

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/314373350>

Multi-GPU maximum entropy image synthesis for radio astronomy

Article · March 2017

CITATIONS

0

READS

18

5 authors, including:



Miguel Cárcamo

University of Santiago, Chile

3 PUBLICATIONS 0 CITATIONS

[SEE PROFILE](#)



Pablo Enrique Roman

University of Santiago, Chile

44 PUBLICATIONS 347 CITATIONS

[SEE PROFILE](#)



Fernando R Rannou

University of Santiago, Chile

39 PUBLICATIONS 1,856 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Protoplanetary disks [View project](#)



Maximum entropy image synthesis for radio astronomy [View project](#)

All content following this page was uploaded by [Pablo Enrique Roman](#) on 03 July 2017.

The user has requested enhancement of the downloaded file.

Multi-GPU maximum entropy image synthesis for radio astronomy

M. Cárcamo^a, P. Román^b, S. Casassus^c, V. Moral^c, F.R. Rannou^a

^a*Departamento de Ingeniería Informática, Universidad de Santiago de Chile, Av. Ecuador 3659, Santiago, Chile*

^b*Center for Mathematical Modeling, Universidad de Chile, Av. Blanco Encalada 2120 Piso 7, Santiago, Chile*

^c*Astronomy Department, Universidad de Chