurements made with a radio interferometer over a limited period? Foremost, the effects of limited sampling in the measurement space (uv plane) must be considered. Formally, the tracks depicted in Figure 3.9 is called the sampling function and is defined as being 1 wherever a measurement is made and 0 elsewhere. The observed measurements (*visibilities*) are therefore a multiplication with this sampling function, $S(u,v)$:

$$V_{\text{obs}}(u,v) := V_{\text{RIME}}(u,v)S(u,v) \tag{4.2}$$

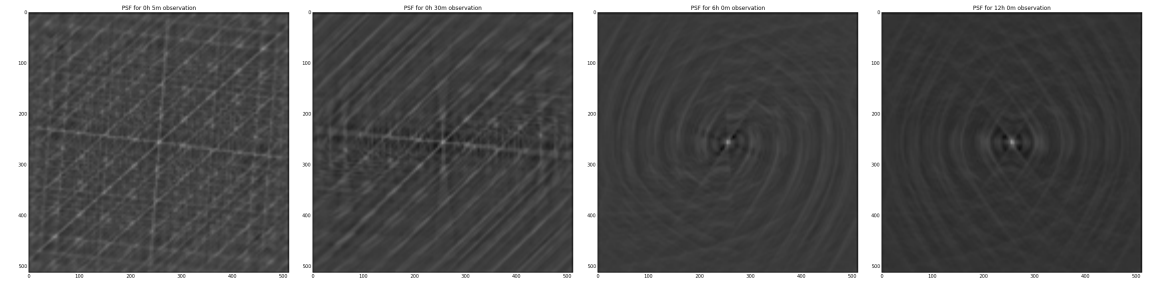
When equation 4.2 is expanded with the terms of the RIME and inverted the extent of the problem becomes apparent:

$$B_{\text{obs}}(l,m) = \sum_{t,\nu}(G_p(t,\nu)D_p(l,m,t,\nu)B_{\text{true}}(l,m)D_q^H(l,m,t,\nu)G_q^H(t,\nu)) * \text{PSF}(l,m,\nu) + \eta \tag{4.3}$$

where the G terms are the directional independent gains, the D terms are the directional dependent gains and the Point Spread Function, PSF, is the Fourier Transform of the sampling function, S. Since

the u,v coordinates depend on the wavelength the PSF in turn also scales depending on wavelength. η represents the background noise level that depends on system temperature, integration time and integration bandwidth, as mentioned previously.

Not all the information needed to accurately reconstruct the true intensity distribution, B_{true}, is available. In fact the PSF alone provides a challenge. The more complete the sampling function, the more accurate the image reconstruction. If measurements are taken over too short an observation there is no hope of deconvolving most of the PSF from the image. Even more concerning is the fact that the directional dependent effects cause a time-dependent convolution with the true measurements and serve to modulate different parts of the synthesized image at different levels during the course of an observation. Figure 4.2 shows how the PSF gradually improves with longer observation time, while Figure 4.3 provides an illustration of the challenge astronomers face when just considering the effects of limited sampling on the true sky, let alone telescope sensitivity and the direction dependent and independent effects.



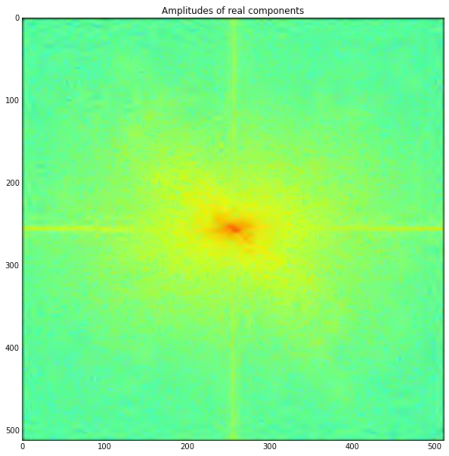
(a) 5 min snapshot (b) 30 min (c) 6 hrs (d) 12 hrs

Figure 4.2: Here the Point Spread Function is shown for various observation lengths on the EVLA D-configuration at $\delta = 30°$. The PSF gradually becomes more defined with longer observation time. Removing the repeating lobe-like structure around the source (clearly visible in the longer observations) is the primary goal of deconvolution, since it both suppresses fainter sources and adds structure and amplification to the background noise in the image.

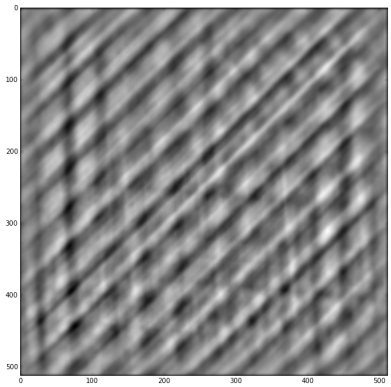
Referring back to the sampling tracks in Figure 3.9, highlights a further complication with the PSF: there are clearly more short baselines than long baselines. The resulting effect of this non-uniform distribution in the sampling function is a broadening of the PSF, and by implication a bias towards resolving extended structure in the image space. In order to resolve finer (compact) emission structures in the image it is necessary to divide through by the number of samples in the neighborhood of each point in the measurement space, *uniformly* weighting the synthesized image. The tracks also highlight an important aspect of interferometers in general: the PSF acts similarly to a high-pass filter. Adding longer baselines to an array increases the compactness of the PSF, which in turn serves to resolve higher frequency structure (edges, points, etc.) in the image. This is not always desirable - observing extended emission sources is equally important, which requires only short baselines, and more robust weighting methods are possible for the latter.

In addition to weighting by a density function the measured visibilities are also tapered by a Gaussian-like function in order to control the shape of the PSF and drive down the first few sidelobes of the PSF, at the slight expense of broadening the main lobe. The final weight also contains the inverse of the expected variance of the measurement in order to improve the signal to noise level in the synthesized image. The technicalities are not important here, and the reader may refer to Briggs et al. [57, Lecture 7] and Thompson et al. [59, p. 387-399] for more detail.

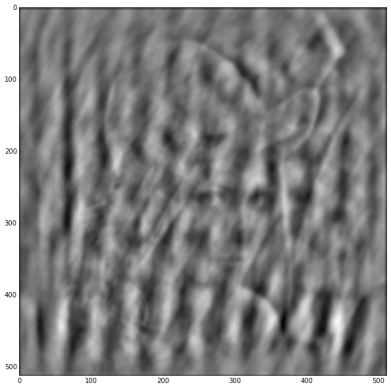
(a) "True"/model sky



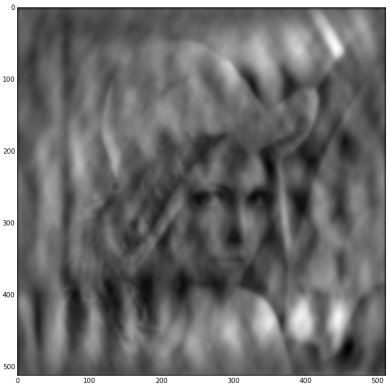
(b) Real component of measurement/fourier space, shifted so that the DC component is at the centre pixel.



(c) Dirty image after 30 min



(d) Dirty image after 6 hrs



(e) Dirty image after 12 hrs

Figure 4.3: If the model sky looked like the standard Lenna image used in image processing then limited sampling with an interferometer will produce a "dirty" image that must be devonvolved in order to recover as much of the "true"/model sky as possible.

This brings us to a strategy to deconvolve the extended lobe structure of the PSF from images. In order to achieve this goal an assumption is made about the distribution of sources in the sky: that the radio sky is mostly devoid of emission. If it is further assumed that most sources are compact point-like sources one possible strategy to remove some of the PSF structure becomes apparent, as exemplified in CLEAN

(Algorithm 1). The algorithm gradually subtracts the PSF-lobe structure from the brightest sources in the image, building up a set of delta functions that represent what the true sky may resemble under the prior assumptions.

Algorithm 1 The Högbom CLEAN algorithm

Given a dirty image, d of size $n \times m$ pixels
Given a synthesized PSF of size $2n \times 2m$ pixels
Let c be an all-zero cleaned image of size $n \times m$ pixels
Given the loop gain $0.0 < \gamma \leq 1.0$
Let $R_p = \infty$
Let $R_c = \infty$
repeat
  Let b be the position of the maximum value in d
  Set $c[b] = c[b] + \gamma \max d$
  Subtract from d the scaled beam $\gamma \max d \max \text{PSF}$, centred on position b
  Set $R_p = R_c$
  Set $R_c = \frac{\max d}{\text{rms } d}$
until $|R_p - R_c| \leq \epsilon$ **or** maximum iterations reached
Convolve c with a Gaussian-like function with half-amplitude width equal to that in the PSF
Set $c[...] = c[...] + d[...]$, ie. add the residual noise back into the cleaned image

Although CLEAN was initially intended only to work on a sky consisting of point sources, practice shows that it also deconvolves regions of extended emission. The output is then a collection of clean components spaced closely together.

The PSF subtraction in the image domain for every detected source is a relatively expensive operation. One of the most notable accelerations of CLEAN is the Cotton-Schwab major-minor cycle adaption. Here only a truncated PSF (up to the first few sidelobes) is subtracted in the image domain. The resulting clean model is then converted back to the continuous measurement domain (here again the measurement is predicted by the RIME) and then subtracted from the observed visibilities. This major-minor cycle approach works well when deconvolving multiple adjacent fields.

The convergence criteria of the algorithm are well beyond the introductory discussion here. Refer to Thompson et al. [59, ch 11] for a more detailed discussion and further reading on the topic. More recently new Compressive Sampling approaches have been suggested as alternatives to CLEAN. An example is the MORESANE [20] algorithm.

There is also the telescope calibration problem to consider. Apart from removing unwanted radio interference and erroneous measurements through flagging and solving for varying instrumental gains, it is necessary to calibrate known, and solve for unknown directional-dependent effects. One of the more pronounced directional-dependent effects in wide-field imaging is the antenna primary beam. We will return to ways of solutions to this problem after discussing the projection effect introduced by the $w(n-1)$ term in the RIME.

Next we discuss how images can be efficiently synthesized (Fourier-inverted / "backwards-processed") using a narrow-field approximation.

4.2 Narrow-field synthesis using the FFT

There are generally two techniques used to approximate the Fourier transforms between the observed visibilities and the (dirty) image: direct Fourier sums or by employing the Fast Fourier Transform. In the brute-force Direct Fourier Transform each image pixel is approximated through an by evaluation over all the observed visibilities (M of them in total). This approach requires on the order of $N^2M \approx N^4$ sine and cosine evaluations (where N is the size of a single dimension of a square grid) for a large number of visibilities [57, Lecture 7]. This equation can be extended to be as accurate as needed, taking per-pixel effects into account if necessary:

$$(\forall c \in \text{correlations})(\forall \text{ pixels } l, m) I_{\text{observed},c}[l, m] = \frac{1}{M} \sum_{k=1}^{M} \frac{V_{c,\text{observed}}[u, v]}{n} e^{2\pi i(ul+vm)} \tag{4.4}$$

This approach is prohibitively expensive when the number of baselines grows or the observation time increases. The second, less accurate, approach employs the Fast Fourier Transform (see Cochran et al. [11] for algorithmic details). Instead of having complexity order N^4, the two-dimensional Fast Fourier Transform can be computed with roughly $2N^2 \log N$ steps. However, there are three very serious problems with the second approach:

1. The Fast Fourier Transform yields a *tangent plane approximation* to the sky dome: the synthesized image is only accurate near the phase centre (by convention the centre pixel) of the image. This implies that the FFT may only be used when $\sqrt{1 - l^2 - m^2} - 1 \ll 1$. When doing wide-field synthesis this assumption is broken and correction is necessary.

2. The Fast Fourier Transform only operates on *regularly sampled* data. This simply means that the data has to be sampled on a uniformly spaced grid in order to apply the two dimensional FFT. This resampling step (essentially a variant of image upsampling) is very computationally expensive: it quickly comes to dominate the entire backwards step in wide-field synthesis. On the other end of the pipeline the prediction step suffers from the same affliction: an FFT can again be employed to convert from the model sky to continuous measurements, the only difference being that the regularly spaced visibility measurements have to be resampled to continuous coordinates.

3. The regularly sampled FFT assumes the signal is periodic: sources outside the desired field of view of the produced image is flipped back into the image on the opposite side. This necessitates filtering the image with a response pattern that only passes signal that is inside the field of view, and suppresses any outside energy.

The second and third problems are related and for now our discussion will focus on solving them, before moving onto the problem of wide-field synthesis.

We will use the terms “interpolation” and “resampling” interchangeably throughout this thesis. By these terms we refer to a process that either takes a set of continuous samples as input and yields a discretized set of samples as output (“gridding”) or works in the opposite direction (appropriately referred to as “degridding”). There is no single best way to interpolate data onto a regularly spaced grid and it is necessary to consider multiple strategies. The resampling problem is shared by many fields, including, but not necessarily limited to, the astronomical and medical imaging subfields. As such literature from both fields can be considered. An excellent comparative discussion on image interpolation is given by Thévenaz et al. [58], while Thompson and Bracewell [60] give a detailed overview and comparison of

different interpolation techniques when it comes to resampling the visibility measurements produced by an interferometer. Briggs et al. [57, Lecture 7] and Thompson et al. [59, p. 387-399] discuss the technical considerations that must be taken into account in the synthesis step. Our discussion highlights some key considerations and findings from these works.

Thompson and Bracewell [60] suggests an exact radial interpolation strategy, but it assumes that the baseline tracks (as depicted earlier) are concentric circles of an East-West array (at high declinations, for instance). Instead a less exact interpolation-by-convolution strategy is widely followed. In radio astronomy the resampling step focuses on a variant of the relatively fast class of linear interpolation operations:

$$(\forall f(a) \in \mathbb{C}, \phi(b) \in \mathbb{C}) f(x) = \sum_{k \in Q \subseteq \mathbb{R}^2} f(k)\phi(x-k) \tag{4.5}$$

The constant-selection function ϕ can be one of a myriad of functions proposed in the literature. These include linear, Lagrange, sinc, Gaussian, modified B-spline, etc. and may additionally be windowed by one of the large number of window functions as in the case of the sinc function, for example Hamming, Hanning, Blackman, Kaiser-Bessel, along with many others.

Notice that if resampling is applied to data that is already regularly sampled, as in traditional upsampling, and the constant selection function ϕ was evaluated at the resulting integer positions, Equation 4.5 would be a discrete convolution. Instead the operator as it stands here is not the regular convolution sum and should be thought of as an approximate convolution. Strictly speaking, convolutional gridding and degridding are not exact interpolation operations as Thompson et al. [59] and Sze Tan [54] points out, but rather interpolation-like (or as Briggs et al. [57, Lecture 7] put it: a *non-discrete integral convolution approximation*).

It is useful to think of this *non-discrete integral convolution* operator in terms of the ordinary upsampling operation. As with upsampling (where the data is already discretized) the space in-between samples is filled with zero values[1], but with gridding the regularly-spaced zeros are added in-between samples at non-regular spacings. Just as with ordinary upsampling it is then necessary to "smoothen" the new data points between the measured points using some form of interpolation or convolution, in order to avoid introducing new, alias-causing, high frequency terms in the higher resolution sampling. It is important to realize that the data is simply smeared out into the continuous measurement space, and by implication also over the newly inserted zeros, but that this does not recover any missing uv information. The rationale behind degridding can be explained using a very similar downsampling argument. To help visualize this "convolutional gridding" process refer to Figure 4.4.

The interpolated measurements taken at the grid points are not known exactly, and normally in interpolation techniques it is hoped that the variance of this error will be small when the step size in the resampling process becomes infinitesimally narrow. In the context of radio astronomy the "significantly oversampled" criterion is not normally met because of the significant memory this would require, especially considering the sizes of the arrays currently under construction. Instead, only a *critically sampled* image is usually produced during synthesis. Note that "critically sampled" refers to the Shannon-Nyquist

[1] "Zero-stuffed" in DSP nomenclature

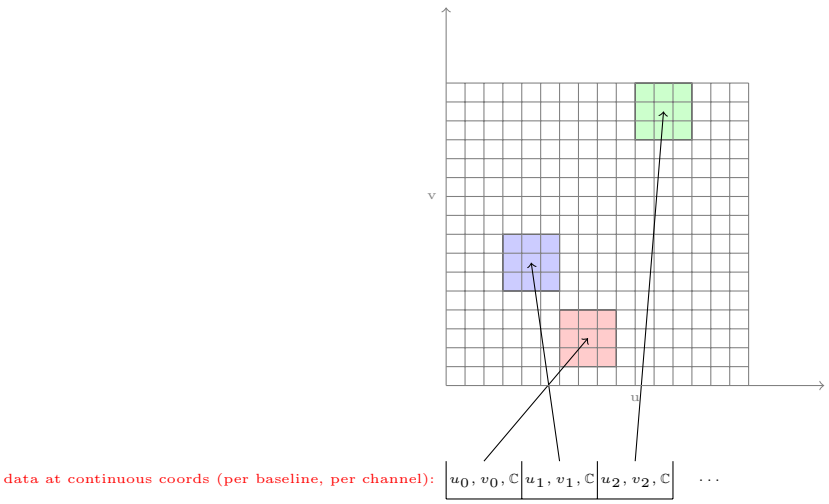


Figure 4.4: Each observed visibility (per channel and baseline) is centered at some precomputed coordinate in continuous u,v space and is "convolved" with some function ϕ, which extends only to a finite "full support" region as illustrated. The result is binned in a regularly spaced grid. This process essentially spreads each visibility out over a larger area in u,v space. It should be clear that this is not the standard convolution operator as defined in Appendix A. After all the observed visibilities have been gridded, an Inverse Fast Fourier Transform is performed and a correcting function is applied. Each correlation term (or Stokes sum) is typically resampled and transformed to its own image.

sampling criterion:

$$\begin{aligned} \text{cell}_l &= \frac{1}{2N_l\Delta u},\ \Delta u := \frac{1}{u_{\max}} \\ \text{cell}_m &= \frac{1}{2N_m\Delta v},\ \Delta v := \frac{1}{v_{\max}} \end{aligned} \tag{4.6}$$

Here cell_l and cell_m are the pixel sizes given in degrees (or equivalently arcminutes, arcseconds or radians). $N_l\Delta u$ and $N_m\Delta v$ correspond to maximum frequencies in the Fourier / measurement space. If the images are sub-sampled the longest baselines will fall off the uv grid and angular resolution will be lost.

This sampling criterion alone justifies our statement that the Fourier response of the resampling function should be considered of higher priority than the approximation criterion, unlike in other contexts where interpolation quality may be most important. The energy reduction properties of ϕ can be stated in terms of maximizing the following integral ratio for all square-integrable ϕ functions[2]:

$$\frac{\int_{\text{FOV}} |[\mathcal{F}\phi](l,m)|^2 dS}{\int_{-\infty}^{\infty} |[\mathcal{F}\phi](l,m)|^2 dS} \tag{4.7}$$

To better understand why aliasing energy is such a big concern we have to define the gridding operation somewhat more rigorously. Each (discrete) measurement taken by an interferometer in the continuous uv space is "convolved" with a two-dimensional interpolation function in order to create a continuous function. This function is then discretized again onto a set of regular coordinates by a "bed-of-nails"

[2]It is possible to define another criterion here, for example to promote accurate interpolation over energy concentration. Sze Tan [54] for instance defined this as minimizing the difference between the Direct Transform and the FFT approach

function. Mathematically we can say:

$$V_{\text{gridded}}[u,v] = [V_{\text{sampled,observed}}(u,v) * \phi(u,v)]\frac{\text{III}(u,v)}{\Delta u \Delta v} \tag{4.8}$$

where III is the shah function defined as:

$$\text{III}(u/\Delta u, v/\Delta v) = \Delta u \Delta v \sum_{j=-\infty, j\in\mathbb{Z}}^{\infty} \sum_{i=-\infty, i\in\mathbb{Z}}^{\infty} \delta(u - j\Delta u, v - i\Delta v) \tag{4.9}$$

Convolution with the Fourier transform of the shah function (composed itself by many band-limited impulses) creates a sum of periodic functions in the image domain [3]. The result is a periodic field of view that repeats at $M\text{cell}_l$ and $N\text{cell}_m$ intervals for an $M \times N$ image (ie. at multiples of the field of view). In practice not all the energy from these replicated fields can be stopped at the edge of the field of view, and this is responsible for the aliasing seen in the images. Further truncation of the shah function to represent only a single field of view, as well as the truncation of the convolving function, also contribute to the aliasing. It should be understood that the PSF sidelobes from sources outside the field of view that legitimately fall inside the field of view are not removed and this will raise the noise levels inside a deconvolved image if the sources responsible for those sidelobes are not included in the deconvolved model.

The ϕ functions considered here all have the property of *separability*, meaning that $\phi(u,v) = \phi(u)\phi(v)$. We will return to this later on when discussing w-projection. For now the discussion will focus on functions of one variable.

After alias reduction speed becomes a major consideration. Due to the large measurement datasets produced with larger arrays the convolutional resampling process has to be fast; the complexity of the resampling step grows as MC^2, where C is the support size of the convolution filter and M is the number of visibilities. Therefore, the convolution function ϕ is normally pretabulated for a given support size (in grid steps). Additionally, it is very important to oversample the precomputed ϕ_{filter} to conserve the spatial relation in the measured coherence function and to attain *high dynamic range images*[4].

To further understand why the filter has to be significantly oversampled (usually dozens of times) consider that interferometers take measurements in the Fourier space, where any rounding operation (or snapping) of the u,v coordinates in either the grid or the filter will cause fringe-like decorrelation in the observed sources (think back to the Fourier shift theorem). Thévenaz et al. [58] also points out that ϕ must be symmetrical (ie. $\phi(x) = \phi(-x)$ and $\phi_{\text{filter}}[x] = \phi_{\text{filter}}[-x]$) to preserve the image phase correctly.

The image phase consideration effectively precludes using nearest-neighbor[5] interpolation. The nearest neighbour technique simply snaps (box function) points close to the grid point into the sum at that point, without any consideration of the visibility's distance from the grid point. Additionally, the Fourier transform of the box function is an *infinite* sinc function. Since the sinc function slowly ripples out towards infinity it is not a good response when it comes to reducing the unwanted energy from sources that fall outside the field of view.

Convolutional gridding is a more attractive approach to cell-summing since the distance between the grid

[3] It is useful to note that the Fourier transform of a band-limited (non-zero over a finite range) function, such as a box function, is a function that stretches over an infinite support region

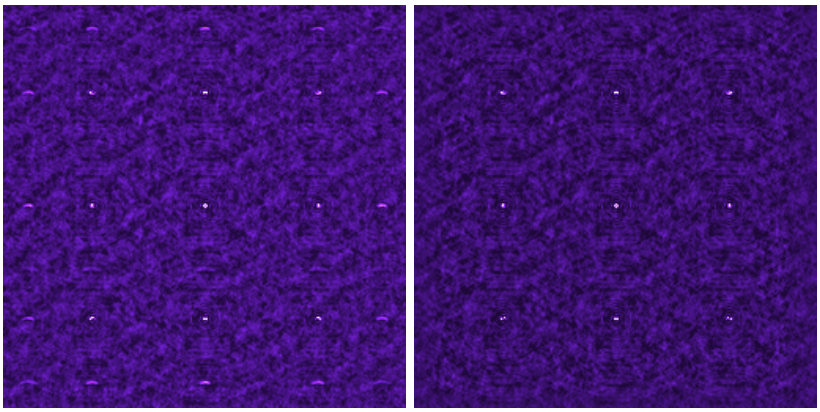
[4] Images having a high peak value to noise level

[5] or *cell-summing* techniques as Thompson and Bracewell [60] puts it

point and the measured uv point is taken into consideration when picking a set of convolution weights from the oversampled filter, as illustrated in Figure 4.5. The fractional offset is simply calculated between the nearest regular grid coordinate and measured uv coordinate and is used to pick the nearest filter value in the oversampled filter.

Figure 4.5: Across the literature the definition for "filter support or window width" varies considerably. To illustrate the use of the terminology in our work a ficticious filter is shown here. Here the padded and oversampled $\phi_{\text{filter}}[x]$ is illustrated for a 3-cell full-support region (half support of 1 to both sides of the centre value), padded with one value on both sides. The filter is 5x oversampled, as indicated by the asterisks between the bars, the latter representing the cell-spacing (Δu for instance) used for the grid. If the measured uv coordinate falls exactly on the nearest grid cell then values 6,11 and 16 are selected as interpolation coefficients. If $\text{round}(\text{frac}(u,v)m_{\text{oversample factor}}) = 2$ for instance then 8, 13 and 18 are selected for the 3 grid points being "convolved" or "smeared" onto. In other words: a denser bed of nails is placed over the bed of nails of the grid and the closest set of coefficients for the convolution is selected. Briggs et al. [57, Lecture 7] notes that this discretization of ϕ will cause a minor replication effect with a very long period of $\frac{l_{\text{oversample factor}}}{\Delta u}$ and $\frac{m_{\text{oversample factor}}}{\Delta v}$ in the respective u,v directions.

If the last observation about box functions is turned on its head we arrive at a partial solution to the problem of selecting a filtering function that better limits aliasing energy; convolving with the *infinite* sinc yields a box response in the image domain. Unfortunately, it is impossible to convolve with an infinite function to exactly reconstruct the sought-after box response in the image domain. It is also computationally prohibitive to increase the support range of the convolution filter, but without large support sizes the filter's Fourier response does not taper (or "roll off") immediately. See Figure 4.6 for a comparative example of the significant improvement in simply switching from nearest neighbor interpolation to the ordinary box-windowed sinc function.



(a) Nearest neighbor interpolation (b) sinc 8x8 full support

Figure 4.6: Here a grid pattern sky model was simulated using MeqTrees [38] and imaged with our imager, first using Nearest Neighbour and then using an ordinary unwindowed sinc function. Sources that fall slightly outside the field of view are aliased back in when the energy outside the field of view is not limited as expected.

To address this problem the literature is filled with alternatives to the truncation window, of which the Kaiser-Bessel window proposed by Jackson et al. [29] yields particularly good results. Offringa et al.

[40][6] used this filter in their implementation of a w-stacking (explained in the next section) imager with great success. The Keiser-Bessel window is defined as:

$$\frac{1}{W} I_0(\beta\sqrt{1-(2x/W)^2}) \tag{4.10}$$

Where W is the full support of the convolution filter. See Jackson et al.[29] for the tabulated constants used for β.

An alternative to using a windowed sinc function is to use a prolate spheriodal function. It is also widely employed in astronomy imaging software. The Spheriodal functions have the property we are looking for, in that most of their energy is concentrated over the centre of the function, as measured by a weighted variant of Equation 4.7. This is proven generally by Donald Rhodes [45]. A later analysis by Frederic Schwab [48] confirms the relatively good performance of the spheriodal functions compared with many others as an anti-aliasing filter. The prolate spheriodal is defined for the special case of $\alpha = 0$ as:

$$|1-(2x/W)^2|^{\alpha}\psi_{\alpha 0}(0.5\pi W, 2x/W) \tag{4.11}$$

Here W is again the full support of the convolution filter and x increases in steps of Δu. When $\alpha > 0$ a weighted energy concentration ratio is maximized instead. The ψ_{xy} function is the one defined by Donald Rhodes [45]. Its definition alone is well beyond the scope of this discussion, and in fact it is quite difficult to compute for arbitrary support and oversampling parameters.

After taking the Fourier transform the tapering effects of ϕ may optionally be corrected for by pointwise division with the Fourier transform of ϕ. This does not completely eliminate ϕ from the resulting expression, but has the effect of flattening the response of the pass band (removing the tapering towards the edges of the image), but at the same time increasing the amplitude of the aliasing-sidelobe responses.

Measuring the remaining energy outside the grid-corrected image, Jackson et al.[29] show that the Keiser-Bessel-windowed sinc achieves very similar performance to the Gaussian and Prolate Spheriodal Function for small (preferable) support regions and similar performance to the prolate spheriodal for larger windows, whilst being considerably easier to compute. One possible downside to windowed sinc functions are their inability to reproduce a constant function [58]. A synthesized image with a non-zero mean will appear either too light or too dark when compared to a model image. It is unclear if the prolate spheriodal function suffers from the same problem. Thévenaz et al. also point out that the sinc introduces blockiness in the image.

Lastly, it is necessary to add that both the Direct Fourier Transform and Fast Fourier Transform approaches have to scale the field of view of the image according to the cell size and number of pixels in the image. In the Fast Fourier Transform approach this is achieved by scaling the measurement domain's u,v coordinates (measured in cycles.m^{-1}.rad^{-1}) such that, when inverted, the image has the desired field of view. To achieve this we use the *similarity* property of the FFT:

$$\alpha^{-n} F(x/\alpha) \rightleftharpoons f(\alpha x) \tag{4.12}$$

Here n depends on the dimensionality of the Fourier transform. If the desired field of view (measured in radians, centred at the field centre) is multiplied to each sampled u,v coordinate then the synthesized

[6]Special thanks goes to André Offringa for informative discussions around this topic

image will have a field of view ranging between $-0.5\text{cell}_l N_l \leq l_{\text{deg}} \leq 0.5\text{cell}_l N_l$ and $-0.5\text{cell}_m N_m \leq m_{\text{deg}} \leq 0.5\text{cell}_m N_m$. It is further important to note that, by convention, the gridded visibilities are shifted before Fourier transform such that the 0 frequency component (the "DC" component) is at the centre of the grid and the corresponding centre of the observed field on the image is at the centre pixel of the image.

To summarize, the convolutional gridding process is shown in Figure 4.7.

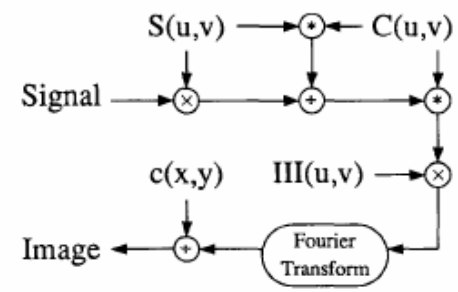


Figure 4.7: The classic synthesis by convolutional gridding and Inverse Fast Fourier transform. The signal (u,v domain) is multiplied by a sampling function (and optionally weighted by the computed and tapered density function). Then it is convolved and resampled onto the regular grid, Fourier transformed and optionally grid-corrected by point-wise division. Courtesy of Jackson et al. [29]

4.3 Wide-field distortions and the problem of non-coplanar baselines

Up to this point the discussion on employing the Fast Fourier Transform to invert interferometer measurements and discretize the sky has assumed that the the field of view can be well-approximated by a tangent projection plane. When synthesizing an image over a larger field of view this assumption is broken. This is especially true for lower-frequency instruments, such as LOFAR. Much of the remainder of this chapter will focus on the effects the additional phase delay term $w(n-1)$ in the RIME has on the synthesized image. This has been extensively studied over the past two decades and is quite well understood. Our discussion covers the proposed solutions in the literature and will draw extensively on the works of Perley [57, Lecture 19][7], Cornwell and Perley [13], Cornwell, Golap and Bhatnagar [12], Kogan and Greisen [34], and Tasse [55][8].

The wide-field effect arises due to the combination of two errors:

1. Firstly the array geometry in significantly non-coplanar arrays will lead to w-values that cannot be ignored. Similarly, the non-East-West baselines in arrays will not remain coplanar over the duration of an observation. These will be rotated up into the w direction as the Earth rotates, even if the physical array layout is on a very flat plane (as is true for the JVLA, for example).

[7] A word of thanks to Richard Perley for his insightful discussions on the problem and clarification on his tangent facet imaging approach in late 2014

[8] This document is an internal memorandum explaining some of the ideas exploited in this work. A great debt of gratitude is owed to Cyril Tasse for compiling this discussion on facet and hybrid facet imaging.

2. Secondly the image projection geometry worsens the signal phase difference between antennae. The distance between the planar projection of the sky and the celestial sphere (or unit radius "sky dome") cannot be discounted far away from the tangent point of the image. This error in distance is expressed as $n - n_0$ where $n = \sqrt{1 - l^2 - m^2}$ at the correct position of the source on the celestial sphere, and $n_0 = \sqrt{1 - l_0^2 - m_0^2}$ is the tangent point / projection pole of the image produced using the Fast Fourier Transform, assuming this projection is the ordinary orthogonal projection of the sphere onto to the image.

As Perley [57, Lecture 19] points out this phase difference is not a physical delay in the strictest sense of the word, but arises merely because of the geometry of the array and coordinate systems. In essence it occurs because the field of view is too wide, or it is sampled with a tilted interferometer or both simultaneously. The effects are combined as $w(\sqrt{1 - l^2 - m^2} - \sqrt{1 - l_0^2 - m_0^2})$ where $\sqrt{1 - l_0^2 - m_0^2} = 1$ if the projection pole is at the same coordinate as the centre of the field being observed (which we take to be true for the remainder of the discussion). Because the phase propagation term $\tau = \frac{\vec{b} \cdot \vec{s}}{c}$ as shown in Figure 3.6 depends on the distance a source is away from the delay tracking centre at $\vec{s}_0$ it will continuously change during the course of an observation. As the baselines are rotated with respect to a fixed position in the sky, the apparent position of sources in the image will move as the earth rotates. To see why the latter statement is true from a pure mathematical standpoint consider again the Fourier Shift Theorem:

$$g(x - \Delta) \rightleftharpoons G(f) e^{-2\pi i f \Delta} \tag{4.13}$$

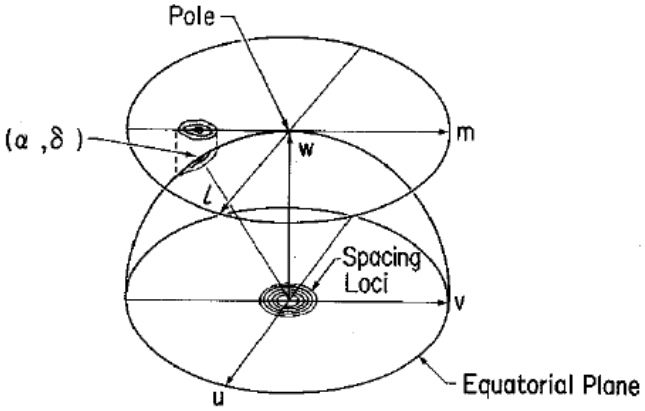
This theorem says that adding a delay term through multiplication by a complex exponential in the Fourier (measurement) domain will serve to shift values in the signal (sky/image) domain. In this context the effect is somewhat more complicated because sources closer to the tracking centre of the field are not as affected by this shift as those further away. Refer to Figure 4.8 for an illustration of the two terms: w and $(n - 1)$.

To add to this problem the difference in the w-term between the antennae increases for observations at lower elevation angles (ie. those observations near the horizon). To see why this is true consider that the baseline tracks of observations near the celestial pole consists almost entirely of level concentric circles. As the declination (and elevation) angle is decreased more and more of the circles have non-zero w-components, to the point that their projection onto the uv plane becomes a straight line at $\delta = 0$. This last observation gives us an estimation for the absolute of the maximum w-value the observation may contain[9], as well as the sample spacing needed to satisfy the Shannon-Nyquist sampling constraint, as Thompson [57, Lecture 2] notes (refer to Figure 4.9 for an illustration of the argument):

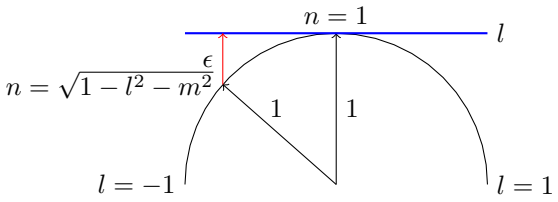
$$\begin{aligned} w_{\max} &\approx |\frac{\vec{b}_{\max}}{\lambda}| \\ \text{cell}_l = \text{cell}_m = \text{cell}_n &\approx 0.5 \frac{\lambda}{|\vec{b}_{\max}|} \end{aligned} \tag{4.14}$$

The resulting effect on the apparent position of a source over time of this declination-dependent increase in w is plotted in Figure 4.10. In addition we have imaged a simulated sky model consisting of point sources in a grid pattern to show the resulting decorrelation in the resolution of sources far from the delay tracking centre and their corrected version in the same figure.

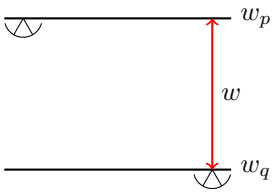
[9] This is a conservative estimate for imaging at lower elevation angles (near the horizon), the maximum w-value will be significantly less near Zenith



(a)



(b) Error between the orthogonal planar FFT approximation to the sky and the correct position on the celestial sphere



(c) Delay in signal propagation between antennas in an array-based telescope at some instant in time

Figure 4.8: In (a) the projected position of a source on the sky dome onto the plane tangent to the projection pole is shown. The baseline tracks are concentric circles exactly parallel to the Earth's equator in the case of East-West arrays. Image adapted from Thompson [57, Lecture 2]. The combined propagation delay of emission from sources far away from the telescope pointing centre is a combination of the error between the planar approximation and the celestial sphere (b) and the phase difference between pairs of antennae in the telescope pointing direction as shown in (c). The total phase error is expressed as $w(n-1)$. The multiplicative effect of this w-term becomes a significant problem in large non-East-West antenna arrays, where the baselines between the furthest-separated antenna pairs become significantly non-coplanar as the Earth rotates.

To correct this effect one may consider employing a three dimensional Fourier transform. Perley [57, Lecture 19] shows that the image along the sky plane can be related to the intensity cube derived by a three-dimensional Fourier transform, assuming n as an independent variable. The sky then is defined by all the points in the transformed cube that lie on a shifted unit sphere. The relation also provides information about the possible effects of the sampling function; the PSF is not only convolved with points on the sky sphere, but the surrounding areas in the cube.

Such a full three-dimensional transform is not a practical solution considering the memory required to form several layers, each the size of the image, and the computational cost because the cube layers in

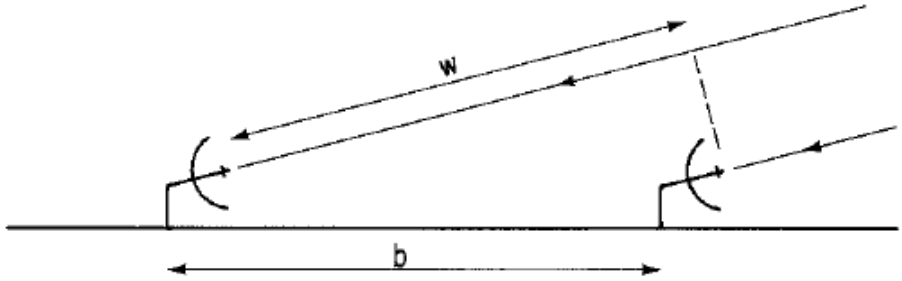


Figure 4.9: When sources are observed at low elevation (or similarly at low declinations) angles, for instance those sources rising or setting at the horizon, the delay between the two observing antennae is maximized and by implication w since w is usually defined to point towards the tracking centre of the field. Image courtesy of Thompson [57, Lecture 2]

the n dimension must be computed using a direct Fourier transform to overcome the aliasing effects introduced by the longest baseline. Instead Richard Perley [57, Lecture 19] mentions two more solutions to the problem and Cornwell et al. [12] later propose a third solution:

1. Warped snapshot imaging. If the observation time is sufficiently short, all the baselines of the array remain relatively coplanar[10]. By neglecting the w-term a small distortion is introduced in the apparent position of sources This becomes the determinant factor in deciding the maximum observation time for each snapshot image. Due to the continuous rotation of the tangent image plane in this solution it is necessary to accurately interpolate the cell coordinates between snapshots. This latter operation becomes the dominant computational factor in snapshot imaging (see Perley [57, Lecture 19] and Cornwell, Voronkov and Humphreys [14] for details).

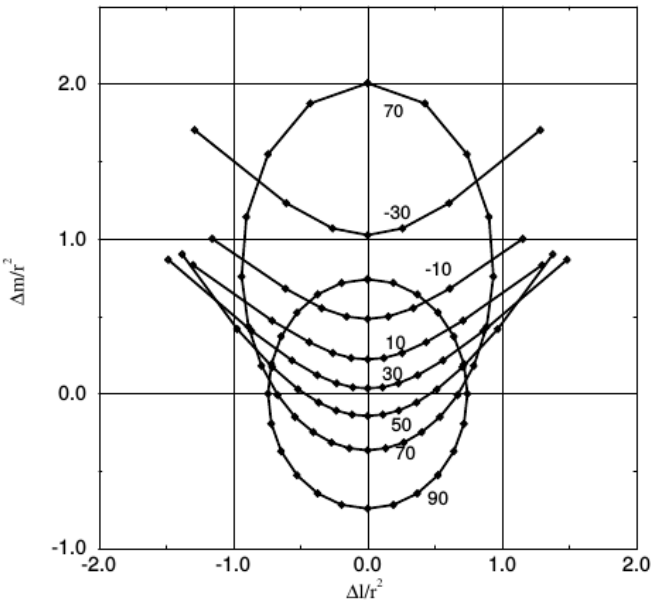
2. Facet imaging [11]. The varying w term can be assumed to be near-constant across a narrow field of view, therefore a wide field of view can be broken up into several narrow field images, each limiting the distance between the approximating plane and the celestial sphere. There are both *non-coplanar* and *coplanar* faceting approaches, as well as the appropriate transforms and coordinate reprojections in the measurement and image spaces to accompany these two variants. The details will be outlined in the next few sections.

3. W-projection [12] and W-stacking [40]. The w-term can be thought of as a w-dependent convolution in the Fourier domain. By convolving each visibility with the Fourier transform of $w(n-1)$ during the gridding process it is possible to re-introduce this multiplicative phase term in the image domain. Similarly, it is also possible to divide the sky image up into several w-dependent layers and point-wise multiply the w phase screen into each layer in image space, reducing the planes into a single image afterwards (w-stacking).

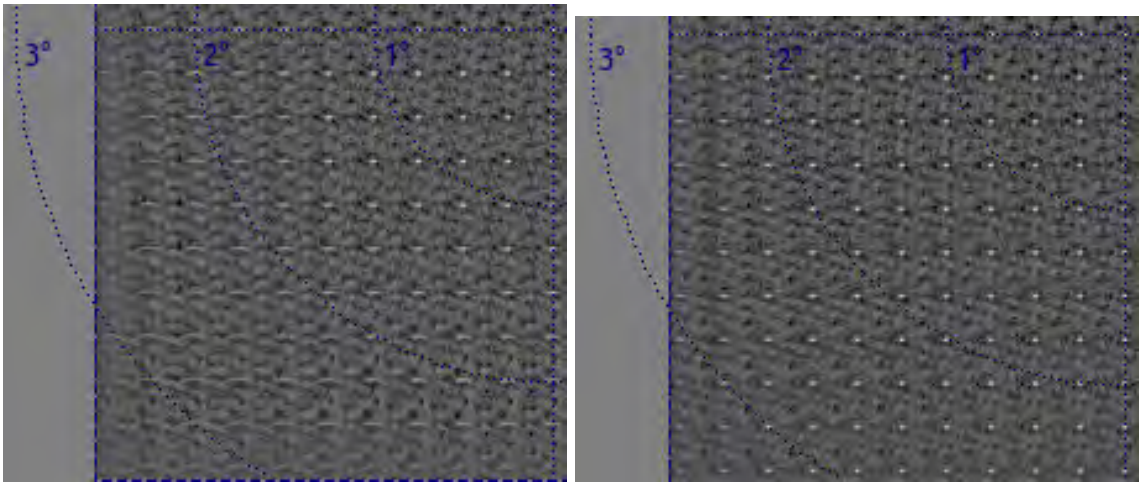
The approaches above attempt to either drive w down to zero (snapshots or w-projection) or drive $(n-1)$ down to zero (facet imaging). There is no reason why the approaches cannot be combined, for instance Cornwell et al. [14] combine w-projection and traditional snapshot imaging, while Offringa et al. include a Zenithal w-snapshot synthesis mode in their work. Our work, on the other hand, focuses on combining traditional facet imaging and w-projection, in order to harness some of the positive aspects of both approaches.

[10] within a fraction of the cell size in n

[11] Or a "fly's eye" imaging approach as Bill Cotton labels it.



(a)



(b) (c)

Figure 4.10: In (a) the apparent source positions for sources less than one radian from field centre in several hourly-interval snapshots on the VLA at 7 declinations are plotted by Perley [57, Lecture 19]. The dots of the zero hour angle is indicated on the vertical axis. The angular position errors here are measured, where $r^2 = l^2 + m^2$. In (b) a simulated gridded sky (without effects of the antenna beam or noise included) is synthesized as a dirty image using the regular planar transform as described in the previous section. In (c) the dirty map is synthesized again, here it is corrected using the w-projection technique explained later on. The point sources near the edge of the field are no longer smeared.

4.4 Non-coplanar facet imaging

The goal in faceting is to approximate a wider field of view with many small narrow field images. Perley and Cornwell [13] propose a polyhedron-like faceting approach, where each narrow-field facet is tangent to the celestial sphere at its own phase tracking centre (l_i, m_i). We will classify this basic faceting approach as non-coplanar uvw-space faceting, because each facet lies on its own tangent plane and the coordinate transformations necessary for this tilted polyhedron approach are done in uvw-coordinates

(measurement coordinates) and not in image space.

Just as with regular narrow field imaging the phase term $e^{-2\pi i\vec{b}\cdot(\vec{s}-\vec{s}_0)}$ is taken relative to a delay tracking centre in the direction $\vec{s}_0$. To synthesize an image with a new phase tracking centre, $\vec{s}_i$, it is only necessary to employ the shift theorem. This follows naturally from the RIME (simplified here). Let $(l_\Delta, m_\Delta, n_\Delta) = (l_i - l_0, m_i - m_0, n_i - n_0)$, then:

$$\begin{aligned} V(u,v,w) &\approx \int\int B(l-l_i, m-m_i, n-n_i) e^{-2\pi i[u(l-l_i)+v(m-m_i)+w(n-n_i)]} \frac{dldm}{n} \\ &\approx \int\int B(l-l_i, m-m_i, n-n_i) e^{-2\pi i[u(l-l_0-l_\Delta)+v(m-m_0-m_\Delta)+w(n-n_0-n_\Delta)]} \frac{dldm}{n} \\ &\approx \left[\int\int B(l-l_0, m-m_0, n-n_0) e^{-2\pi i[u(l-l_0)+v(m-m_0)+w(n-n_0)]} \frac{dldm}{n}\right] e^{2\pi i[ul_\Delta, vm_\Delta, wn_\Delta]} \end{aligned} \tag{4.15}$$

This says that the telescope can be electrically steered to take measurements with respect to a new phase centre by multiplying each known measurment in the database by a complex exponential. However, the intensity measurements are still measured with respect to the original u,v,w coordinates. The result of the latter observation is that the sky is projected onto facet planes that are tangent to the original phase tracking centre, (l_0, m_0) and not the new phase tracking centres, (l_i, m_i).

Perley [57, Lecture 19] shows that only employing a phase shift per visibility without regard to the tangency of the resulting facets will require the creation of many more facets further from the original field centre. This is illustrated in Figure 4.11 Employing a small angle approximation to estimate $(n-1)$ and applying the critical sampling criteria estimated in Equation 4.14, he estimates the maximum undistorted field of view to be:

$$\theta_{\max} \approx \frac{\lambda}{2|\vec{b}_{\max}|\theta_i} \tag{4.16}$$

where θ_i is the angle to the new phase centre.

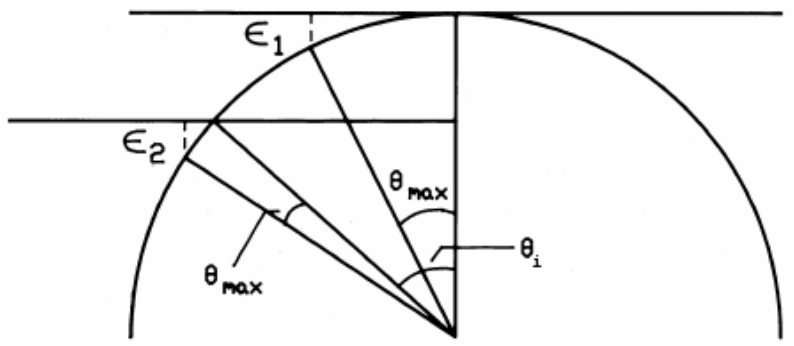


Figure 4.11: Only phase steering the visibilities to new phase centres without tilting the u,v,w coordinates to correspond to the new phase tracking centre significantly reduces the achievable field of view. Here instead each facet is parallel to the original tangent plane. As the new centre is taken further away from the original phase tracking centre the effective facet size must be shrunk down to achieve a comparable projection error at the edge of the synthesized facets. Image adapted from Rick Perley [57, Lecture 19]

In order to make each facet tangent to the celestial sphere at (l_i, m_i) it is necessary to employ the rotation matrices from Equation 3.4 to compute new u',v',w' coordinates *after* the visibilities have been phase

shifted using the old u,v,w coordinates.

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = R(\alpha_i, \delta_i) R^T(\alpha_0, \delta_0) \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{4.17}$$

Ensuring the facets are tangent to the new phase centres in effect ensures that the height difference between the sky and the projected facet remains comparable between corresponding pixels. Perley estimates that such a polyhedron approach achieves a field of view estimated by:

$$\theta_{\max} \approx \frac{\sqrt{\lambda}}{\sqrt{|\vec{b}_{\max}|}} \tag{4.18}$$

Using a polyhedron faceting approach is clearly an improvement over naive phase steering if enough facets are created to satisfy the sampling criterion. To determine how many facets are required we first have to define the phase error as (refer back to Figure 4.8):

$$\xi_{\max} := \frac{2\pi w_{\max} \epsilon}{\lambda_{\min}},\ \epsilon := \sqrt{1 - l^2 - m^2} - \sqrt{1 - l_0^2 - m_0^2} \text{ and ideally } 0 \leq \xi \ll 1 \tag{4.19}$$

Then we derive an indication of the number of facets needed to limit this w-dependent phase error between the celestial sphere and orthogonally projected facets (linearly spaced) at the corner of each facet:

$$N_{\text{facets}} = \frac{\max(\theta_l, \theta_m)}{2\cos^{-1}\left[\sin(\delta_0 + \theta_l/2)\sin\delta_0 + \cos(\delta_0 + \theta_l/2)\cos\delta_0\cos(\theta_m/2) - \frac{\lambda_{\min}\xi}{2\pi w_{\max}}\right]} \tag{4.20}$$

Here the half facet size (in l and m) is given as $\theta_{l_f}/2 = \theta_l/(2N_{\text{facets}})$ and $\theta_{m_f}/2 = \theta_m/(2N_{\text{facets}})$, respectively. The arc subtended by the angle to the corner of the facet has length $\cos(\theta_{l_f}/2)\cos(\theta_{m_f}/2)$ using the spherical rule of cosines and assuming a unit celestial sphere and orthogonal u and v bases. The angle to the corner of the image is small, so we might as well just use the small angle approximation.

For reference a sketch algorithm for facet synthesis (backwards step) is given in Algorithm 2. The forward step will work roughly in reverse. The faceting transformations in practice are combined with the gridding algorithm.

For reference the relationship between the celestial and projected image plane coordinates are given here, where α_p, δ_p is the tangent point[12]:

$$\begin{aligned} l &= \cos\delta\sin(\alpha - \alpha_p) \\ m &= \sin\delta\cos\delta_p - \cos\delta\sin\delta_p\cos(\alpha - \alpha_p) \\ n &= \sqrt{1 - l^2 - m^2} \end{aligned} \tag{4.21}$$

For our work we assume the projection is the orthogonal coordinate system specified here. It is well worth noting that the orthogonal projection of sources is not accurate for large fields of view, as noted

[12]The tangent point is assumed to be the same as the phase reference centre for the orthogonal projection, which is a special case of an Azimuthal projection. For more details see the AIPS convention [25] and Calabretta and Greisen [8]

Algorithm 2 The Perley polyhedron faceting algorithm (sketch)

Let g_f^b be all-zero complex $n \times m$ pixel grids, one per continuum band (spectral window), b, of facet f
Given (α_0, δ_0), the original field centre of the telescope (assuming field centre and phase tracking centre are the same)
for all facet centres (α_i, δ_i) **do**
  for all channels c in the continuum image band **do**
    Let uvw be a set of M measurement coordinate triples
    Let uvw' be a set of rotated uvw coordinates, applying $R(\alpha_i, \delta_i)R^T(\alpha_0, \delta_0)$
    Let vis be a set of M measurements, each corresponding to a uvw triple
    Obtain projected (assumed orthogonal/"SIN") (l_i, m_i, n_i) and (l_0, m_0, n_0) coordinates
    Let λ be the channel wavelength
    Let $p_f = \exp(2\pi i/\lambda[u(l_i - l_0) + v(m_i - m_0) + w(n_i - n_0)])$
    Let vis' = vis * p_f, by element-wise multiplication
    Call convolve_grid(g_f^b,vis',uvw')
  end for
end for
Invert g_f^b using shifted Inverse FFT

by Calabretta and Greisen [8].

Also important is that the resolution of each facet image is not arbitrary, but is subject to the same Nyquist sampling criteria outlined for regular images in Equation 4.6.

4.5 Coplanar facet imaging

One of the biggest downfalls to the non-coplanar polyhedron faceting approach is that the minor cycle deconvolution becomes very complicated: the measurement coordinates are rotated and because rotations are preserved by the Fourier transform, the PSF is also rotated. Since the PSF is not necessarily symmetric each facet has its own PSF. Combining the CLEAN components in the subtraction phase will also require careful consideration during the minor cycle.

One way around this problem would be to re-project each non-coplanar facet into a single plane after synthesis (ie. in the image-space). Doing the necessary re-projections and inevitable (and expensive) corrections for the areas where the facets overlap can be done through astronomy mosaicking software packages such as the Montage [30] suite. It is also possible to do the necessary rotation and sheering operations in the u,v space through linear[13] operations given by Sault et al. [47, Appendix A]. Although the major cycle, would, in turn, require that the regular visibilities be read off all the constituent facets' Fourier transforms, the latter approach of re-projecting the u,v coordinates and phase steering is no more expensive than the backward step.

Another way to do (approximate) coplanar faceting is to consider taking the phase error in Equation 4.19 into account while gridding the facets. One such strategy to include $w\epsilon$ per gridded visibility is to Taylor expand the term to a first order approximation around the original phase centre. The first order approximation leads to a remarkable transformation for u and v as pointed out by Kogan and Greisen

[13] A linear operation is required to conserve the Fourier relation between the sky and visibilities

[34]:

$$\begin{aligned}\epsilon &\approx \left[\frac{\partial \epsilon}{\partial l}\right]_{l_i}(l-l_i) + \left[\frac{\partial \epsilon}{\partial m}\right]_{m_i}(m-m_i) + \dots \\ &\approx \frac{1}{\sqrt{1-l_i^2-m_i^2}}(l_i(l-l_i) + m_i(l-l_i)) + \dots\end{aligned} \tag{4.22}$$

Substituting into Equation 4.15 they obtain:

$$\begin{aligned}V(u,v,w) &\approx \int\int B(l-l_i, m-m_i, n-n_i) e^{-2\pi i[u'(l-l_i)+v'(m-m_i)]} \frac{dldm}{n} \\ u' &= u - w\frac{l_i}{\sqrt{1-l_i^2-m_i^2}} \\ v' &= v - w\frac{m_i}{\sqrt{1-l_i^2-m_i^2}}\end{aligned} \tag{4.23}$$

The first order approximation approach may also be thought of as reducing the projection error, ϵ, to near zero, provided the new facet is small enough. The resulting facets are thus all (nearly) parallel. Since the w-term is merely an approximation we assume the phase steering term is the same as in the original polyhedron-faceting algorithm. Cotton[14] uses this coplanar facet synthesis method in a joint major-minor cycle deconvolution approach in Obit [18].

As one would expect the first order approximation still breaks down if the facet edges are several degrees away from the facet phase centre. Unfortunately, the second degree terms cannot easily be separated as was done for u' and v' and require using a convolution in the Fourier domain in order to multiply the approximate $w(n-n_i)$ phase screen into the image domain. As Tasse [55] points out one can additionally improve the approximation by fitting a smooth polynomial through w. Unfortunately, it should be noted that this implies that a set of w-planes has to be stored per facet, so one might as well do a hybrid approach between faceting and traditional w-projection, although the W-kernels themselves arguably are smoother for the first approach. The will become clearer once the original w-projection algorithm is discussed. After discussing w-projection we compare the error in doing a first order expansion against a full w-projection approach.

4.6 The W-projection algorithm

As already mentioned the core idea in w-projection is to multiply the intensity distribution by a w-dependent phase screen. Cornwell et al. [12] note that observations with non-zero w coordinate can be related to a single observation plane (with $w = 0$) through convolution in the measurement domain. To make this more concrete consider that a measurement of the sky brightness distribution, $B(l,m)$, is modulated by fringes defined by $\mathfrak{w}_w := e^{-2\pi i w(\sqrt{1-l^2-m^2}-\sqrt{1-l_0^2-m_0^2})}$ if the w-dependent $\mathfrak{w}$ term is separated from the usual phase term. Unlike in faceting this additional phase term cannot be taken out of the integral due to the dependence on l and m. Instead, we consider employing the convolution

[14] A word of thanks goes to Bill Cotton for useful discussions on facet imaging in late 2014

theorem to multiply the phase term into the measurement integral:

$$\begin{aligned} V(u,v,w) &\approx \int\int B(l,m)e^{-2\pi i[u(l-l_i)+v(m-m_i)]}e^{-2\pi i[w(n-n_i)]}\frac{dldm}{n} \\ &\approx \int\int B(l,m)e^{-2\pi i[u(l-l_i)+v(m-m_i)]}\mathfrak{w}_w(l,m)\frac{dldm}{n} \\ &\approx V(u,v,w=0) * \mathfrak{W}_w(u,v) \\ V(u,v) * \mathfrak{W}_w(u,v) &\rightleftharpoons B(l,m)\mathfrak{w}_w(l,m) \end{aligned} \tag{4.24}$$

Because of the w dependence of $\mathfrak{w}$, this convolution is done while gridding (and degridding) the visibilities, just as is done with the regular anti-aliasing filter, only now the filter must have complex values (as provisioned for in Equation 4.5). Just as with FFT-based narrow field imaging, it is still important to include the response of the anti-aliasing function. There is no obvious closed-form expression for the Fourier transform of $\mathfrak{w}_w\mathcal{F}[\phi]$ for arbitrary ϕ functions, so the Fourier transform of the combined function must be precomputed for several values of w. During convolution the nearest w-layer is picked, depending on the value of the w coordinate. Aside from this minor modification of the interpolation step, the rest of the previous discussion remains valid.

The immediate concern is determining how many w-planes must be precomputed. Here it is useful to note that the difference in $w(n-1)$ during the observation is responsible for the decorrelation in source brightness as explained earlier. One would therefore hope to minimize $\xi_{\max}$ by introducing N_{planes} different w-planes (here assumed to be linearly separated[15]) between the $w_{\max}$ and the 0^{th} plane, such that the difference between planes is a fraction of the image cell size:

$$N_{\text{planes}} = \frac{2\pi w_{\max}\epsilon}{\lambda_{\min}\xi} \text{ and ideally } 0 \leq \xi \ll 1 \tag{4.25}$$

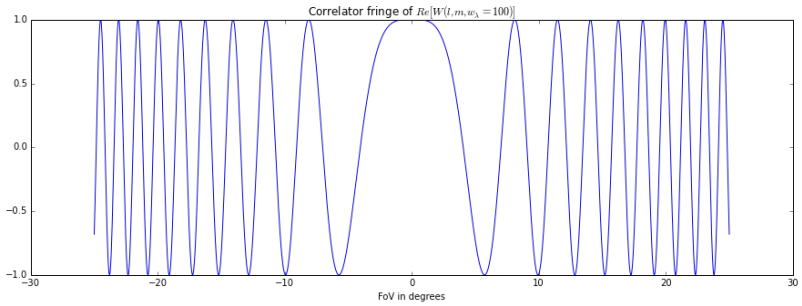
The relation above only takes positive w values into account. The planes corresponding to negative w values do not have to be computed, as any negative w value can be related to a positive value by negating the baseline vector and gridding the conjugate of the visibility. This observation allows for a significant computational (and memory) saving in the precomputation of these filters.

The next question that arises is why it may be useful to combine faceting and w-projection if w-projection can remove the wide field effects altogether. The answer lies in the computational complexity of the convolution operation itself. The complexity of the gridding step is given as MC^2 operations where C is the support size of the convolution kernel. The support size of the kernel in turn depends on the size of the image in l and m; $\mathfrak{w}$ is scaled by an ever-increasing $n-n_i$ term and therefore the w-phase screen varies faster further away from the delay tracking centre, as plotted in Figure 4.12. It is clear that larger images require convolution kernels with greater support sizes. Tasse et al. [56] give an expression for computing the necessary support size of the kernel (D_{im} is the diameter of the constructed image):

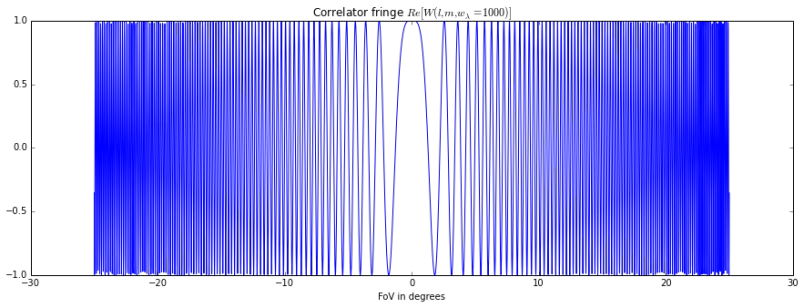
$$W_{\text{sup}} = \frac{4\pi w_\lambda D_{\text{im}}^2}{\sqrt{2-D_{\text{im}}^2}} \tag{4.26}$$

By decreasing the field of view of each of the synthesized facets, we ensure that the phase screen does not oscillate nearly as frequently as with a regular w-projection approach, and decrease the required filter support size. Additionally the $\mathfrak{w}_w$ term can be separated into two functions of l and m. If a small

[15] For arrays that are dense in the core region it may be better to consider a denser distribution of planes for lower w



(a) Low maximum w



(b) Higher maximum w

Figure 4.12: Here the real part of the w phase term is plotted. It is clearly visible that the fringe is not scaled evenly over the field of view. Instead the oscillations increase in frequency at the edge of the field of view, and have to be sampled in the image space at a rate that satisfies the Nyquest sampling criteria. As expected for larger values of w (longer baselines) the frequency of the fringe oscillations also increase.

angle approximation ($\sqrt{1+x} \approx 1 + \frac{x}{2}$) to $\mathfrak{w}_w(l,m)$ with respect to a facet centre is used, the following separable relation is obtained:

$$\mathfrak{w}_w(l,m) = e^{-2\pi i w[(l_i^2 - l^2)/2]} e^{2\pi i w[(m_i^2 - m^2)/2]} \tag{4.27}$$

Assuming the normal anti-aliasing gridding function is separable as well this becomes:

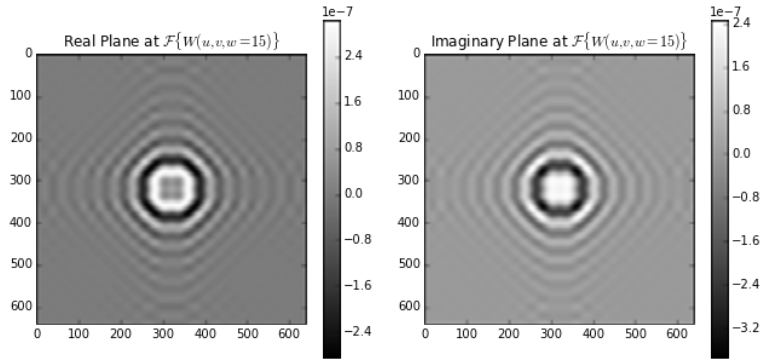
$$\begin{aligned} \mathcal{F}[\phi](l,m)\mathfrak{W}(l,m,w) &= \mathcal{F}[\phi](l)\mathcal{F}[\phi](m) e^{-2\pi i w[(l^2)/2]} e^{-2\pi i w[(m^2)/2]} \\ &= k(l)k(m) \\ &= k(l,m) \\ &\rightleftharpoons^{\mathcal{F}} K(u,v) \\ &= K(u)K(v) \end{aligned} \tag{4.28}$$

This observation decreases the memory requirements considerably, though the gridding time is slightly increased since two filter positions have to be looked up and an additional complex multiplication is necessary during gridding. Note that the memory required to store filters is given by $N_{\text{planes}} \times [W_{sup} + (W_{sup} - 1) \times (m_{\text{oversampled}} - 1)] \times \mathbb{C}$. This is a $[W_{sup} + (W_{sup} - 1) \times (m_{\text{oversampled}} - 1)]$ reduction in memory consumption, opening up the possibility of increasing N_{planes} as well as the oversampling rate. The accuracy of this small angle approximation is discussed in the next section.

To conclude this section we note that the sampling step size in the image domain (where the combination

of $\mathfrak{w}_w \mathcal{F}[\phi]$ is sampled) is again given by the Nyquist relation. Sample phase screens that include a simple sinc anti-aliasing filter are plotted in Figure 4.13.

$$\begin{aligned} \Delta^{l}_{\text{convolution step}} &= \frac{1}{2N_{\text{sup}} \frac{\Delta u}{m_{\text{oversample}}}} \\ \Delta^{m}_{\text{convolution step}} &= \frac{1}{2N_{\text{sup}} \frac{\Delta v}{m_{\text{oversample}}}} \end{aligned} \tag{4.29}$$



(a) (b)

Figure 4.13: Here the real (a) and imaginary (b) parts of the $\mathfrak{w}_w$ phase screens are plotted. In this plot $w_{\text{max}}/\lambda_{\text{min}} = 2000.000$. There are 32 planes between $w = 0$ and the maximum. The phase screen is simulated for a 5.1° square image, with its half support size at 15 cells and $m_{\text{oversample}} = 20$. Here, only a plane near the centre of the w range is plotted. Some aliasing effects are noticeable due to undersampling (due to memory constraints).

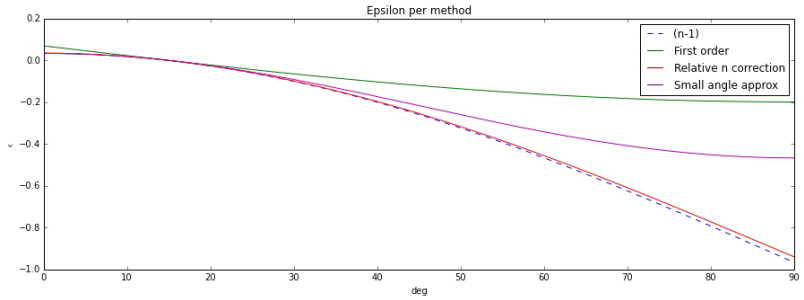
4.7 Error estimations

Several methods of creating copanar facets have been discussed in the previous section. We give comparative plots in Figure 4.14 where each of the methods is compared to a traditional w-projection approach.

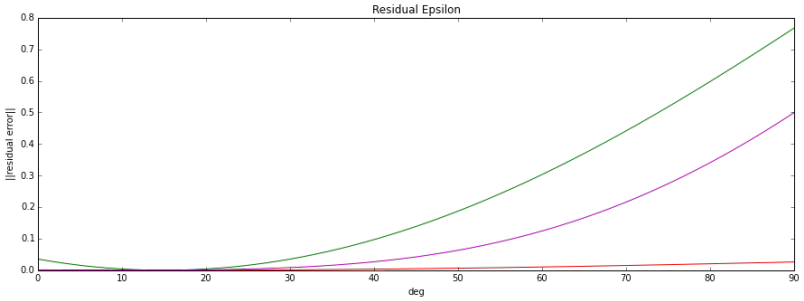
4.8 Revisiting the direction-dependent effects

Traditional calibration pipelines assumes that the same "apparent sky" is sampled by all antennae, and attempts to solve only the unknown direction-independent gain terms. This process is known as *self-calibration*. Furthermore some packages like the CASA framework consider directional dependent terms as simple effects that do not vary in time and is identical per antenna. The addition of directional dependencies within the all-sky integral, primarily those caused by the ionosphere and modulation effects by the primary beam, violates this premises.

The self-calibration process relies on a knowing some aspects of the sky and can include one or more well-described sources. The directional independent and directional dependent terms can be solved for by a fitting the predicted model visibilities to the observed data [38]. Only adjusting the gains based on directional-independent effects cannot remove the complex polarization effects caused by the primary



(a)



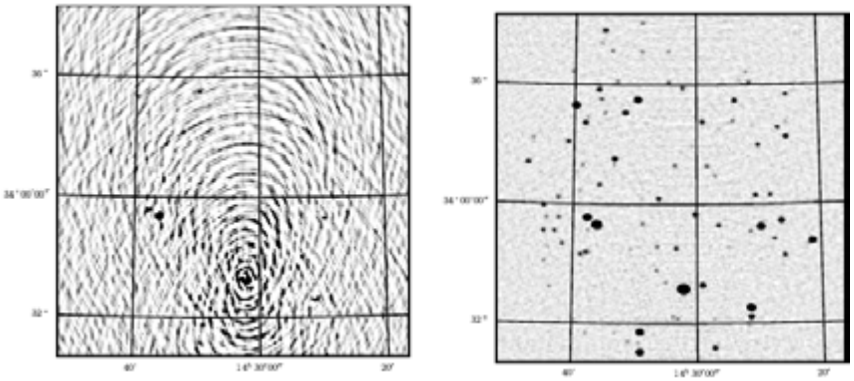
(b)

Figure 4.14: In these two plots the phase error introduced by the orthogonal projection is plotted; the dashed line is a plot of $n-1$ as would be convolved into the image by classic w-projection; a full w-projection approach is assumed to be the most accurate way of dealing with the orthogonal projection geometry. In green we show the deviation between a first order Taylor approximation and w-projection[34]. Then we show the small angle w-projection suggested above in magenta; it clearly performs significantly better than the first order approximation, only breaking down tens of degrees from the phase centre. Lastly we also plotted the improved facet-centre-relative w-kernel approximation to ϵ that Cyril Tasse [55] suggested as an improvement on the work of Kogan and Greisen [34] in red. Cyril Tasse's polynomial approach [55] performs even better than the latter. The error between w-projection and the three approaches is plotted in (b).

beam, nor can it solve for unknown slow-varying directional dependent gain terms. If we, for the moment, only consider the known polarization effects of the antenna beam (which can be modeled or measured using holography) it becomes clear that some sources will be resolved more than others for the same amount of observation time, depending on their position in the sky. In practice, the beam pattern is also anisotropic; one possible cause can be the struts above a prime-focus antenna. If the antennae are placed on alt-azimuth mounts the sky rotates with respect to this anisotropic beam pattern over the course of an observation. This results in sources that is only partially resolved outside of the primary beam. The correction of this effect alone is not a trivial undertaking - as the reader may suspect one possible solution is to attempt to "invert" the directional dependent effects on the visibility inside the convolution integral of the RIME.

Since these effects vary with both direction and time, removing them is a tricky proposition; at every timestep only a handful of points from these convolving functions are sampled! More recently several solutions to solve for slow-varying directional-dependent effects using self-calibration and removing known effects through calibration have been proposed. These include solving direction dependent effects through the method of differential gains, peeling and A-projection. Peeling involves iteratively removing the effects from the brightest sources through direct Fourier approaches, while differential gains simultaneously solves for the gain effects from bright and faint sources [50, 51]. When dealing with known

(or modeled) directional-dependent effects the A-projection algorithm [2] has proven very successful in removing the polarization effects contributed (predominantly) by the primary beam for the LOFAR array (see figure 4.15). Refer to the LOFAR imager implementation by Tasse et al. [56] for a full mathematical treatment of the algorithm.



(a) Cleaned w-projected image (b) Cleaned aw-projected image

Figure 4.15: Figures (a) and (b) shows the result of correcting for the complex polarization leakage patterns introduced by the LOFAR individual element beams and phased-array stations

In A-projection the "onion-form" of the RIME stated before can be rewritten in terms of 16-element Muler matrices, for which first-order inverses can be computed. Provided that the 16 element terms vary slowly over the sky the inverses can be sampled at a limited number of support points (far fewer than the w term stated previously) and can be convolved as part of the inversion step. A-projection does however require a significant amount of memory and time to precompute the 16 baseline-dependent (due to individual antenna pointing error) convolutions per processed visibility. As the number of baselines grow as the square of the number of antennae in this approach fast becomes prohibitively expensive [56].

An alternative strategy that can prove useful is to consider an amended faceting approach[16]. Here the direction-dependent effects are assumed to stay constant over a small area of the sky (centred at some l_i, m_i) and the Jones matrices may simply be inverted and applied as part of the part of the resampling process:

$$\begin{aligned} B_{\text{corrected,dirty}} &= \mathcal{F}^{-1}(D_p^{-1}(l_i, m_i, t, \nu) V_{\text{obs}}(u, v, t, \nu) D_q^{H^{-1}}(l_i, m_i, t, \nu)) \\ &= \mathcal{F}^{-1}(D_p^{-1}(l_i, m_i, t, \nu) V_{\text{obs}}(u, v, t, \nu) D_q^{-1^H}(l_i, m_i, t, \nu)) \end{aligned}$$

This approach has the additional advantage of being arbitrarily accurate: as the facet image size is decreased the inversion step becomes a per-pixel corrected direct Fourier inversion. This makes this approach a viable alternative to A-projection. However, it must be stressed that the computation cost of both inversion and prediction steps rises sharply when creating polarization-corrected images with either approach: all four correlations must be gridded in stead of taking either only the parallel or cross-hand terms into account when doing traditional imaging. This effectively means that the number of floating point operations required by resampling effectively quadruples (the work in the inner most loop is four times more) without considering the additional conjugates taken and matrix multiplications for each observed visibility. Using this approach the additional storage required to store the 2x2 Jones matrices

[16] Suggested by Cyril Tasse and Oleg Smirnov

grows approximately as:

$$N_{\text{Jones}} \approx N_{\text{sources}} N_{\text{integration steps}} N_{\text{channels}} \sqrt{N_{\text{Baselines}}}$$

4.9 Computational considerations

We now draw a comparison of the computational complexities of the methods discussed here. Much of this discussion is taken from the detailed analysis of Yashar & Kemball [65]. The data rates produced by interferometers is roughly [17] given as:

$$r_{\text{datarate}} \approx \frac{1}{1024^4 \tau_{\text{integration}}} N_{\text{baselines}} N_{\text{channels}} N_{\text{bands (spw)}} N_{\text{corr}} \text{sizeof}(\mathbb{C}) \text{ (TiB/s)} \tag{4.30}$$

Here $\tau_{\text{integration}}$ is the correlator integration time, which depends on the effective resolution of the instrument and the rotation speed of the earth. The maximum integration time must be short enough to ensure sources don't move more than the angular resolution of the telescope:

$$\tau^{-1}_{\text{integration}} = Q_t \frac{|\vec{b}_{\text{max}}|}{D_{\text{antenna}}} \omega_{\text{earth}}, \text{ where } \omega_{\text{earth}} \approx 7.29 \times 10^{-5} \text{rad/sec}_{\text{Mean sidereal}} \tag{4.31}$$

Observing a band-limited range of frequencies causes smearing in the image. Although a good signal to noise ratio depends on observing a large band of frequencies the smearing is limited by keeping bandwidth, $\Delta\nu$, of each channel narrow, centred at ν_i. To compensate for the associated decrease in the signal to noise ratio of the observation many adjacent channels may be integrated if sources of continuum emission are being observed[18]. In addition multiple channel bands (spectral windows) may be observed simultaneously. For continuum imaging the number of channels required is at minimum[19]:

$$N_{\text{channels}} = Q_c \frac{|\vec{b}_{\text{max}}|}{D_{\text{antenna}}} \frac{\Delta\nu}{\nu_i} \tag{4.32}$$

In both number of channels and integration time Q_t and Q_c are quality control factors.

Next we use slightly different equations for N_{planes} and N_{facets}, in order to at least sample n at the Nyquist rate in 3D imaging, W-projection (and W-stacking) and facet imaging. These equations assume only the primary beam is being imaged at the full effective resolution of the array (see e.g. Perley [57, Lecture 19]) for details on the derivation). Yashar & Kemball [65] also use a different relation for the

[17] There are additional terms such as tapering weights, flags and metadata to consider, but these are ignored for now

[18] We assume the intensity distribution of these sources do not vary significantly with frequency. This is in fact not true for wider bands; here it may be necessary to account for the drop in emission intensity, especially when deconvolving (using a Multi-frequency approach, see for instance Conway and Sault [57, Lecture 21])

[19] It should be stressed that this number only limits the smearing in images. The actual number of channels used also depends on the science being done; spectral line imaging is just one instance where many more densely-packed channels may be required.

support of the convolution filters for W-projection.

$$\begin{aligned} N_{\text{planes}} &\approx \frac{\lambda_{\max}|\vec{b}_{\max}|}{\xi D_{\text{antenna}}} \\ N_{\text{facets}}^2 &\approx \left(\frac{2\lambda_{\max}|\vec{b}_{\max}|}{\xi D_{\text{antenna}}}\right)^2 \\ w_{\text{sup}}^2 &\approx \left(\frac{2\lambda_{\max}|\vec{b}_{\max}|}{D_{\text{antenna}}}\right)^2 \end{aligned} \tag{4.33}$$

Table 4.1 outlines the computational complexities for the backward synthesis step only. Some of these methods may have additional computational costs for prediction, deconvolution and reprojections. For convenience we summarize the advantages and disadvantages of the various approaches here:

- The cubed Direct Fourier Transform is a per-voxel complex exponential and multiplication. The real sky lies on a unit sphere in the cube. The approach is prohibitively expensive both in terms of memory and computational complexity (M scales as N^2), though arbitrarily accurate.

- In the FFT-based 3D Imaging approach each plane has its visibilities phase-steered, interpolated onto a grid and Fourier transformed separately using a 2D Fourier Transform [57, Lecture 19]. Alternatively if there are enough planes in the n dimension the visibilities can be interpolated directly into a cube and a three-dimensional Fast Fourier Transform taken [65]. This approach also has prohibitive memory requirements for arrays with very long baselines.

- A faceting approach requires that the set of visibilities are phase-steered and sampling coordinates transformed while gridding before a Fourier transform of each of the smaller fields are taken. Each field have approximately $N_f^2 = \left(\frac{0.4|\vec{b}_{\max}|}{D_{\text{antenna}}}\right)^2$ pixels for $\xi = 0.2$.

- W-projection becomes expensive with a large field of view, but can be faster than w-stacking for a small number of visibilities [40]. Keeping many oversampled convolution kernel layers in memory is one drawback of this method, unless a small-angle approximation can be applied to the kernels, or the support size of the convolution filters is limited in the filters representing the lower w-layers.

- In W-stacking each visibility is gridded with the usual gridding convolution function ϕ, but to different planes depending on w. Each plane is then Fourier transformed and multiplied by a complex phase screen. This approach works well for large sets of visibilities where the FFT costs per grid is negligible. The approach can be demanding in terms of memory depending on the number of w-planes being gridded. This constraint may require the implementation to presort and grid only a subset of layers at a time, increasing disk access. The approach is faster than w-projection for larger fields of view at lower elevation angles. Offringa et al. [40] also shows that w-snapshots is only faster than w-stacking for very large fields of view at lower elevation angles.

Approach	**Computational complexity (backwards step only)**
(Cubed) Direct Fourier Transform	MN^3
3D Imaging using FFT	$N_{\text{planes}}(M + MC^2 + 2N^2 \log N)$
Traditional non-coplanar faceting	$N_{\text{facets}}^2(M + MC^2 + 2N_f^2 \log N_f)$
W-projection	$Mw_{\text{sup}}^2 + 2N^2 \log N$
W-stacking	$MC^2 + N_{\text{planes}}(2N^2 \log N + N^2)$

Table 4.1: Computational complexities of various wide field correcting approaches

From the estimates above for number of projection planes and facets it is clear that w-stacking will outperform faceting if all the planes can be kept in memory and the gridding costs exceed Fourier transformation costs. However, as already pointed out faceting has more moderate memory requirements than w-stacking and can be used as a method for correcting Directional Dependent Effects, provided the facets are small in comparison to the variation of the effects across the field of view.

4.10 Review of previous literature

The imaging pipeline has been investigated extensively in recent years. In particular the gridding operation has been parallelized multiple architectures including CPU [40, 23, 19], GPU [28, 46, 37, 21], as well as the CELL/B.E. processor [61].

Both scatter- and gather-based GPU gridding algorithms have been investigated in the literature. In gridding scatter-based approaches lead to race conditions between threads when updating grid cells, since multiple visibilties may be interpolated over the same grid cells. Edgar et al. [21] used a gather approach where threads would scan through a subset of visibilities and add those that contribute to a the grid point in question. Edgar et al. attains a 22x speedup compared to a CPU implemented pipeline.

Humphreys and Cornwell [28] benchmarked a scatter-based GPU w-projection gridder that assigns a thread per convolution filter tap and can grid multiple visibilities in parallel when there is no overlap between the convolution regions. They report figures up to 3.5 Giga grid point additions per second (abbreviated as "G" is the number of grid point updates - C^2 updates per visibility record) on a Tesla C2070 compared to just over 1.5 G on a multicore Intel Xeon X5570 CPU architecture.

Romein [46] presents a novel scatter distribution strategy, where global memory accesses are limited by accumulating visibilities in register memory instead of atomically updating each grid points for every visibility record. The locality of consecutive integration periods on the loci of each baseline is used to accumulate consecutive visibilities to great succcess. Muscat [37] uses this distribution strategy in a new imager called the *Malta Imager*. He achieves gridding rates of around 55-60G when gridding 4 correlations. He also shows that the main limiting factor in this distribution strategy is the time spent looking up convolution filter values and extends Romein's strategy by accumulating measurements that have a high probability of being equal. When this "compression" mode is enabled he achieves gridding rates more than 250G. We will be using an adapted strategy based on Romein's distribution strategy in this thesis. The exact algorithmic details will be given in the design chapter.

The CPU-based parallel strategies are somewhat different to the GPU-based strategies. A simple data-parallel approach in spectral imaging consists of splitting the image cube up between threads as is done by Humphreys and Cornwell [28] for their CPU-based benchmark. A data-parallel approach for continuim imaging consists of keeping multiple copies of the uv grid and reducing the grids into one average afterwards. This is not feasible when the uv grids become very large. Goulap [23] investigates several CPU-based gather approaches applicable to w-projection and show that this can achieve near-linear speedups (compared to a serial implementation). When doing faceting another lock-free data-parallel approach is to process each facet in parallel, ensuring lock-free accumulation to each facet uv grid. Bill Cotton [19] achieved speedups up to 50% using data-parallel approaches.

Chapter 5

Bullseye: A parallel targeted facet imager

5.1 Design objectives

The primary design objective of this work is to build a scalable parallel facet-based imager. A full deconvolution pipeline is currently out of scope, instead the focus is on accelerating the gridding step, since this step will be called on multiple times in a major-minor cycle deconvolution pipeline, as discussed in chapter 4. To this end we will focus on comparing performance between parallel CPU-based resampling and a GPU approach.

5.2 Architecture and implementation

We have decided to split the implementation into two major components:

1. A front-end program dealing with the logic of reading in measurement data, dealing with user options, overall program flow and image finalization.
2. A set of back-end libraries that implement a common interface and house the resampling and transformation routines. The resampling routines include options to resample multiple correlations and enable faceting and w-projection logic.

The architecture above allows one set of resampling routines to be easily swapped out for another and a comparison between CPU and GPU-based implementations. Figure 5.2 shows the major components of our imager, along with several major dependencies.

We opted to implement the front end of the program using Python 2.7 and extension packages due to the ease it provided in reading of Measurement Sets, user option parsing and writing to FITS files. The backend libraries containing the resampling routines have to be implemented in C++ to be efficient and have a common ctypes interface that can be called from Python. We opted to implement the backend libraries using constructs from the C++ 11 standard. The libraries contain a set of templates shared between CUDA and CPU code to implement the resampling routines for the various use cases of the

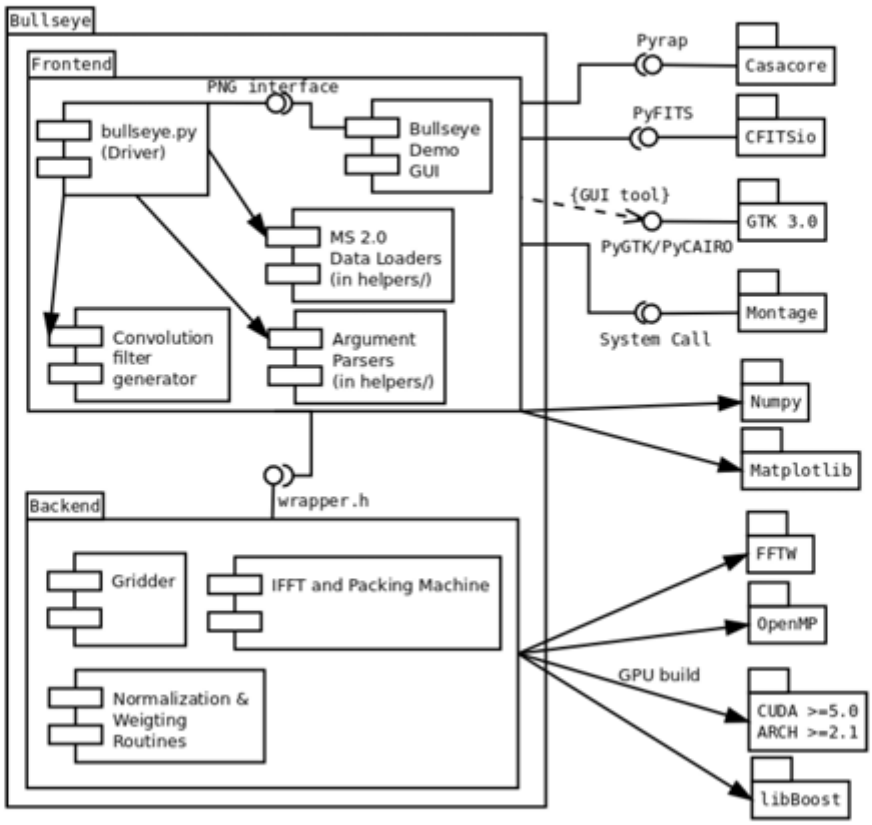


Figure 5.1

Figure 5.2: The front end of the solution contains a command line utility that has very similar options to that provided in other imagers, such as lwimager. Included with it are modules required to generate the w-projection convolution functions and a prototype graphical interface illustrating targeted faceting. The backend consists of the resampling routines along with routines for Fourier shifting, inversion and normalization on a set of facets that may each contain multiple spectral images.

imager. Both the CPU and GPU code therefore have to be compiled with the NVIDIA NVCC compiler (versions 5.0 or above) toolkit.

5.3 Normal workflow

During imaging the user will supply a set of facet centre coordinates or the number of facets splitting the sky (or both), along with a measurements database and image dimensions. The imager outputs a set of facet images that can, optionally, be recombined.

Program flow is indicated in Figure 5.3. The resampling and fourier inversion step may include sampling and transforming the sampling function. The latter requires that all measurements are set to unity. With this option enabled a PSF is synthesized for each facet image. Additionally, each facet may have resampling grids allocated for multiple correlations and spectral bands, and can therefore be facet cubes instead of simple 2D images.

The fourier inversion step also entails shifting each of the facet grids such that the base uv-frequency of each grid is located in the middle of the grids, thus shifting the image phase centre to the middle of each

of the images.

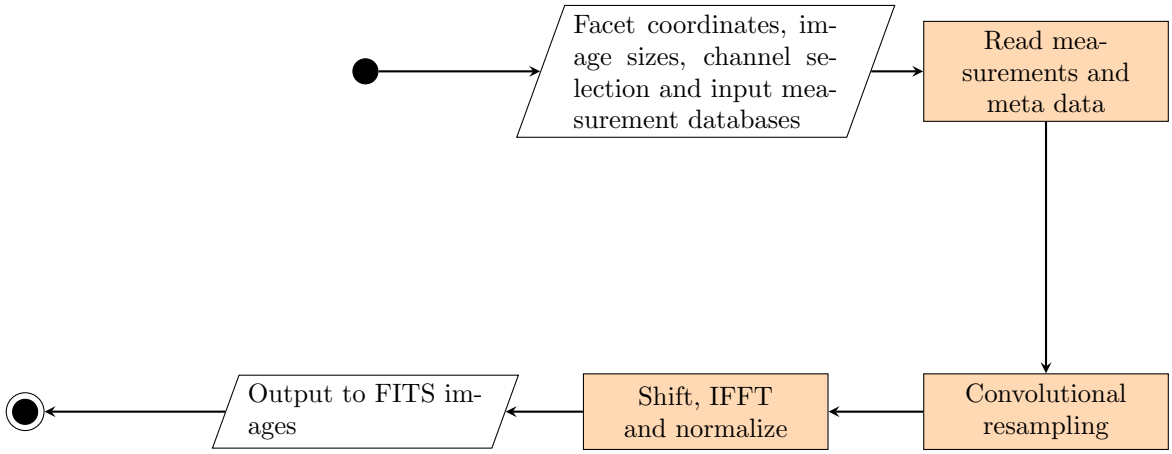


Figure 5.3: This diagram shows the major steps involved in the synthesis process. The convolutional resampling step includes faceting transforms and w-projection logic.

5.4 Input/Output formats

The Measurement Set standard [63, 62] is an AIPS++ (later CASA) database format containing telescope observation data, along with observation metadata. Correlated observation data is stored in a "MAIN" table, along with uvw coordinates, antenna identifiers, timestamp information, weighting and flagging information and foreign keys to subtables with metadata for the observation. The metadata describes everything from the antenna positions and mounts to feeds, spectral window descriptions and the observed fields. The standard database schema is summarized in Figure 5.4.

Each row in the MAIN table contains measurements for a particular spectral band and may contain multiple correlations per spectral channel. Each row contains $N_{\text{channel}} \times N_{\text{correlation}}$ measurements. The rows are not necessarily ordered by baseline or time by default. In total, the Measurement Set MAIN table will contain $N_{\text{baseline}} \times \frac{\tau_{\text{observed time}}}{\tau_{\text{integration length}}}$ rows for each of the observed fields. The number of baselines includes those rows contributed by auto-correlated antennas. It is worth noting that during flagging and calibration some rows may be deleted from the Measurement Set.

Bullseye is designed to handle observations split into multiple Measurement Sets as input. When more than one Measurement Set is specified as input it is assumed that the two measurement sets contain observations of the same set of fields and that all the metadata remains the same between the two databases.

Our facet imager has to output a series of image cubes, one per facet. Each facet cube is a 3-dimensional array containing continuum images at multiple frequency bins, where several channels in the measurement set may be averaged into each of these slices. The Flexible Image Transport System (FITS) [43] format has become the defacto standard for data sharing between observatories. A FITS file can be used to store an N-dimensional data cube where N can range between 1 and 999 and is structured as a series of Header Data unit blocks, each 2880 bytes in size. Each header contains a series of 80 character key-value pairs that describe the coordinates of each of the axes, along with projection and transformation

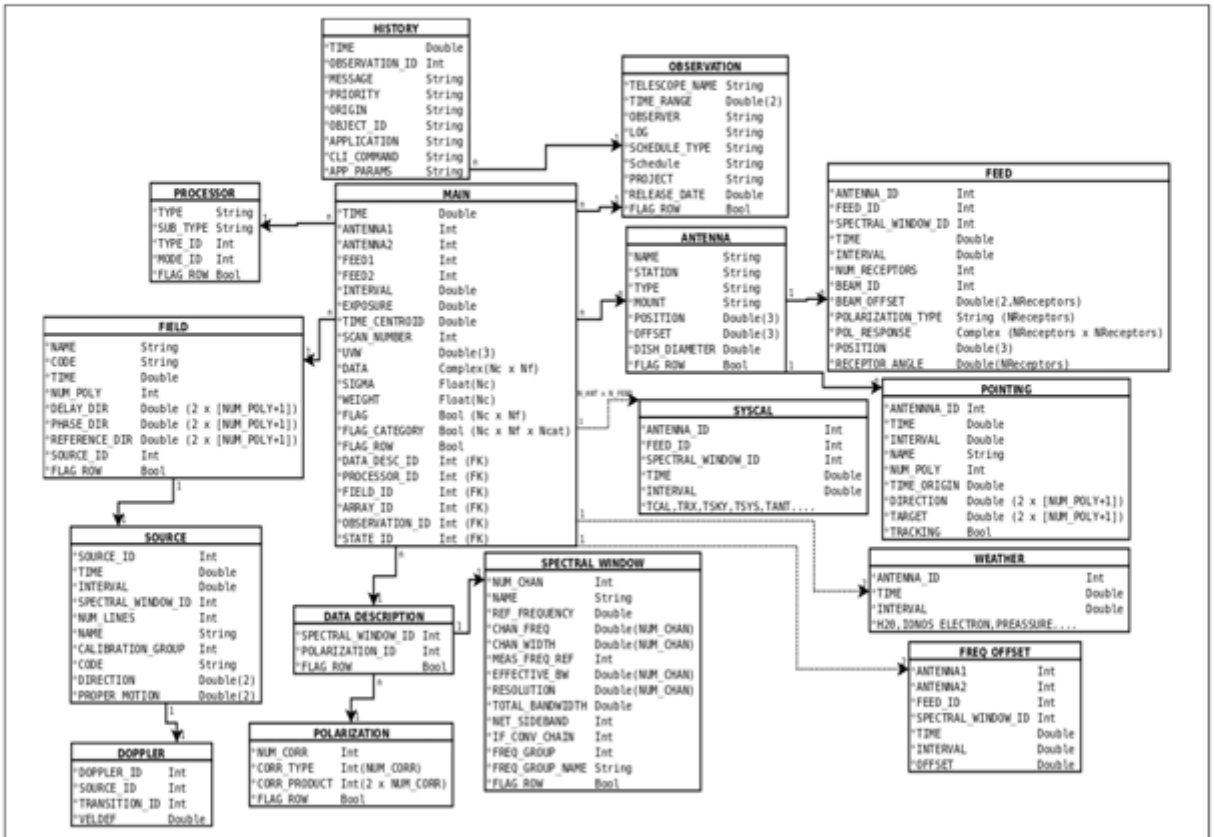


Figure 5.4: This diagram depicts the standard layout of a Measurement Set database. Only some of the optional fields are included in this diagram, for instance the MAIN table may have multiple data columns and per channel weights. Only the most important relations are shown in this diagram. Data can be queried using a SQL-like language known as TaQL.

information.

Since the primary goal of our imager is to make facet images, we choose to use the orthogonal projection coordinate system for the l and m coordinates. The orthogonal projection results in coordinate distortions away from the projection pole. Since the primary goal of our imager is to create narrow-field facet images, using this projection is justified.

5.5 Parallelizing data precomputation and resampling

5.5.1 Disk I/O vs. compute

Although it is important to consider the computational costs involved with the resampling operation it is also necessary to take the latencies of loading and preprocessing into account. When performing narrow field imaging, the costs of disk I/O alone can amount to a significant portion of the run time. This is less of a problem when performing faceting and w-projection, but cannot be ignored. In order to mitigate this latency we opt to split the data into many pieces, making it possible to load the next chunk into buffers while still imaging data already loaded and preprocessed as illustrated in Figure 5.5.

When performing wide-field imaging with a large number of facets and/or w-projection with filters of

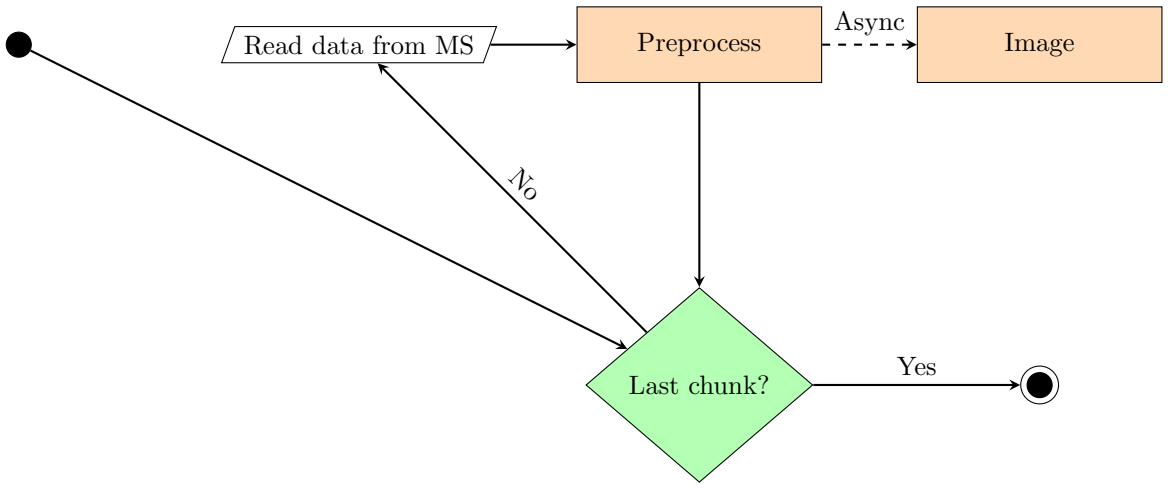


Figure 5.5: Asynchronous compute while data is buffered. The imaging algorithms are invoked on a background thread while the main program thread continues to buffer in the next chunk of data. We use a thread barrier mechanism to ensure that the previous round of imaging has completed before starting to image the next chunk.

large support the latency involved with disk I/O is effectively hidden by this approach, and the runtime becomes bound by the compute time as indicated in Figure 5.6. All but the cost of loading the first chunk of data is hidden by this strategy.

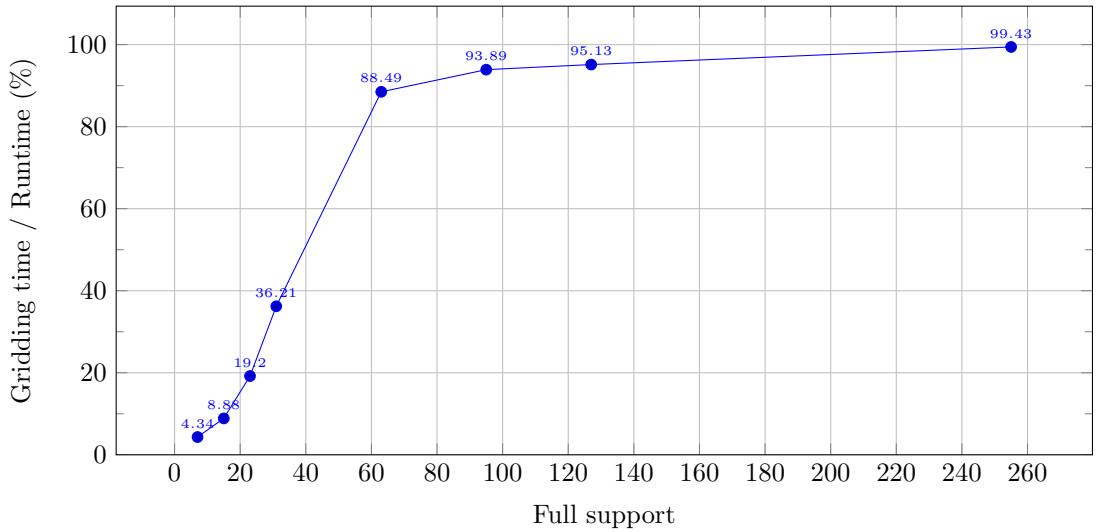


Figure 5.6: Typical gridding time (CPU-based w-projection) is plotted against runtime. Disk I/O and associated total runtime may be slightly lower depending on drive speeds, but these figures are for a typical desktop hard drive at the time of writing.

5.5.2 The CPU-based resampling algorithm

The requirement of creating a field containing multiple facets provides a course-grained approach to parallelizing the resampling process, without the need for synchronization and locks. This parallelization

strategy works well when there are at least as many facets as CPU cores on the system, especially when the number of facets is close to a multiple of the number of CPU cores, balancing the workload between cores. This parallelization over facets is implemented using the lightweight OpenMP framework of macros.

The resampling algorithm should cater for gridding multiple correlations and enabling faceting and w-projection-based resampling when required. To achieve this efficiently and with maximum reuse of code the resampling algorithm is implemented using a set of traits and policy C++ templates. Algorithm 3 is simplified pseudo-code, but explains the core resampling steps on a CPU.

If the channels all contribute to different grids in the cube this can provide another avenue of parallelization. However, we assume that the input data is used to construct a set of continuuim images per cube. As such the wavelengths used in the algorithm to scale the uv tracks are those provided in the measurement set and are in the topocentric frame of reference. As such the imager should not be used to create spectral line images where it is essential that the frame of reference is stable during the course of the observation. Such observations require that the uvw coordinates are corrected for the dopler-shifts introduced by the Earth's rotation around its axis and around the sun. This is an expensive time-dependent correction and provided by the casacore libraries. The correction is not used at present.

The innermost loops compute and writes values to consecutive grid locations and are responsible for the quadratic scaling factor that makes w-projection-based resampling such an expensive operation. This opens up the opportunity to further parallelization on CPUs that support vector processing operations through SSE and AVX, see Chapter 2. The GCC compiler suite does not automatically vectorize the code in the inner-most loop during automatic optimization due to its depth. We therefore had to vectorize the code for the innermost loop by hand for the seperate cases of gridding 1, 2 and 4 cross correlations. Vectorization will have the greatest impact when performing w-projection and therefore only this use case was vectorized. Instead of writing separate SSE and AVX versions of the code, we decided to only support modern processors that include the AVX instruction set. Older processors can be targeted during compile time by turning off the vectorizated code.

The vectorized instructions load the original values from consecutive grid positions, parallelize the complex multiplications and additions, finally storing the values back to the grid positions in memory. Note that both the optimized load and store instructions require that the index of the first float be aligned to 16-byte boundaries. Since the rounded u,v coordinates can fall anywhere on the grid this alignment cannot be guarenteed, even if the grid memory is aligned when allocated. It is therefore necessary to use unoptimized load and store operations, which will hamper performance to some degree.

5.5.3 The GPU-based resampling algorithm

The GPU work distribution strategy is more complicated than that of the CPU. A GPU co-processor has thousands of processing cores and it is therefore imperative to spawn enough threads to fully occupy all the cores of the GPU. Furthermore, Kepler generation devices can perform about 60 floating point operations in the time taken to read 1 floating point value from memory [16]. This highlights the importance of limiting the total global memory accesses made by the gridding algorithm. The approach we choose for our implementation is a distribution strategy first suggested by John Romein [46].

His strategy exploits the spacial locality between consecutive measurements made by the same baseline. As long as the integration time between consecutive measurements is short the measurements should

Algorithm 3 CPU facet-based convolutional resampling

Allocate a complex grid g of size $n \times m$ pixels for each facet and cube slice
Assume N_{rows} of complex visibilities, uvw coordinates, weights and flags are read from database
Let c be a padded complex filter with N_{planes} w-layers
By scaling the uv tracks the IFFT is scaled to the desired field of view (similarity theorem):
Let u_{scale} = ARCSEC_TO_RADIANS($n \times cellsize_l$)
Let v_{scale} = ARCSEC_TO_RADIANS($m \times cellsize_m$)
for (in parallel) $f \in [0...N_{\text{facets}})$ **do**
  for $r \in [0...N_{\text{rows}})$ **do**
    if $field[r]$ not being gridded **or** $rowFlagged[r]$ **then**
      continue
    end if
    for $q \in [0...N_{\text{channels}})$ **do**
      if $channelFlagged[r, q]$ **then**
        continue
      end if
      for $x \in [0...N_{\text{cross correlations}})$ **do**
        Let cube_slice be the index of the grid frequency $spw[r] \times N_{\text{chan}} + q$ is to be accumulated to
        Let $u, v, w = \frac{u[r] \times u_{\text{scale}}}{\text{wavelength}[q]}, \frac{v[r] \times v_{\text{scale}}}{\text{wavelength}[q]}, \frac{w[r]}{\text{wavelength}[q]}$
        First apply facet phase steer to original scaled uvw coordinates and orthogonal lmn coordinates:
        Let $p_f = \exp\left(2\pi i/\lambda[u(l_i - l_0) + v(m_i - m_0) + w(n_i - n_0)]\right)$
        if doing polyhedron faceting **then**
          Tilt the facet by rotating the uv plane:
          Let uvw' be a set of rotated uvw coordinates, applying $R(\alpha_i, \delta_i)R^T(\alpha_0, \delta_0)$
        end if
        Let $u'_{\text{int}}, v'_{\text{int}}$ be the rounded (to nearest integer) u', v'
        Let $u'_{\text{frac}}, v'_{\text{frac}} = -u' + u'_{\text{int}}, -v' + v'_{\text{int}}$
        Let $vis = vis[r, q, x]$
        Instead of storing convolution kernels for negative w grid the complex conjugates of the baselines:
        if $w' < 0$ **then**
          Let vis =conjugate(vis)
          Let $u', v', w' \times = -1$
        end if
        Assuming linear spacing between sampled w layers:
        Let $w_{\text{plane}} = \text{round}(w'/w_{max} \times (N_{\text{planes}} - 1))$
        if $u'_{\text{int}}, v'_{\text{int}} \pm c_{\text{half support}}$ within grid boundaries **then**
          for $v_{\text{tap}} \in [-c_{\text{half support}} ... c_{\text{half support}}]$ **do**
            for $u_{\text{tap}} \in [-c_{\text{half support}} ... c_{\text{half support}}]$ **do**
              Let $c_{\text{weight}} = c[(u_{\text{tap}} + c_{\text{half support}} + u'_{\text{frac}} + 1)) \times c_{\text{oversample}}, (v_{\text{tap}} + c_{\text{half support}} + v'_{\text{frac}} + 1)) \times c_{\text{oversample}}, w_{\text{plane}}]$
              Let $g[u'_{\text{int}} + u_{\text{tap}} + \frac{n}{2}, v'_{\text{int}} + v_{\text{tap}} + \frac{m}{2}, \text{cube\_slice}] += vis \times p_f \times c_{\text{weight}}$
            end for
          end for
        end if
      end for
    end for
  end for
end for

fall on the same grid points. The strategy accumulates the values that fall within the same grid cell in a register before writing the value out to global memory when the uv track moves onto the next grid point. Each thread is assigned one filter tap in the support area of the convolution filter and one of each of these groups are assigned to a baseline. Many baselines can therefore be gridded in parallel, assuming the write accesses to global memory are atomic so as to prevent race conditions between two or more baseline thread groups.

The approach assumes, of course, that the data is grouped per baseline and then ordered by time. The data within a measurement set is not generally ordered in this way and it is necessary to repack the data as part of the loading and preprocessing step, as indicated in Figure 5.7.

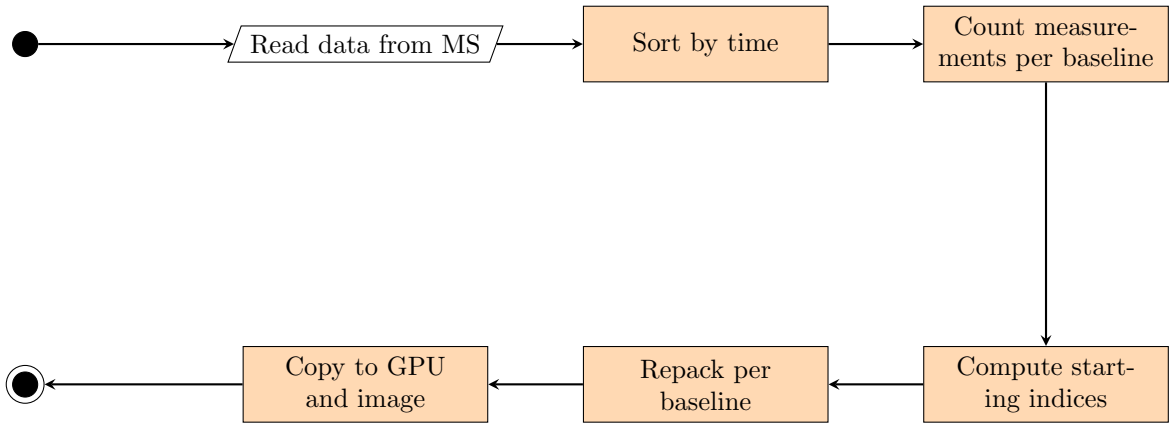


Figure 5.7: The preprocessing workflow necessary for gridding using Romein's distribution strategy.

After reordering the entire measurement set by time it is important to repack the data by baseline. Not all baselines have the same number of timesteps (some may have been split out of the measurement set during the flagging and calibration process), requiring that a variable-length array be allocated per baseline. Such an array of variable length arrays is not easy to transfer onto the GPU and we instead opt to flatten this array into a one dimensional structure, storing the positions of the first time step of each baseline as a separate array. This approach requires a single pass over the data counting the number of timesteps per baseline. A running accumulation (or *prefix scan*) of these counts is then computed, yielding the starting indices of the baseline subarrays. The baseline index is not stored directly in a measurement set: only the id's of the antennae are stored. The unique identifier for each of the baselines used during counting has to be computed and is given by the following quadratic series:

$$i_{\vec{b}=a_2-a_1} = (-S^2 + 2 \times S \times N_{\text{antennae}} + S)/2 + |a_1 - a_2|, S := \min(a_1, a_2) \tag{5.1}$$

The measurement set only stores unique measurements and not their conjugate counterparts, and hence there are only (at maximum) $\frac{N_{\text{antennae}}(N_{\text{antennae}}-1)}{2} + N_{\text{antennae}}$ measurements per timestep, including the autocorrelated measurements of each of the antennae.

The prefix scan operator (for the normal associative binary addition operator) is defined as:

$$\text{prescan}(C)[i] = \begin{cases} 0 & i = 0 \\ \sum_{k=0}^{i-1} C[k] & i > 0 \end{cases} \tag{5.2}$$

It is therefore necessary to allocate an array to store the baseline counts with $N_{\text{baselines}} + 1$, setting the last element to zero, in order to store the starting indicies of all the baselines and implicitly store the number of elements contained in the sub-array of the last baseline.

Once the starting indexes are computed the repacked arrays are transferred onto the GPU and when completed the gridding GPU kernel is launched asyncronously, while the next set of uv coordinates and visibilities are read and the required preprocessing started, similar to the CPU implementation. It is also important to mention that the correlations not being gridded are stripped out of the visibility array before transfer during the preprocessing stage to reduce the transfer time.

The resampling approach taken on GPUs is presented in Algorithm 4. The kernel, as stated in the algorithm, is dispatched to $c_{\text{full support}} \times c_{\text{full support}} \times N_{\text{baselines}} \times N_{\text{facets}}$ threads when launched and broken into blocks of threads a multiple of the CUDA device *warp length* in size (refer to the discussion on how parallel work is laid out in CUDA in Chapter 2). Given enough registers for each thread this approach should achieve high occupancy on the GPU. It may be necessary to adjust the maximum number of registers per thread through command line arguments to the compiler to ensure that the maximum number of registers available per Streaming Multiprocessor is not exceeded, depending on the targeted device.

5.6 GPU filter caching option

The GPU has several on- and off-chip memory types and caching mechanisms, as already discussed. Due to the spatial locality of the convolution filter lookups it is advantageous to store the filter values in a memory that is cached on chip. GPU texture memory is off-chip read-only memory, but 48 KB (Kepler architecture) is cached on-chip and is ideal for storing convolution kernels.

When convolving with seperable filters like those used in narrow-field imaging this memory is more than enough to store highly oversampled filters. However, as already pointed out, w-projection filters are not generally separable filters. This results in high memory usage when the oversampling factor and number of w layers are increased. However, for the GPU implementation we assumed that w-projection will always be used in conjunction with faceting, since coplanar w-faceting is the primary mechanism by which our imager removes the widefield effect. As we already showed w-projection filters are approximately seperable when imaging over narrow fields of view, and these filters have a much better chance of fitting into texture cache. Our GPU w-projection implementation assumes that the seperable filter implementation is accurate enough for the user and it is important that faceting always be enabled when imaging on the GPU.

We have provided an alternative cache-less implementation of the GPU imager for comparison and compatibility. If the stack of filters is too big to fit into memory the caching code can be turned off at compile time. We have seen a 20% drop in performance on compute 3.0 hardware with this option disabled.

5.7 Precision

During the course of an observation the gridding operation accumulates thousands of complex-valued visibilities into a set of grid bins. Since this operation is done in finite precision, the summation of such a

Algorithm 4 GPU facet-based convolutional resampling

Allocate a complex grid g of size $n \times m$ pixels for each facet and cube slice
Assume N_{rows} of complex visibilities, uvw coordinates, weights and flags are read and transfered to GPU
Let c be a padded complex filter with N_{planes} w-layers
By scaling the uv tracks the IFFT is scaled to the desired field of view (similarity theorem):
Let $u_{\text{scale}} = \text{ARCSEC\_TO\_RADIANS}(n \times cellsize_l)$
Let $v_{\text{scale}} = \text{ARCSEC\_TO\_RADIANS}(m \times cellsize_m)$
Compute $u_{\text{tap}}, v_{\text{tap}}, b_i, f_i$ from the thread index
Let s be the computed starting indexes (computed prior to launch)
Let $u_{\text{prev}}, v_{\text{prev}}$ be the previous u,v grid coordinates initially 0
Let $vis_x = 0 + 0i$ be the visibility accumulators for $x = 1, 2$ or 4 correlations
for $q \in [0...N_{\text{channels}})$ **do**
  Let cube_slice be the index of the grid frequency $spw[r] \times N_{\text{chan}} + q$ is to be accumulated to
  for $r \in [s[b_i]...s[b_i + 1])$ **do**
    if $field[r]$ not being gridded **or** $rowFlagged[r]$ **or** $channelFlagged[r, q]$ **then**
      continue
    end if
    Let $u, v, w = \frac{u[r] \times u_{\text{scale}}}{\text{wavelength}[q]}, \frac{v[r] \times v_{\text{scale}}}{\text{wavelength}[q]}, \frac{w[r]}{\text{wavelength}[q]}$
    First apply facet phase steer to original scaled uvw coordinates and orthogonal lmn coordinates:
    Let $p_f = \exp(2\pi i/\lambda[u(l_i - l_0) + v(m_i - m_0) + w(n_i - n_0)])$
    if doing polyhedron faceting **then**
      Tilt the facet by rotating the uv plane:
      Let uvw' be a set of rotated uvw coordinates, applying $R(\alpha_i, \delta_i)R^T(\alpha_0, \delta_0)$
    end if
    Let $u'_{\text{int}}, v'_{\text{int}}$ be the rounded (to nearest integer) u', v'
    Let $u'_{\text{frac}}, v'_{\text{frac}} = -u' + u'_{\text{int}}, -v' + v'_{\text{int}}$
    for $x \in [0...N_{\text{cross correlations}})$ **do**
      Let $vis = vis[r, q, x]$
      Instead of storing convolution kernels for negative w grid the complex conjugates of the baselines:

      if $w' < 0$ **then**
        Let $vis =$conjugate(vis)
        Let $u', v', w' \times = -1$
      end if
      Assuming linear spacing between sampled w layers:
      Let $w_{\text{plane}} = \text{round}(w'/w_{max} \times (N_{\text{planes}} - 1))$
      if $u_{\text{prev}} \neq u_{\text{int}}$ **or** $v_{\text{prev}} \neq v_{\text{int}}$ **then**
        if $u'_{\text{int}}, v'_{\text{int}} \pm c_{\text{half support}}$ within grid boundaries **then**
          Let $g[u'_{\text{prev}} + u_{\text{tap}} + \frac{n}{2}, v'_{\text{prev}} + v_{\text{tap}} + \frac{m}{2}, \text{cube\_slice}] += vis_x$
        end if
        Let $u_{\text{prev}} = u_{\text{int}}, v_{\text{prev}} = v_{\text{int}}, vis_x = 0 + 0i$
      end if
      Let $c_{\text{weight}} = c[(u_{\text{tap}} + c_{\text{half support}} + u'_{\text{frac}} + 1)) \times c_{\text{oversample}}, (v_{\text{tap}} + c_{\text{half support}} + v'_{\text{frac}} + 1)) \times c_{\text{oversample}}, w_{\text{plane}}]$
      Let $vis_x += vis \times p_f \times c_{\text{weight}}$
    end for
  end for
  if $u'_{\text{int}}, v'_{\text{int}} \pm c_{\text{half support}}$ within grid boundaries **then**
    Let $g[u'_{\text{prev}} + u_{\text{tap}} + \frac{n}{2}, v'_{\text{prev}} + v_{\text{tap}} + \frac{m}{2}, \text{cube\_slice}] += vis_x$
  end if
  Let $u_{\text{prev}} = u_{\text{int}}, v_{\text{prev}} = v_{\text{int}}, (\forall x \in [0...N_{\text{cross correlations}}))vis_x = 0 + 0i$
end for

large number of visibilities is prone to rounding error. Nicholas Higham [27] shows that an upper bound for summation error is given by the following for any well-behaved summation method:

$$|E_n| \leq (n-1)u\sum_{i=1}^{n}|x_i| + O(u^2), u := \frac{\beta}{2}\beta^{-\rho} \tag{5.3}$$

Here u is the machine epsilon and assumed to have $\beta = 2$ and $\rho = 24$ or $\rho = 53$ for IEEE 754 single and double precision, respectively (David Goldberg [24] gives a detailed discussion), and hence the u^2 term can be ignored.

The error scales with the number of terms as well as the magnitude of the terms. This is concerning because recent trends in interferometers have seen a significant growth in the number of baselines and channels, and hence the number of visibilities being accumulated. Large arrays also require convolution kernels with large support, which will further worsen the effects of total rounding error across the grid. We have decided to provide a set of single and double precision gridding routines in order to test how this effects the accuracy of the synthesized images. Since the FFT preserves the total energy between the spacial and frequency domains (Parseval's theorem) we expect the rounding error to be present in the brightness of sources in the reconstructed images. The difference between single and double precision synthesis for extended observations will be investigated in the results section 6.5.

The choice between single and double precision synthesis has a significant performance impact on GPUs where the number of double precision units is significantly less than single precision units (1 to 3 in the case of Kepler generation GPUs). The performance impact will be determined experimentally in the results section 6.5.

Our implementation therefore supports single precision and double precision imaging. To clarify what is meant by "double precision imaging" we stress that in double precision imaging all the input, computation and output products are kept in double precision, up to the point that the images are written out to FITS files. The uv data (visibilities, weights, uvw coordinates, etc.), complex-valued w-filters and importantly the complex-valued facet grids are, therefore, all stored in double precision. The input data is cast to double precision as part of the preprocessing routines and once gridding starts all computation (including facet phase-rotations and filter convolution (complex multiply-add)) are done in double precision [1].

5.8 Faceting options

Our imager supports both targeted and continuous faceting. In the instance of targeted faceting the user specifies a set of coordinates for the facet centres, along with the size of the facet images to be created. In continuous faceting mode the user specifies the size of the facets along with the number of facets in both l and m. The current implementation joins the facets at the facet edges, with no overlap. It is therefore necessary to pad each of the facet images to reduce the remaining aliasing energy that is normally encountered right at the edge of the image (as discussed not all the energy is stopped due to filter sidelobes).

The `CRVALn` keywords for the l and m coordinates in each of the facet image cubes are the coordiantes

[1] It may be possible to keep some of the data and operations in single precision to save on computation time on the GPU, while only keeping the atomic additions onto the grid in double precision but this was not explored in detail due to time constraints. We note that double precision attomics (at the time of writing) involve a while loop around an `atomicCAS` operation. Profiling with the Nvidia Visual Profiler highlighed significant branch divergence at this point.

of the original phase tracking centre l_0, m_0, even if those coordinates fall outside the area of the facet images. This is necessary to ensure that the facets all share the same divergence in coordinates that is introduced by the orthogonal ("SIN") projection.

We have included a prototype graphical interface to demonstrate facet imaging, shown here in Figure 5.8. The prototype tool is implemented in Python and uses the GTK framework. It calls directly on the command line interface of our imager.

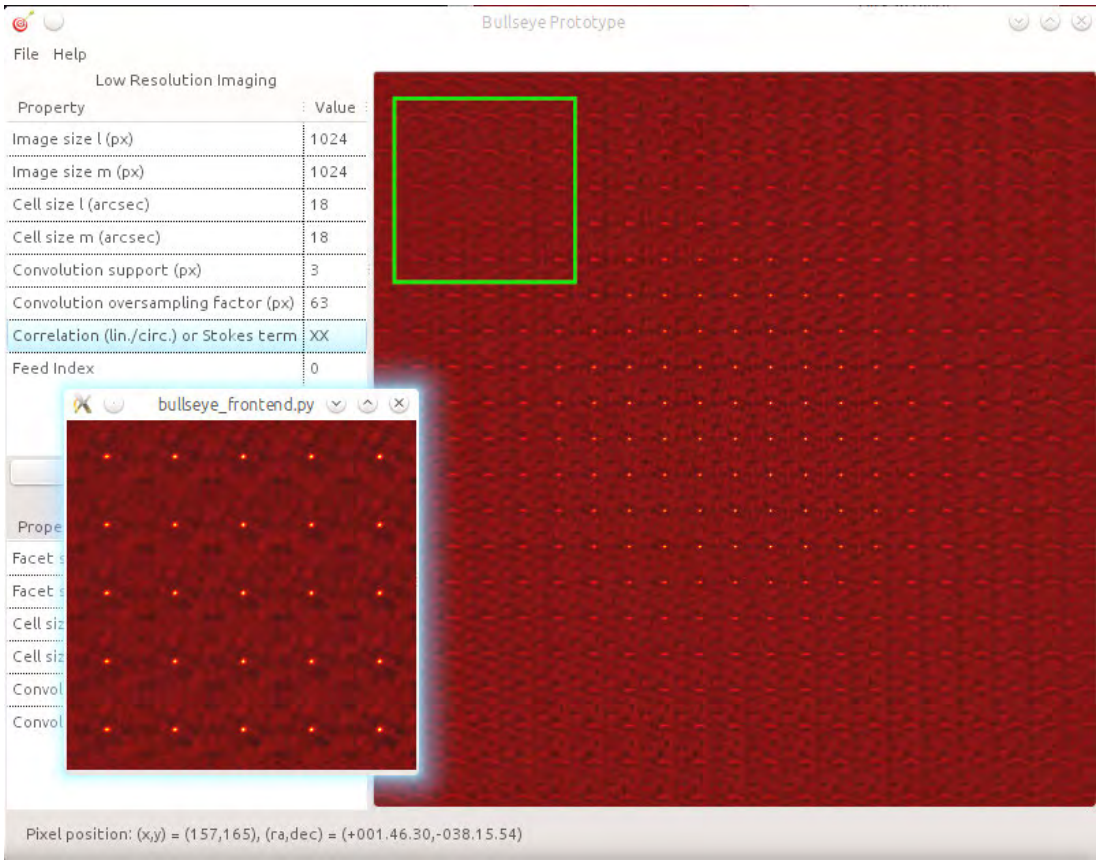


Figure 5.8: The result of faceting a narrow piece of sky that is far away from the phase tracking centre of the telescope. Here a simulated gridded sky pattern shows the widefield distortions of sources far away from the field centre. In the facet image of this area all the sources are corrected.

When constructing continuous images the user has the option to reproject and combine the individual facet images using the Montage [30] suite of tasks after the facets have been synthesized.

5.9 Image normalization

Each of the images are normalized by a counter that takes weights, flags and visibilities that are not gridded into account:

$$N = \sum_{k=0}^{M-1} \sum_{u_{\text{tap}}=0}^{\text{full sup}-1} \sum_{v_{\text{tap}}=0}^{\text{full sup}-1} \Re c[(u_{\text{tap}} + u[k]_{\text{frac}} + 1) \times c_{\text{oversamp}}, (v_{\text{tap}} + v[k]_{\text{frac}} + 1) \times c_{\text{oversamp}}] W[k] \neg F[k] \tag{5.4}$$

Due to the linearity of the Fourier transform this normalization can be applied per grid point or synthesized image pixel, and we choose to apply it as part of the image finalization step.

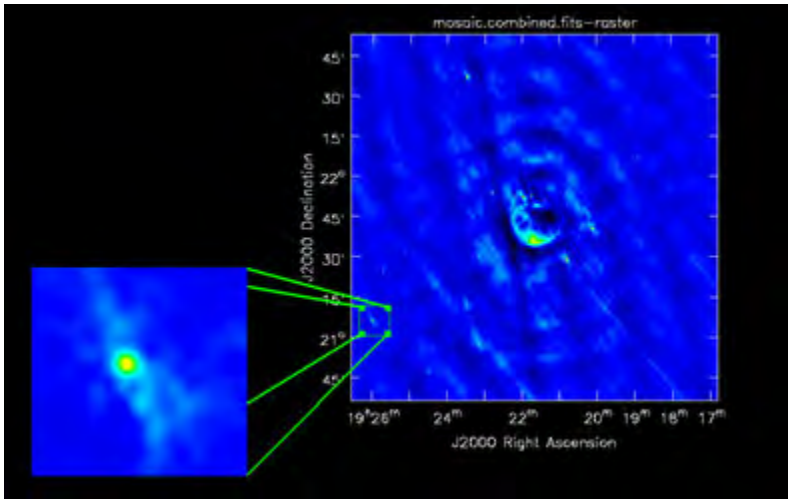
Unlike the CPU implementation on the GPU it is necessary to keep an extra register per thread and reduce to a single normalization value after gridding is completed in order to limit atomic accesses.

It is also possible to normalize by the centre value of the point spread function, but this requires that the PSF always be synthesized. When a full deconvolution pipeline is added to the imager this step can be changed.

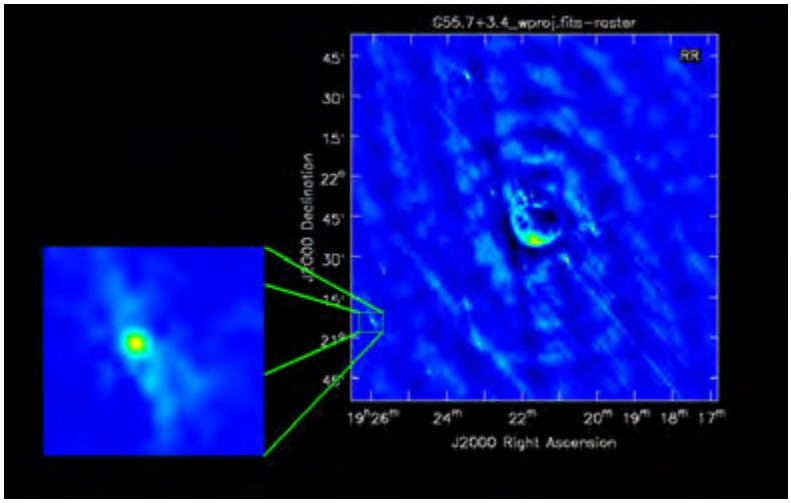
5.10 Validation and testing

We used both simulated and real data to to test our imager. Although our real data is only generated by the JVLA, we chose to support the Measurement Set format because of its widespread use in radio astronomy and the generalizations it presents to uv data storage. As for testing the widefield imaging capabilities of the imager, we followed the calibration and flagging guidelines for the supernova remnant G55.7+3.4 observed by the JVLA in L-band[2]. The observed field contains sources that had previously been identified as being distorted by the wide-field effect and these sources are also bright enough to be clearly visible in the dirty images, making this dataset ideal for use in testing our imager. The results are shown in Figure 5.9

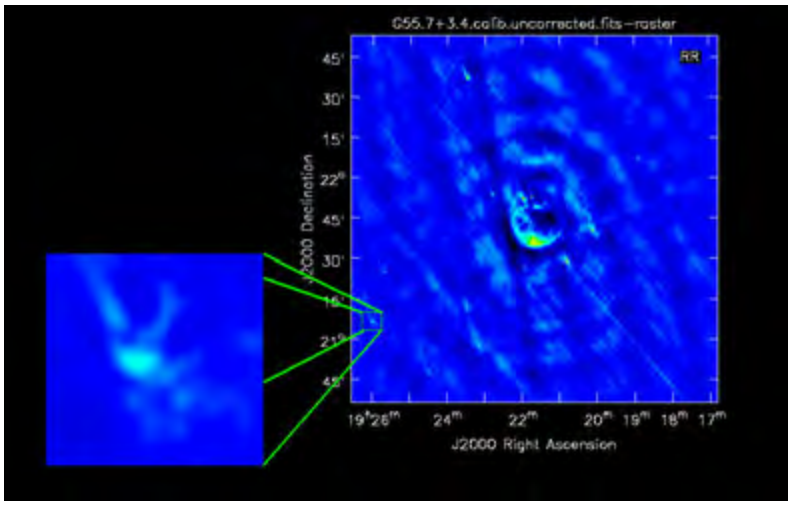
[2]The guide we used is available at `https://casaguides.nrao.edu/index.php/EVLA_Wide-Band_Wide-Field_Imaging:_G55.7_3.4-CASA4.4`



(a) Mosaiced faceted image (2x2 facets, 1.2% padding each). No additional w-projection.



(b) W-projected image. No faceting.



(c) Uncorrected image (normal narrow field imaging).

Figure 5.9: In this field there is a clear wide-field distorition near right ascension $19h26m$, declination $21°7'$. Both the faceting and w-projection implementations corrects the distortion as expected. The w-projected image here uses no faceting and similarly the faceting was done by enabling only a first order approximation for w and disabling w-projection.

Chapter 6

Performance analysis

In this chapter an analysis is given on the scalability of the various resampling implementations and policies. To draw a comparison between the various combinations of approaches we investigate the performance of faceting, w-projection and w-faceting, both in single and double precision. For scaling performance we used the same datasets and imaging parameters. After this we investigated the impact of the choice of precision, both with increasing support and observation time.

6.1 Testing aparatus

The following machines were used in generating the results in this chapter:

- System A:
  One node of the UCT ICTS High Performance HEX cluster, which has 4 16-core AMD Opteron 6376 CPUs. The CPUs have a maximum memory bandwidth of 51.2 GB/s and average power consumption of 115W [1].

- System B:
  Most of the GPU results were generated on a machine with 4x 8-core Intel Xeon E5-2690 with 1 NVIDIA Tesla K40m co-processor (compute capability 3.5) at Rhodes University. The K40m has a maximum memory bandwidth of 288 GB/s and maximum power usage of 215W [2]. The NVIDIA CUDA toolkit version 6.5 was used to compile our imager.

- System C:
  A development machine with an Intel Core i7-4770, a quad-core processor with a maximum memory bandwidth of 25.6 GB/s [3]. The machine has a NVIDIA GTX 770 card installed with a maximum memory bandwidth of 224.3 GB/s and power usage of 230W [4]. The Nvidia toolkit version 6.5 was used to compile our imager.

[1] According to AMD, `http://www.amd.com/en-us/products/server/opteron/6000/6300#`
[2] According to the NVIDIA Tesla K40 GPU Active Accelerator Board Specification
[3] `http://ark.intel.com/products/75122/Intel-Core-i7-4770-Processor-8M-Cache-up-to-3_90-GHz`
[4] According to NVIDIA `http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-770/specifications`

6.2 Metrics

The primary performance metric used to measure gridder performance was Giga Grid Point Additions per second (abbriviated "G"). This metric includes the size of the convolution filter support, number of correlations and number of channels being gridded per dataset record:

$$\text{Giga grid point additions per second}(G) := \frac{\tau_{\text{observed}}}{\tau_{\text{integration}}} N_{\text{baselines}} N_{\text{channel}} N_{\text{corr}} C_{\text{sup}}^2 10^{-9} s^{-1} \tag{6.1}$$

In our discussion we include the time needed to copy visibilities onto the GPU and images back to the host in this calculation, but exclude Fourier transform costs; these costs become negligible with increased observation time and data-rates.

Another important cost to consider is the power efficiency of the gridding operation, in order to consider the running costs of a CPU vs. GPU-based solution. To this end we define the following ratio:

$$\text{Power Efficiency} := GW^{-1} \tag{6.2}$$

We will use the advertised power consumption (in Watts) in our discussion at the end of this chapter.

6.3 Dataset simulation

Ten datasets were simulated for the experiments presented in this chapter. The datasets in Measurement Set format were generated using the Makems tool[5]. Dataset 1 is generated using a benchmarking tool included with our imager. Both the JVLA and MeerKAT observations have many more available channels than those used: the JVLA has a minimum of 16,384 spectral channels [44], for instance, but we decided to produce continuum images with far fewer channels to cut down on the run-time of these experiments. Note that we already established the validity of the images produced by comparing to other tools such as the CASA imager, so we opt to use simulated datasets here to control the parameters such as integration time.

For each of the telescopes the approximate diffraction-limited beam radius is shown, as well as the minimum Nyquest cell size and corresponding number of pixels. The experiments used grids padded with enough pixels to account for the size of the convolution filter.

6.4 Scalability

This section presents the results from experiments with various imager configurations. The discussion of these results follow in section 6.6.

[5] Available at `https://github.com/ska-sa/makems`

	(1) JVLA (small)	(2) JVLA (large)	(3) MeerKAT (small to large)
τ_{observed}	4hrs	4hrs	5mins, 10mins, 15mins, 30mins, 45mins, 60mins, 2hrs & 3hrs
$\tau_{\text{integration}}$	8.49056s	3.00s	3.00s
ν_{min}	1.2 GHz (L-band)	1.2 GHz	0.9 GHz (L-band)
$\Delta\nu$	1.0 MHz	1.0 MHz	1.0 MHz
D	25m	25m	13.5m
B	1km (D-conf)	1km	8km
N_{ant}	27	27	64
N_{bl}	378	378	2080
N_{spw}	1	1	1
N_{chan}	128	128	256
N_{corr}	1,2,4	4	4
Type	benchmark	ms	ms
Dataset size (vis only)	313.03 MiB	6.92 GiB	1.59 GiB, 3.17 GiB, 4.76 GiB, 9.52 GiB, 14.28 GiB, 19.04 GiB, 38 GiB & 57.13 GiB
Diffraction limited Image Radius	34.35368 arcmins	34.35368 arcmins	84.8239 arcmins
Critical sampling cell size	0.42942 arcmins	0.42942 arcmins	0.07157 arcmins
Minimum number of pixels	80^2	80^2	1186^2

Table 6.1: Specifications of the datasets used in benchmarking and results generation.

6.4.1 Faceting only

This set of performance benchmarks focuses on creating narrow-field facets using the first order approximation to $w(n-1)$. As such full w-projection is disabled.

For the first experiment 144 facets are created on both the CPU and GPU. For this experiment w-projection is disabled and the convolution filter (real) is limited to a full support of 7 pixels. Only a single correlation of dataset 2 is gridded. Both single and double precision performance are indicated in Figure 6.1, along with their respective power efficiencies in Figure 6.2.

Figure 6.3 shows performance on the K40m GPU when the number of facets are varied. In each case the total field of view is kept constant. W-projection remains disabled.

The next experiment used dataset 1 and was run using system C. It shows the dependance of gridding performance on the facet transforms. Again no w-projection was enabled, full filter support of 7 pixels was used and only a single facet is created. The results are shown in Figure 6.4.

6.4.2 W-projection scaling

Figure 6.5 shows how w-projection scales with full filter support size on the K40m. The performance for both real and complex (separable) filters is indicated. Here dataset 2 was used and faceting transforms were disabled. Only a single correlation was gridded.

The CPU implementation uses vectorization when w-projection is enabled. Figure 6.6 shows the speedup between AVX-vectorized gridding compared to a non-vectorized implementation when gridding 1, 2 and

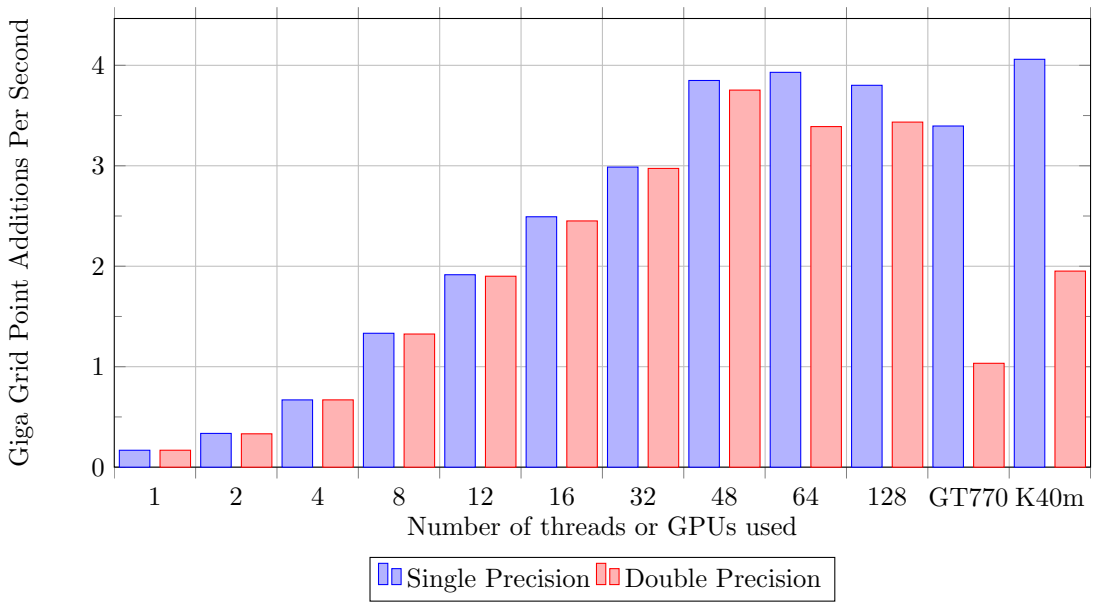


Figure 6.1: GPU vs CPU Facet Imaging Performance

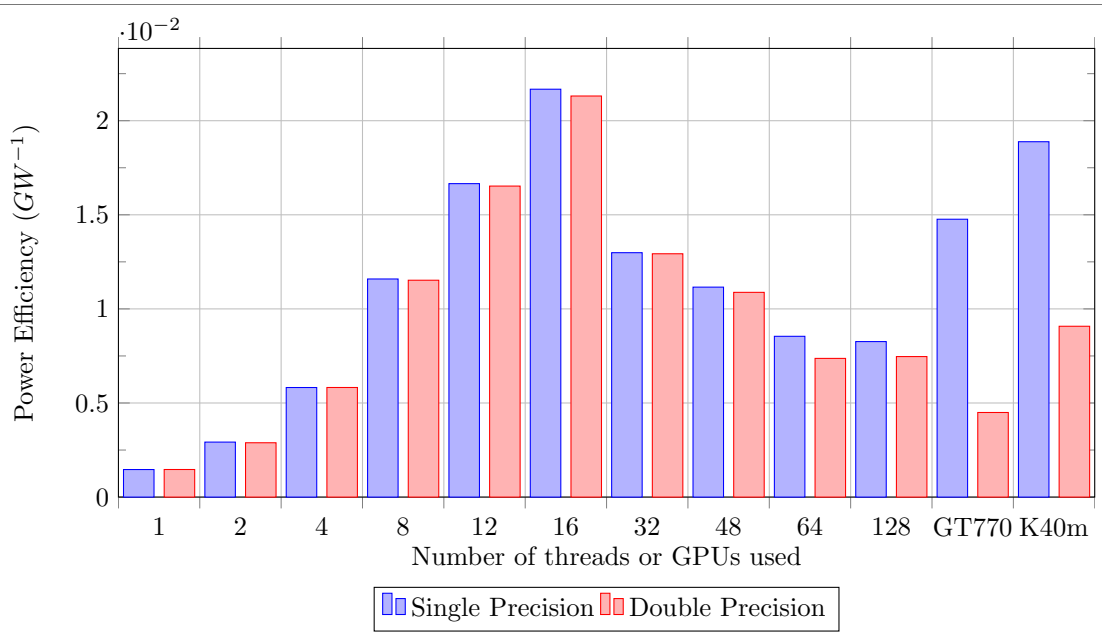


Figure 6.2: CPU vs GPU Facet Imaging Power Efficiency

4 correlations. System C was used to generate these results.

Figure 6.7 shows the performance of w-projection (2D filter lookup table) with AVX enabled, as well as the performance of real-valued (separable 1D kernel) filtering on the CPU. Here we gridded dataset 1 on system C.

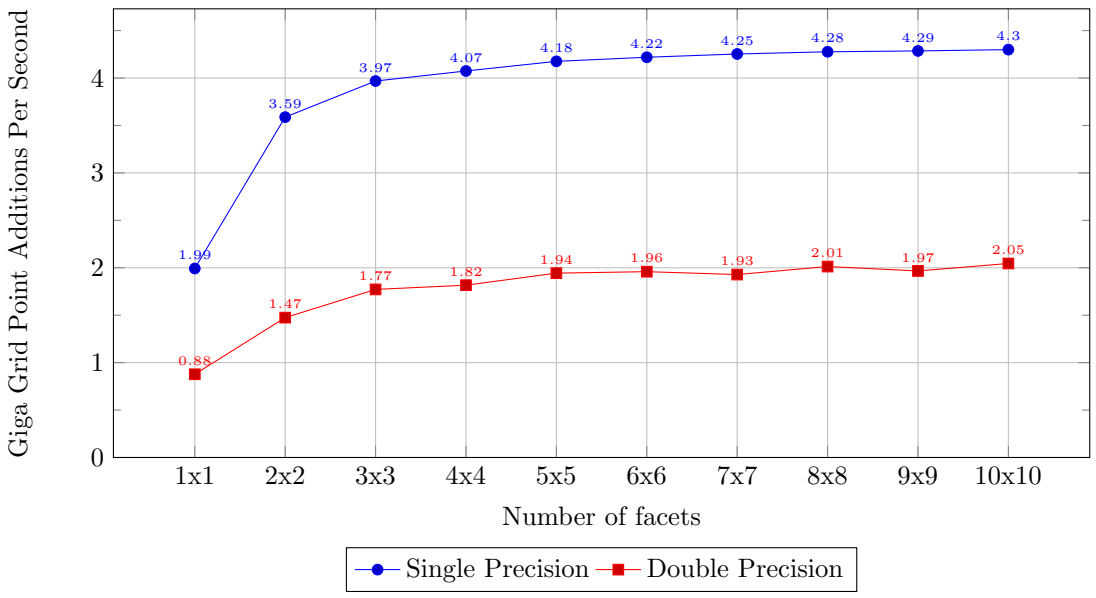


Figure 6.3: Scalability of GPU-based faceting (first order phase approximation)

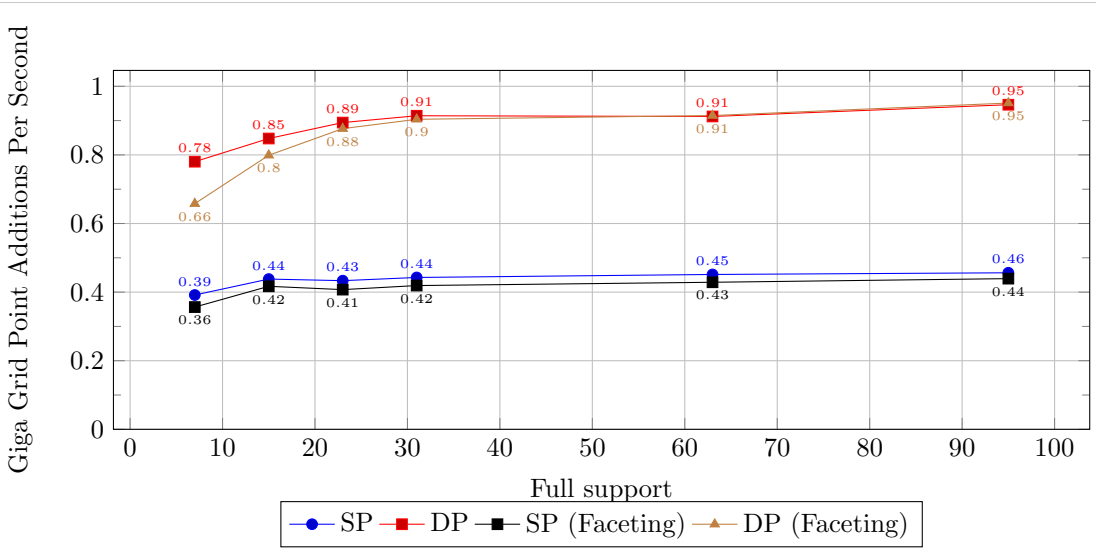


Figure 6.4: Effect of facet transforms (CPU imaging)

6.4.3 W-faceting

The next experiment shows scaling performance relative to filter support size for the K40m when faceting transforms are added to w-projection. Dataset 2 was used in generating these results. Figure 6.8 shows the results of gridding in single precision.

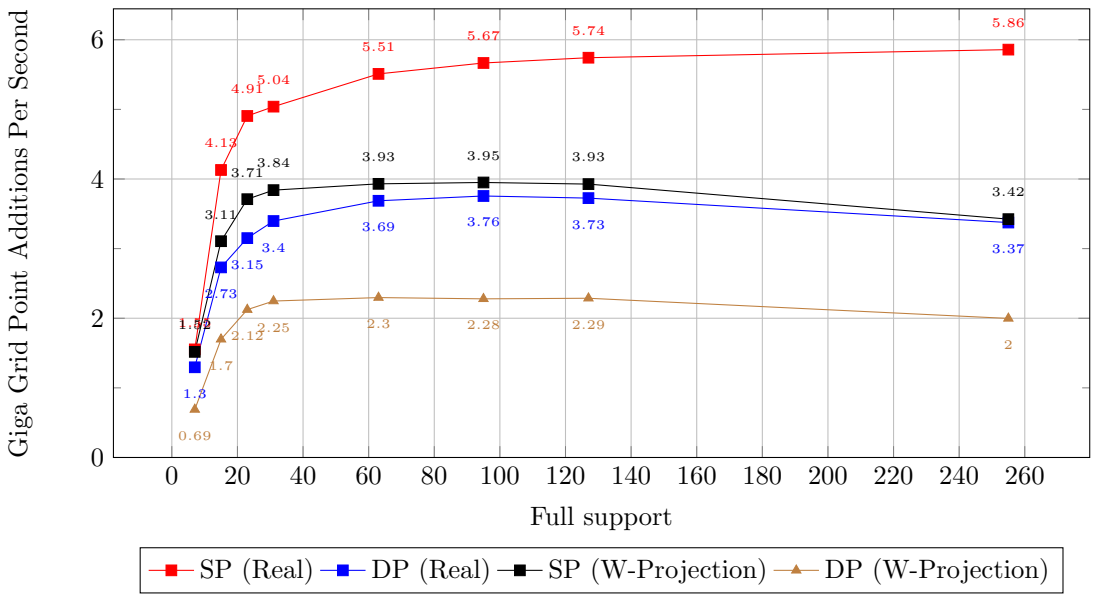


Figure 6.5: GPU-based scaling with filter support size (no faceting)

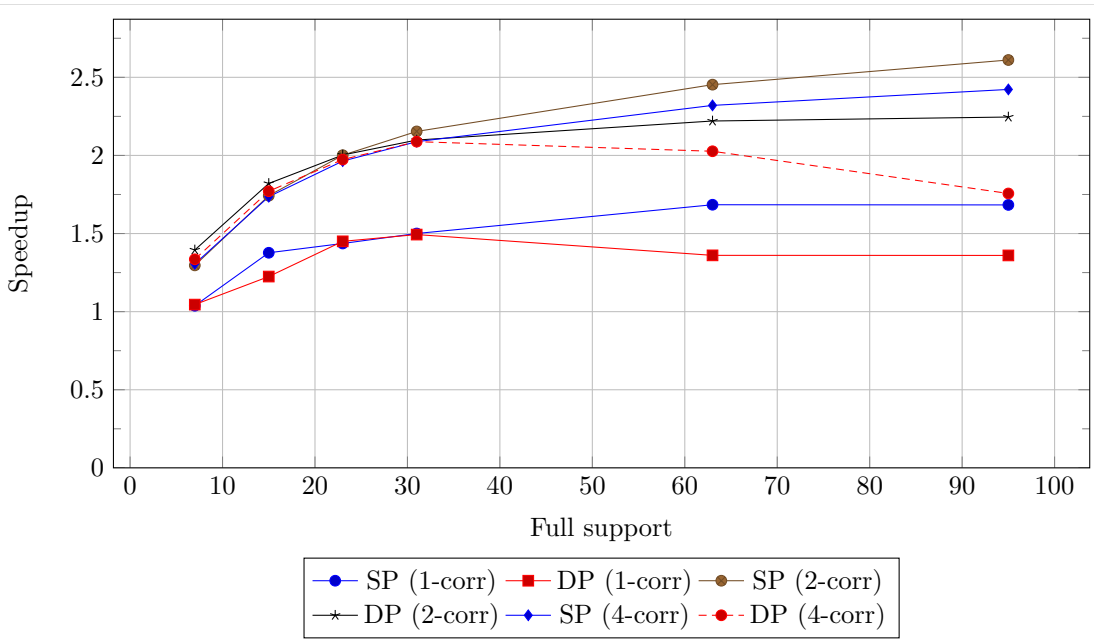


Figure 6.6: Speedups obtained due to vectorizing (CPU-based w-projection)

6.5 Precision

Although single precision gridding is desirable in terms of GPU performance (refer to figures 6.1 & 6.2), the floating point error introduced to the images cannot be ignored and has (to our knowledge) not been documented in previous literarure on the subject. We propose the following two experiments:

- Determine the relative error introduced in measured source brightness by longer observations on a large telescope such as MeerKAT.

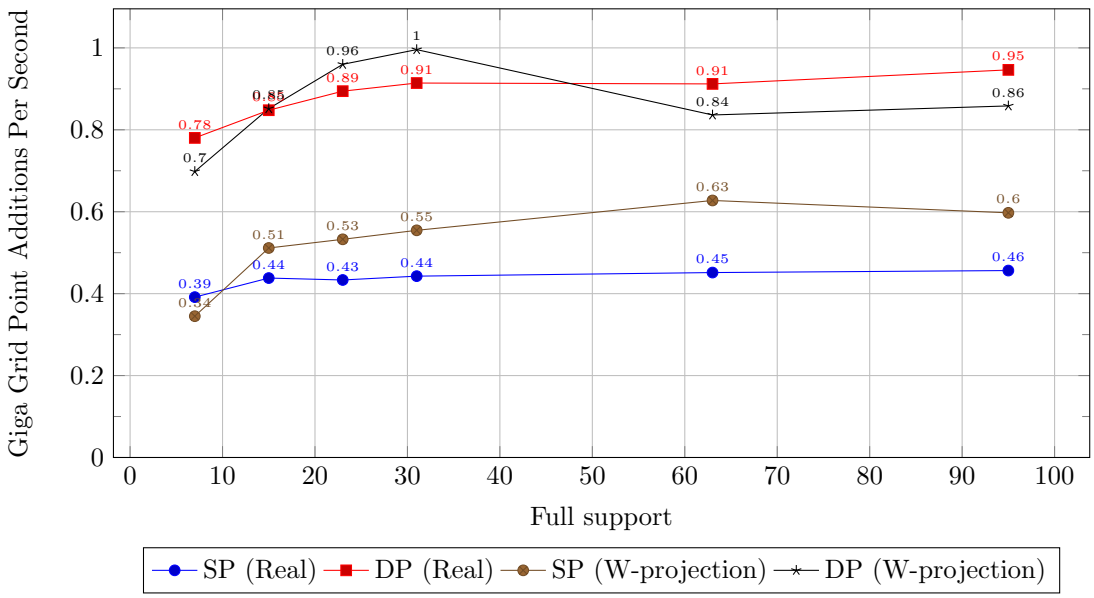


Figure 6.7: CPU performance using real and w-projection filtering.

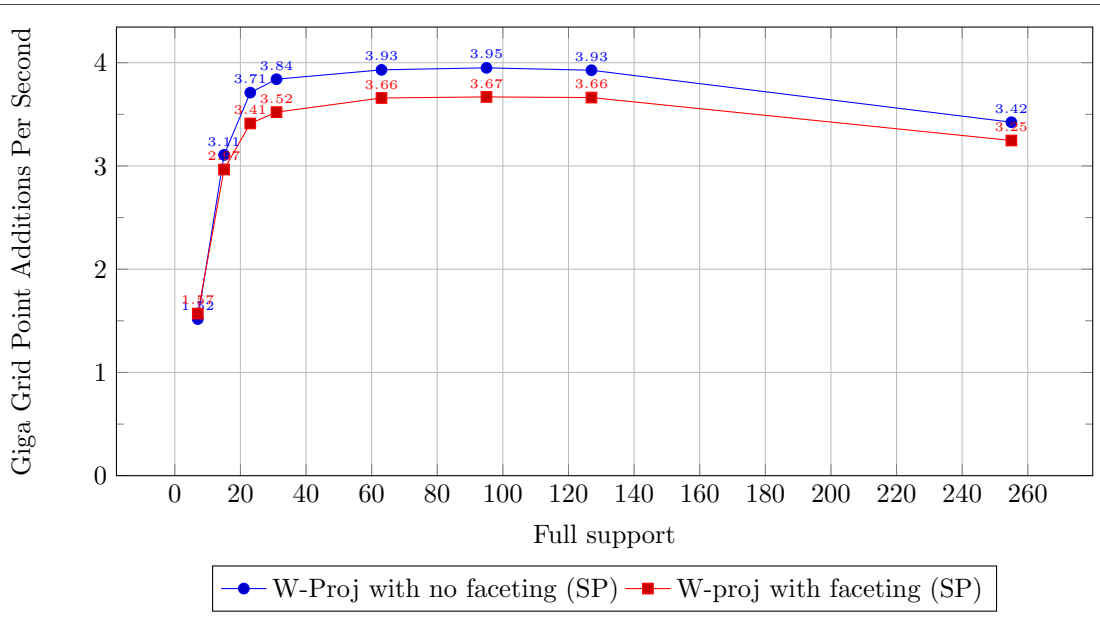


Figure 6.8: GPU scaling performance when employing both faceting and w-projection

- Determine how the relative error grows with filter support size.

For the first experiment we generated datasets 3 through 10, increasing the observation time, and therefore total number of visibilities, of each dataset. Prior to imaging we set all the visibilities equal to $1+0i$, which must produce a 1 Jy source in the centre of the image. The relative error is then computed as follows:

$$\text{Relative Brightness Error (RBE)} := ||\frac{\text{Measured brightness}}{\text{True brightness}}|| \tag{6.3}$$

The relative brightness of the produced source is not the only important metric to consider. We also include an indication of the noise level in the image. Since the simulated data does not contain the expected instrumentation and environmental noise, we expect to see a combination of noise introduced by floating point error and gridding interpolation error. By comparing the relative noise level between single and double precision gridding at various integration times and filter support sizes the latter interpolation error, as well as the differences in the amplification effects of the sidelobes of the instrument's PSF can be eliminated as additional noise and noise-amplification sources. To that end we propose the following relative measure of summation error:

$$\text{Relative Summation Noise (RSN)} := \frac{\text{SNR}(I_{\text{single}})}{\text{SNR}(I_{\text{double}})}, \text{SNR}(I) := ||\frac{\text{Measured brightness}}{\text{mean}(I)}|| \tag{6.4}$$

Ideally this measure should be very close to 1.0 if there is no inaccuracy introduced by single precision gridding. Since the image is mostly devoid of emission, the signal to noise defined by the mean over the entire image should give a good estimation of the background noise levels of the images.

6.5.1 Effect of longer observation time

For this experiment we used measurement sets of 5, 10, 15, 30, 45, 60, 120 and 180 minutes in length. We fixed the filter support size at 7 pixels with an oversampling factor of 63 pixels and used the truncated (box-windowed) sinc function. A continuum image is made by averaging all 256 channels per observation. The RBE and RSN for these continuum images are plotted in Figure 6.9.

6.5.2 Effect of increasing convolution filter support

For this experiment we kept the observation time constant at 5 mins and varied the filter support size: 7, 15, 23, 31, 63 and 95px. Again all 256 channels are averaged into a single continuum image. Both the RBE and RSN are plotted in Figure 6.10.

6.6 Discussion

Figure 6.1 shows that, at least, for small facets GPU performance is slightly better than that of a multicore CPU node when gridding in single precision. It is also clear that the parallel CPU imager does not scale linearly with number of cores, most likely due to an imbalance in the workload (the number of facets being gridded cannot be divided evenly between CPU cores). This leads to an overall drop in power efficiency when using multiple cores (Figure 6.2). In terms of power efficiency the K40 outperforms the CPU node when comparing the peak performance of the CPU cluster at 64 cores, but is slightly less power-efficient than employing a single multi-core CPU.

When gridding with double precision the GPU is outperformed by a single multi-core CPU and is therefore significantly less power efficient when compared to a single 16-core Opteron CPU, of course assuming that enough facets are created to fully utilize the CPU.

Figure 6.3 shows that once the GPU is saturated with enough work, gridding performance increases up to around 4.3 G when faceting and 3.42 G when doing w-projection (Figure 6.5). In both single and

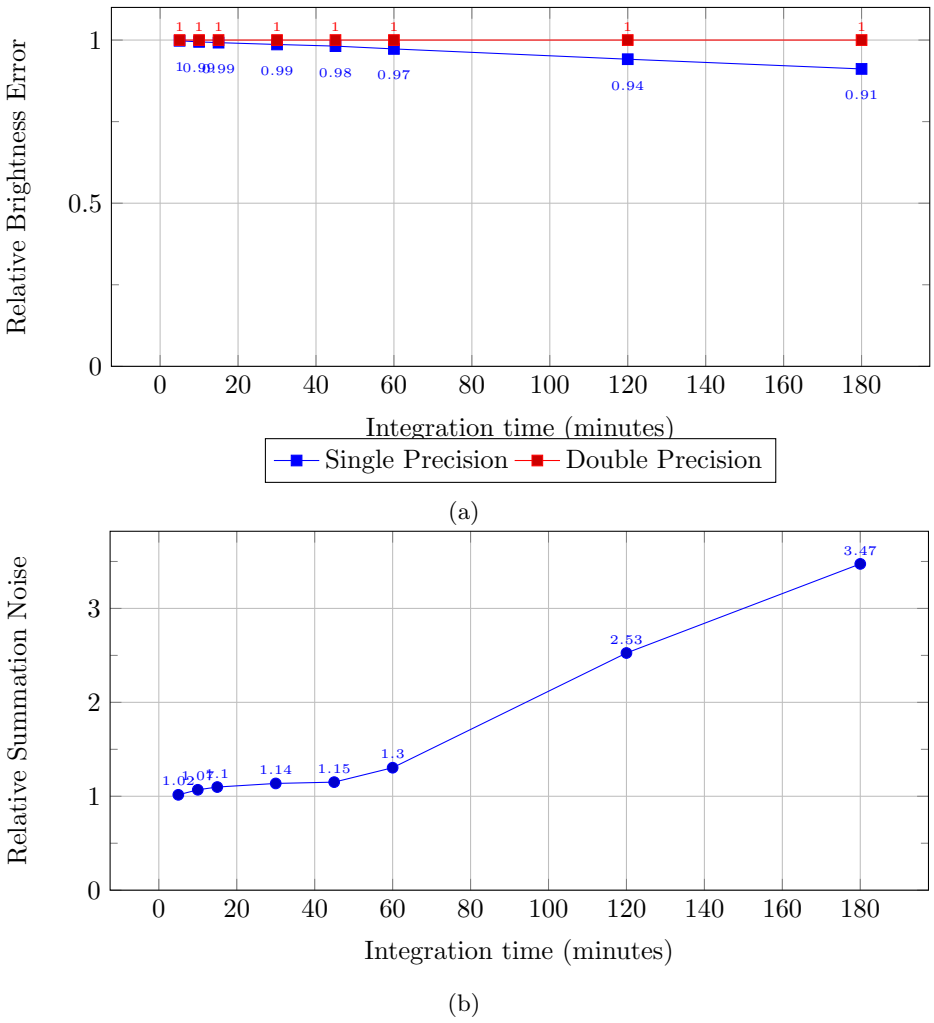


Figure 6.9: (a) The Relative Brightness Error is plotted for single and double precision gridding for various observation times. (b) The associated Relative Summation Noise is shown for the same observation times.

double precision w-projection filtering is significantly slower than using real filters of the same size. By combining w-projection and faceting to create smaller coplanar facets the drop in gridding performance observed for large filters can be avoided. In this context smaller w-facets scales slightly better than regular w-projection with large filters (Figure 6.8).

When the CPU-implementation of the imager is run on newer Intel (Haswell Generation) hardware, double precision outperfroms single precision (Figure 6.7). Considering that the faceting transforms have very little effect on the run-time for larger convolution kernels (Figure 6.4) in the CPU imager this result is yet another argument against using GPUs when gridding with double precision. Assuming near-linear scaling for the first 4 cores (a well-balanced faceting workload as seen in Figure 6.1 for the Opteron hardware) this indicates that a single newer generation processor will outperform the K40 when synthesizing multiple larger w-facets using AVX-vectorized convolution operations. The power consumption of that single processor is also much less than the power consumption of the K40 (84W compared to 230W).

It should be pointed out that the performance difference between real-valued and w-projection filters for

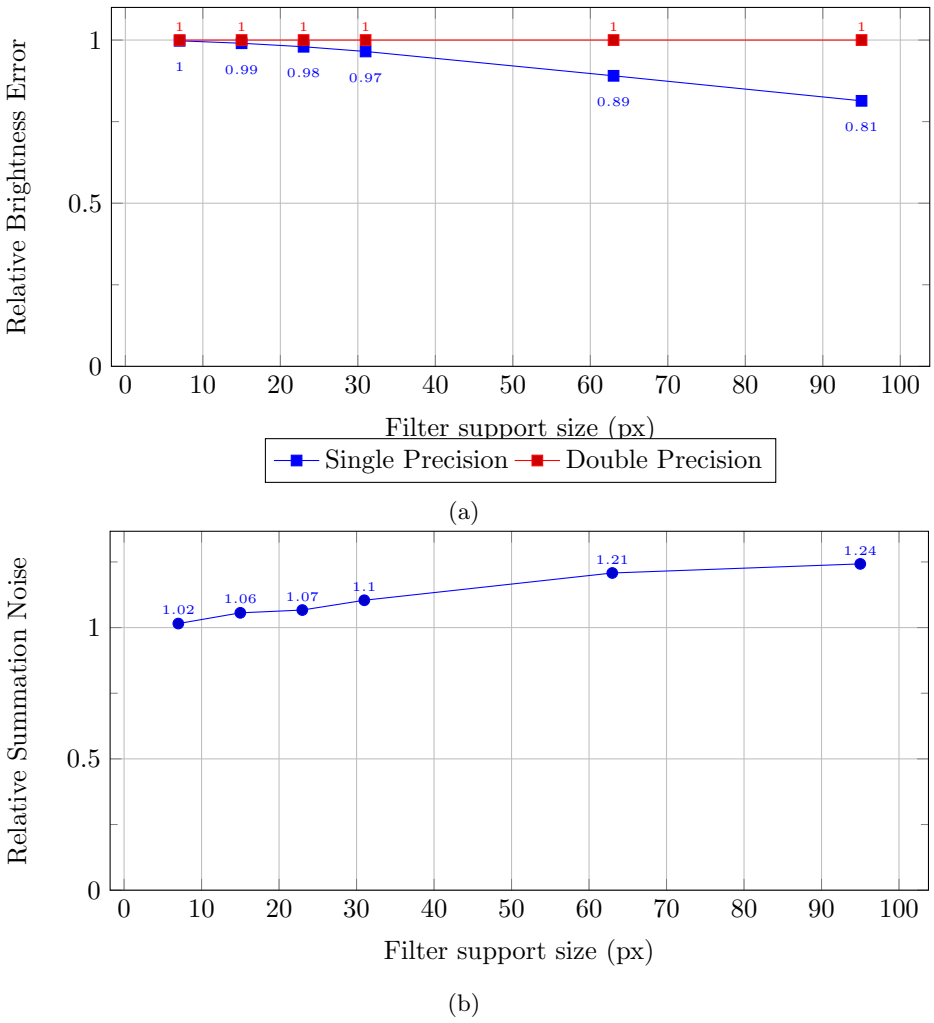


Figure 6.10: (a) The Relative Brightness Error is plotted for single and double precision gridding, and the Relative Summation Noise is plotted in (b) for various filter support sizes.

both single and double precision on the CPU can be explained by the fact that the w-projection logic is vectorized (single correlation gridding is 1.35x faster than its counterpart as shown in Figure 6.6), whereas in the real-valued logic is not. We do not expect that the anti-aliasing filter would ever need such large support sizes and hence did not provide a vectorized implementation for it. Figure 6.6 also shows that when gridding more correlations at a time (necessary when creating multiple Stokes images or doing Jones corrections per facet) the speedup due to vectorization increases to more than 2.5x when gridding with larger filters. We suspect that the drop in double precision quad correlation gridding performance may be memory related due to the latencies involved from the increased number of memory accesses to filter and grid memory before vector computations are performed; a quad-correlation gridding operation would require at least 8 accesses to memory when the address is aligned to the memory banks per convolution filter tap, possibly stalling the CPU in the process. We reran the duel correlation gridding experiments with filters supports up to 255 pixels to see whether this became a problem in those cases as well and we noticed a similar drop in attained speedup.

The results from our precision experiments show that constructing continuum images for a MeerKAT-sized telescope using single precision will introduce significant floating point error. Even though we used

input data of the same magnitude (all visibilities were $1 + 0i$) there was a significant divergence in the brightness of the produced source, culminating in a brightness error of 9% for the 3 hour observation (Figure 6.9). This is likely a slight underestimate for realistic calibrated observations where the observed visibilities may differ more in magnitude. Even more concerning is the fact that the error is distributed across the image and grows even more rapidly than the RBE with increased observation time. Even though the measured brightness of the source is decreasing, the SNR of the single precision image increases with observation time, while the SNR of the double precision image stays close to constant across all observations. This indicates that the mean value in the single precision images is decreasing with observation time and is also decreasing much faster than the measured brightness. This supports our intuition that the low frequency components of the image (as sampled by the shortest baselines) are effected worse than the high frequency components (such as the point source in the centre of the image) sampled by the longer baselines. It also apears that the growth in the error of the low-frequency components increases non-linearly with observation time, whereas the error in the measured brightness increases linearly. Figure 6.10 shows that the brightness error grows more rapidly with an increase in filter size than the error seen with increasing observation time. In this figure the RSN grows slower compared to the RSN with increasing observation time - this may be because there are far fewer measurements taken in the central region of the uv plane with a short observation than with a longer observation.

For comparison we re-ran the same support-size experiment on the long 3 hour observation and saw devastating effects in the fidelity of the images produced using single precision gridding (while the images synthesized using double precision gridding were uneffected), see Figure 6.11. The error in the PSF sidelobe structure across the images support our statement that the low frequency terms are more susceptible to the noise introduced by this rounding error than the high frequency components.

6.7 Profiling and implementation comments

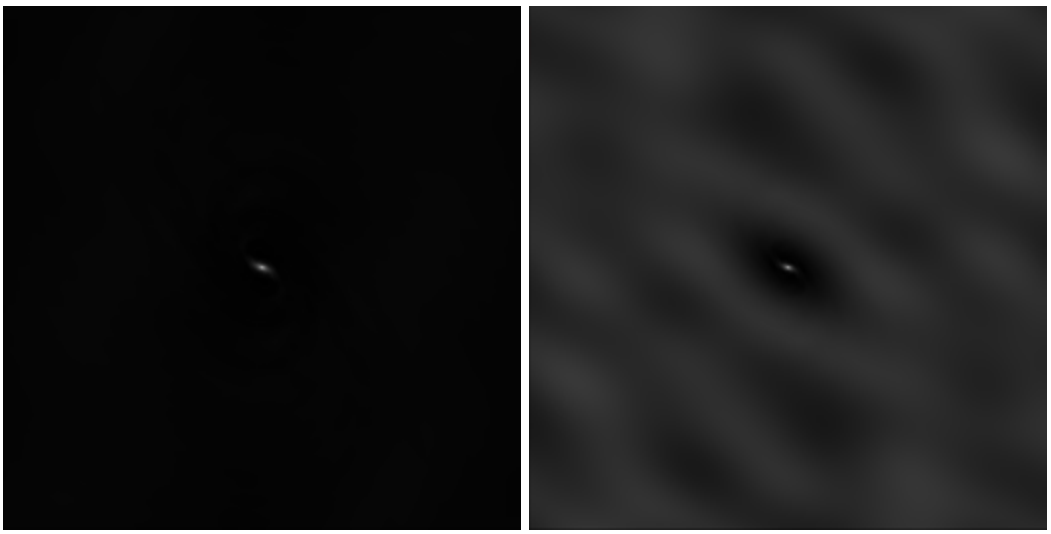
The NVIDIA Visual Profiler shows that the single precision gridding algorithm[6] consists of just over 40% integer arithmetic and only about 10% single precision floating point arithmetic. These figures exclude the instructions wasted on branch divergence. Our profiling also shows that the implementation is firmly bounded by memory latencies. Surprisingly this is not due to the global memory bandwidth (profiling shows that only about 10% of the peak bandwidth is used), instead kernel register preasure is responsible for lowering the obtained occupancy (even for single correlation gridding). The combination of the low single precision floating point instruction usage and memory latencies results in a 27% utilization of available compute (Function Unit Utilization, as reported by the NVIDIA profiler). This also explains why the results indicate that double precision performance is not a third of single precision performance [7].

We suspect that the increased register usage of our algorithm and the resulting low compute utilization is one of the primary causes of the large discrepency between the performance figures seen in John Romein's gridding benchmark[8] [46] and our implementation. Our implementation is somewhat more general, and therefore algorithmically more complex than the benchmark implementation. Bullseye is able to handle common imaging use-cases, such as forming continuum image cubes, by dividing the available observation

[6] Faceting a single correlation, with w-projection code disabled by means of templated traits and policies. These figures were obtained running on a GT960.

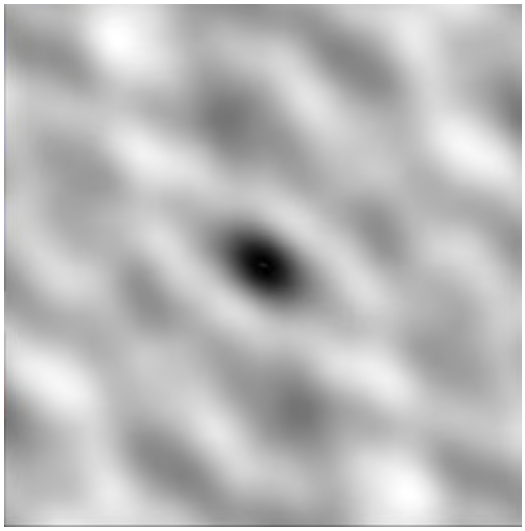
[7] The Kepler Architecture has 1 double precision arithmetic unit for every 3 single precision units

[8] Code available at `www.exaska.org/?q=Codes`



(a) 7px full support

(b) 15px full support



(c) 23px full support

Figure 6.11: This figure contains synthesized images when resampling 3 hour observations using filters with 7, 15 and 23 px full support sizes in single precision. We suspect that there may be overflows on some of the grid cells because the measured brightness on the 23 px filter support image is negative. We eliminated the normalization counter as the cause of this problem since it is always computed in double precision and the double precision images remain accurate at a measured brightness of 1 Jy with a constant SNR.

bandwidth (which may be spread out over multiple spectral windows) between grids for example, whereas this would require preprocessing the data and running Romein's implementation multiple times. We also do not precompute and store a set of uvw coordinates per channel because of the considerable memory requirements when dealing with larger databases, instead these are scaled by frequency on the fly. Some necessary memory accesses to visibility weights and flagging information, and the assocated branching that is common in imaging implementations are also missing from the benchmark.

Chapter 7

Conclusion and future work

In this work we investigated coplanar faceting as a solution to the problem of wide-field synthesis imaging using non-coplanar baselines. Our work combines the approaches of w-projection and traditional facet imaging in order to reduce the significant memory requirements of storing and looking up values in the large convolution filters associated with large fields of view. We compared the scalability of this coplanar faceting approach to w-projection with both single and double precision, drawing a comparison between an optimized multi-threaded CPU facet imager and a GPU imager. We also investigated the accuracy of gridding in single and double precision.

In line with previous work, we found that single precision imaging scales well on the GPU. Our work shows that the GPU is substantially (2.28x) more power-efficient than using a multi-core multi-processor processing solution. Muscat [37] points out that double precision gridding will be slower on GPUs, but does not investigate further. Our work sheds some light on GPU double precision scalability and we find that w-projection using double precision is only on average 71% slower than single precision w-projection for filter support sizes of 23px and upward, primarily because the performance of gridding is constrained by memory access latencies.

We showed that the combination of w-projection and faceting (which we refer to as "w-faceting") will scale slightly better than when using w-projection alone, since the efficiency of the w-projection algorithm drops with increasing filter size on GPUs. In light of the results generated using newer CPU hardware we recommend that w-faceting be performed on a multi-core (possibly multi-processor) CPU environment instead of a GPU when imaging using double precision; when gridding in double precision the CPU-cores attain w-projection gridding rates on average 0.91 G per core over filter support sizes of 23-95px, whereas the GPU averages 2.23 G.

Our investigations into the accuracy of single and double precision yielded concerning results, especially since current GPU implementations assume single precision gridding. We found that for arrays containing a larger number of antennae, such as MeerKAT, the error introduced in images due to choice of precision is considerable (9% when when using only small 7px filters). We also found that this additional source of noise effects the lower frequency components of the image more than higher frequency terms (such as point sources). This non-uniform structure in the noise is worrisome because it will adversely effect the dynamic range of images with extended emission structures. This merits further investigation to determine the suitability of current imaging implementations for core-dense arrays such as the SKA [5]. We note that the image oversampling factor (cell size above Nyquist rate) and gridding implementations

using time and frequency compression may decrease this error on the short baselines, but should be investigated in future work.

We suggest that a facet-based deconvolution strategy, including possibly an accelerated degridding ("prediction") step be added to our software package in future work on the topic. This is essential for any scientific use of the software package. Adding support for correcting for the direction- and time-dependent beam gain on bright sources through targeted faceting is also essential and may prove more feasible than current A-projection software packages. Lastly, we note that the current gridding implementation lacks support for time and frequency compression, which would increase the performance of both the CPU and GPU implementations.

Appendices

Appendix A

Signal processing refresher

Radio astronomy has its roots in signals processing. Before diving into the details of how these telescopes work we refer the reader to the *Scientist and Engineer's guide to Digital Signals Processing* by Steven Smith[1][53] for a detailed, yet simple overview of the field. A very brief overview of some of the core terminology is given here.

A signal is simply a description of how one parameter depends on another. A typical example of this is how voltage varies over time. Since the focus in this thesis lie solely in digitized signals, it is necessary to carefully define under what circumstances the underlying continuous analog signal can be reconstructed. The Shannon-Nyquist *sampling theorem* forms the cornerstone of Digital Signals Processing. It simply states that the rate at which samples are taken from a continuous signal has to be at least twice the highest frequency component of that signal. Such a signal is said to be critically sampled if it obeys this minimum requirement. If the requirement is not satisfied the sampled signal is aliased (higher frequencies appear as low frequency components). This can be illustrated through figure A.1. Aliasing is usually reduced by applying a filter that have low responses at frequencies outside a limited passband of frequencies. A filter can be something as simple as a windowed-sinc function for instance. Although a sharp cutoff frequency is ideal, in reality this is not achieved and responses usually decline more slowly, this is known as filter *roll-off*.

The next observation is that the propagation of electromagnetic waves and their interactions can be considered as a linear system. This means the signals exhibit three properties: homogeneity, additivity and shift-invariance (the third is a non-compulsory property of linearity). The following are true for such systems:

if $x[n] \rightarrow \text{System} \rightarrow y[n]$ **then** $kx[n] \rightarrow \text{System} \rightarrow ky[n]$ *(homogeneity)*

if $x_1[n] \rightarrow \text{System} \rightarrow y_1[n]$ **and** $x_2[n] \rightarrow \text{System} \rightarrow y_2[n]$
then $x_1[n] + x_2[n] \rightarrow \text{System} \rightarrow y_1[n] + y_2[n]$ *(additivity)*

if $x[n] \rightarrow \text{System} \rightarrow y[n]$ **then** $x[n+s] \rightarrow \text{System} \rightarrow y[n+s]$ *(shift-invariance)*

Much of the filtering and imaging techniques described in this thesis rely on the theory of Fourier

[1] Available freely at `http://www.dspguide.com/`

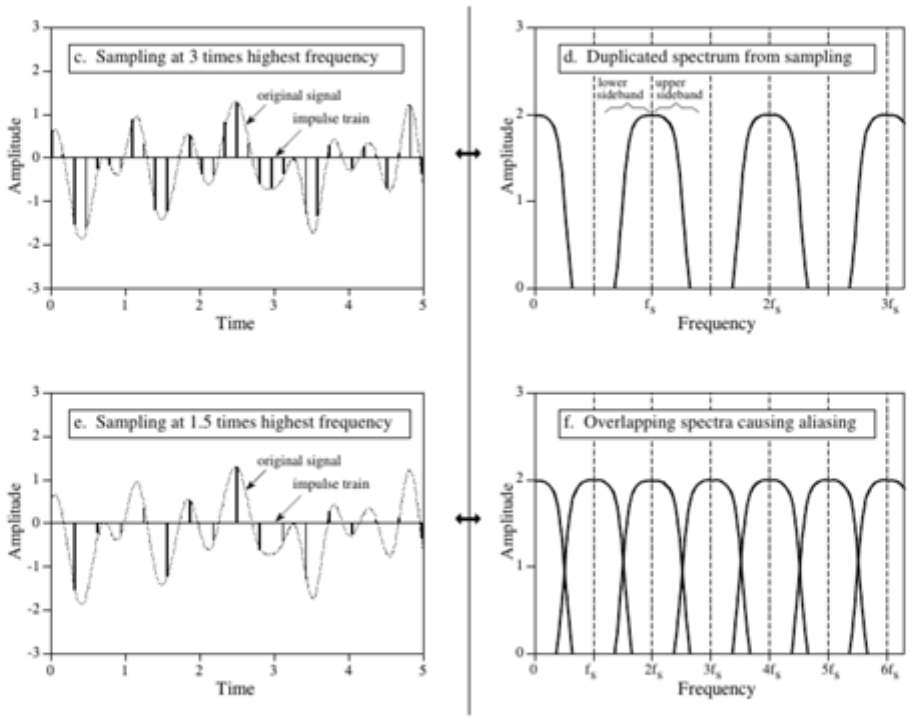


Figure A.1: The first signal is sampled at 3 times the highest composing frequency. Its frequency spectrum shows no overlap between sidebands. The second signal is sub-sampled and higher frequencies overlap lower frequencies in its frequency spectrum. This illustrates a key observation: discarding samples in one domain causes aliasing in the other [53].

transforms. The Fourier transform simply decomposes a continuous periodic signal into a series of sinusoidal terms. A detailed description can be found in Smith [53, ch 8-12,31], but the general relations governing conversion between the time and frequency domains (and therefore between the intensity and visibility spaces) are as follows:

$$G(f) = \int_{-\infty}^{\infty} g(x) e^{-2\pi i f x}$$
$$g(x) = \int_{-\infty}^{\infty} G(f) e^{2\pi i f x}$$

Here capital letters indicate the Fourier space. We will be using this convention throughout this document. Note that the Fourier transform conserves the total energy between the signal and Fourier spaces (Parseval's Theorem) if all samples are available in both domains (unlike in radio interferometry). It is also possible to take the Fourier transform with discrete signals (indicated with square brackets as per convention). In that case the Fast Fourier Transform algorithm and its inverse is employed to move between these domains. There are highly optimized libraries available for example FFTW or cuFFT.

For linear systems the following theorems are true (stated without proof):

$$
\begin{aligned}
g_1(x) * g_2(x) &:= \int_{-\infty}^{\infty} g_1(t) g_2(x-t) \ \textit{(convolution)} \\
&= g_2(x) * g_1(x) \textit{(commutativity)} \\
g_1(x) \star g_2(x) &:= \int_{-\infty}^{\infty} g_1(t) g_2(t+x) \ \textit{(cross-correlation)} \\
&= g_1^*(-x) * g_2(x)
\end{aligned}
$$

Addition Theorem:

$$\mathcal{F}(g_1(x) + g_2(x)) = G_1(f) + G_2(f)$$

Convolution Theorem:

$$\mathcal{F}(g_1(x) * g_2(x)) = G_1(f).G_2(f)$$

$$\mathcal{F}(g_1(x).g_2(x)) = G_1(f) * G_2(f)$$

Shift Theorem:

$$\mathcal{F}(g(x-\Delta)) = G(f)e^{-2\pi i f \Delta}$$

In image processing the convolution operation is known as filtering, the cross-correlation can be seen as measuring the correspondance between two signals (one of which may be delayed). The definitions for the two dimensional versions of convolution, cross-correlation, Fourier transform and its inverse are analogous to their one dimensional counterparts. The two dimensional Fourier transform first transforms over one axis and then transforms the output on the second axis.

Bibliography

[1] Shameen Akhter and Jason Roberts. *Multi-core programming*, volume 33. Intel press Hillsboro, 2006.

[2] S Bhatnagar, TJ Cornwell, K Golap, and Juan M Uson. Correcting direction-dependent gains in the deconvolution of radio interferometric images. *Astronomy & Astrophysics*, 487(1):419–429, 2008.

[3] Lamont V Blake and Maurice Long. *Antennas: Fundamentals, Design, Measurement, 3rd Edn (Standard)*. IET, 2009.

[4] OpenMP Architecture Review Board. Openmp application programming interface. Technical Report 4.5, 2015.

[5] R. Bolton, R. P. Millenaar, and G. D. Harris. Ska configurations design document. Technical Report WP3 050.020.000 R 002 (rev. A), Square Kilometre Array Consortium, 2011.

[6] RS Booth, WJG De Blok, JL Jonas, and B Fanaroff. Meerkat key project science, specifications, and proposals. *arXiv preprint arXiv:0910.2935*, 2009.

[7] Ian Buck, Tim Foley, Daniel Horn, Jeremy Sugerman, Kayvon Fatahalian, Mike Houston, and Pat Hanrahan. Brook for gpus: stream computing on graphics hardware. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 777–786. ACM, 2004.

[8] Mark R Calabretta and Eric W Greisen. Representations of celestial coordinates in fits. *Astronomy & Astrophysics*, 395(3):1077–1122, 2002.

[9] Christopher Carilli and Steve Rawlings. Science with the square kilometer array: motivation, key science projects, standards and assumptions. *arXiv preprint astro-ph/0409274*, 2004.

[10] Wilbur Norman Christiansen and Jan Arvid Högbom. *Radiotelescopes*. Cambridge University Press, 1969.

[11] William T Cochran, James W Cooley, David L Favin, Howard D Helms, Reg Kaenel, William W Lang, George C Maling Jr, David E Nelson, Charles M Rader, Peter D Welch, et al. What is the fast fourier transform? *Proceedings of the IEEE*, 55(10):1664–1674, 1967.

[12] Tim J Cornwell, Kumar Golap, and Sanjay Bhatnagar. The noncoplanar baselines effect in radio interferometry: The w-projection algorithm. *Selected Topics in Signal Processing, IEEE Journal of*, 2(5):647–657, 2008.

[13] TJ Cornwell and RA Perley. Radio-interferometric imaging of very large fields-the problem of non-coplanar arrays. *Astronomy and Astrophysics*, 261:353–364, 1992.

[14] TJ Cornwell, MA Voronkov, and Ben Humphreys. Wide field imaging for the square kilometre array. In *SPIE Optical Engineering+ Applications*, pages 85000L–85000L. International Society for Optics and Photonics, 2012.

[15] Intel Corporation. Intel 64 and ia-32 architectures optimization reference manual. Technical report, 2015.

[16] NVIDIA Corporation. Whitepaper, nvidia's next generation cuda compute architecture: Kepler gk110. Technical Report 1.0, 2012.

[17] NVIDIA Corporation. Geforce gtx 980 whitepaper. Technical Report 1.1, NVIDIA Corporation, 2014.

[18] W. D. Cotton. Multi-facet cleaning in obit. Technical Report 15, National Radio Astronomy Observatory, Charlottesville, Virginia. Associated Universities, Inc., 2009.

[19] W. D. Cotton. Parallel facet imaging in obit. Technical Report 6, National Radio Astronomy Observatory, Charlottesville, Virginia. Associated Universities, Inc., 2009.

[20] Arwa Dabbech, Chiara Ferrari, David Mary, Eric Slezak, Oleg Smirnov, and Jonathan S Kenyon. Moresane: Model reconstruction by synthesis-analysis estimators-a sparse deconvolution algorithm for radio interferometric imaging. *Astronomy & Astrophysics*, 576:A7, 2015.

[21] RG Edgar, MA Clark, K Dale, Daniel A Mitchell, Stephen M Ord, Randall B Wayth, H Pfister, and Lincoln J Greenhill. Enabling a high throughput real time data pipeline for a large radio telescope array with gpus. *Computer Physics Communications*, 181(10):1707–1714, 2010.

[22] Richard Gerber, Aart J.C. Bik, Kevin B. Smith, and Xinmin Tian. *The software optimization cookbook*. Intel Press, 2002.

[23] Kumar Golap. Mutithreading gridders without copying and locks. 2015.

[24] David Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys (CSUR)*, 23(1):5–48, 1991.

[25] Eric W. Greisen. Non-linear coordinate systems in aips, reissue of november 1983 version. Technical Report 27, National Radio Astronomy Observatory. Associated Universities, Inc., 1993.

[26] Khronos Group and other. Opencl 2.1 api specification. Technical Report 2.1, 2015.

[27] Nicholas J Higham. The accuracy of floating point summation. *SIAM Journal on Scientific Computing*, 14(4):783–799, 1993.

[28] B Humphreys and T Cornwell. Analysis of convolutional resampling algorithm performance. *SKA Memo*, 132, 2011.

[29] John I Jackson, Craig H Meyer, Dwight G Nishimura, and Albert Macovski. Selection of a convolution function for fourier inversion using gridding [computerised tomography application]. *Medical Imaging, IEEE Transactions on*, 10(3):473–478, 1991.

[30] Joseph C Jacob, Daniel S Katz, Thomas Prince, G Bruce Berriman, John C Good, Anastasia C Laity, Ewa Deelman, Gurmeet Singh, and Mei-Hui Su. *The montage architecture for grid-enabled science processing of large, distributed datasets*. Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2004.

[31] S Jaeger. The common astronomy software application (casa). In *Astronomical Data Analysis Software and Systems XVII*, volume 394, page 623, 2008.

[32] Simon Johnston, R Taylor, Matthew Bailes, N Bartel, C Baugh, M Bietenholz, Chris Blake, R Braun, J Brown, S Chatterjee, et al. Science with askap. *Experimental Astronomy*, 22(3):151–273, 2008.

[33] David B Kirk and W Hwu Wen-mei. *Programming massively parallel processors: a hands-on approach*. Newnes, 2012.

[34] Leonid Kogan and Eric W. Greisen. Faceted imaging in aips. Technical Report 113, National Radio Astronomy Observatory. Associated Universities, Inc., 2009.

[35] JP McMullin, B Waters, D Schiebel, W Young, and K Golap. Casa architecture and applications. In *Astronomical data analysis software and systems XVI*, volume 376, page 127, 2007.

[36] Enno Middelberg and Uwe Bach. High resolution radio astronomy using very long baseline interferometry. *Reports on Progress in Physics*, 71(6):066901, 2008.

[37] Daniel Muscat. High-performance image synthesis for radio interferometry. *arXiv preprint arXiv:1403.4209*, 2014.

[38] JE Noordam and OM Smirnov. The meqtrees software system and its use for third-generation calibration of radio interferometers. *Astronomy & Astrophysics*, 524:A61, 2010.

[39] CUDA Nvidia. Compute unified device architecture programming guide. Technical Report 7.5, 2015.

[40] AR Offringa, B McKinley, Natasha Hurley-Walker, FH Briggs, RB Wayth, DL Kaplan, ME Bell, L Feng, AR Neben, JD Hughes, et al. Wsclean: an implementation of a fast, generic wide-field imager for radio astronomy. *Monthly Notices of the Royal Astronomical Society*, 444(1):606–619, 2014.

[41] John D Owens, Mike Houston, David Luebke, Simon Green, John E Stone, and James C Phillips. Gpu computing. *Proceedings of the IEEE*, 96(5):879–899, 2008.

[42] David A Patterson and John L Hennessy. *Computer organization and design: the hardware/software interface*. Morgan Kaufmann, 3 edition, 2013.

[43] William D Pence, L Chiappetti, Clive G Page, RA Shaw, and E Stobie. Definition of the flexible image transport system (fits), version 3.0. *Astronomy & Astrophysics*, 524:A42, 2010.

[44] R. A. Perley, C. J. Chandler, B. J. Butler, and J. M. Wrobel. The expanded very large array: A new telescope for new science. *The Astrophysical Journal Letters*, 739(1):L1, 2011.

[45] Donald R Rhodes. On the spheroidal functions. *J. Res. Nat. Bureau S*, pages 187–209, 1970.

[46] John W Romein. An efficient work-distribution strategy for gridding radio-telescope data on gpus. In *Proceedings of the 26th ACM international conference on Supercomputing*, pages 321–330. ACM, 2012.

[47] RJ Sault, L Staveley-Smith, and WN Brouw. An approach to interferometric mosaicing. *Astronomy and Astrophysics Supplement Series*, 120:375–384, 1996.

[48] FR Schwab. Optimal gridding of visibility data in radio interferometry. In *Indirect Imaging. Measurement and Processing for Indirect Imaging*, volume 1, pages 333–346, 1984.

[49] O. M. Smirnov. Revisiting the radio interferometer measurement equation. I. A full-sky Jones formalism. *Astron.Astrophys.*, 527:A106, March 2011.

[50] O. M. Smirnov. Revisiting the radio interferometer measurement equation. II. Calibration and direction-dependent effects. *Astron.Astrophys.*, 527:A107, March 2011.

[51] O. M. Smirnov. Revisiting the radio interferometer measurement equation. III. Addressing direction-dependent effects in 21 cm WSRT observations of 3C 147. *Astron.Astrophys.*, 527:A108, March 2011.

[52] O. M. Smirnov. Revisiting the radio interferometer measurement equation. IV. A generalized tensor formalism. *aap*, 531:A159, July 2011.

[53] Steven W Smith et al. The scientist and engineer's guide to digital signal processing. 1997.

[54] Sze Meng Tan. *Aperture-synthesis mapping and parameter estimation.* PhD thesis, University of Cambridge, 1986.

[55] Cyril Tasse. Facet imaging - tentative document. This document outlines Cyril's approach to correcting directional dependent terms using faceting, as well as an explanation on new hybrid w-faceting techniques that are more accurate than that of Kogan and Greisen (AIPS Memo series 119). The document is available on the repository `https://github.com/cyriltasse/DocFacetMachines/` under commit 029a9723915882a1c77497bfc6410947d89d32c8.

[56] Cyril Tasse, S van der Tol, J van Zwieten, Ger van Diepen, and S Bhatnagar. Applying full polarization a-projection to very wide field of view instruments: An imager for lofar. *Astronomy & Astrophysics*, 553:A105, 2013.

[57] Greg B Taylor, Chris Luke Carilli, and Richard A Perley. Synthesis imaging in radio astronomy ii. In *Synthesis Imaging in Radio Astronomy II*, volume 180, 1999.

[58] Philippe Thévenaz, Thierry Blu, and Michael Unser. Image interpolation and resampling. *Handbook of medical imaging, processing and analysis*, pages 393–420, 2000.

[59] A Richard Thompson, James M Moran, and George W Swenson Jr. *Interferometry and synthesis in radio astronomy.* John Wiley & Sons, 2008.

[60] AR Thompson and RN Bracewell. Interpolation and fourier transformation of fringe visibilities. *The Astronomical Journal*, 79:11–24, 1974.

[61] AL Varbanescu, A van Amesfoort, T Cornwell, BG Elmegreen, R van Nieuwpoort, G van Diepen, and H Sips. The performance of gridding/degridding on the cell. *month*, 2008.

[62] M.H. Wieringa and Kemball A.J. Measurementset definition version 2.0. Technical Report 229, National Radio Astronomy Observatory. Associated Universities, Inc. and Australia Telescope National Facility, 2000.

[63] M.H. Wieringa and Cornwell T.J. Definition of measurementset aips++. Technical Report 191, National Radio Astronomy Observatory. Associated Universities, Inc. and Australia Telescope National Facility, 1996.

[64] Thomas L Wilson, Kristen Rohlfs, and Susanne Hüttemeister. *Tools of radio astronomy*, volume 86. Springer, 2009.

[65] Mark Yashar and Athol Kemball. Tdp calibration & processing group cpg memo# 3 computational costs of radio imaging algorithms dealing with the non-coplanar baselines effect: I. 2009.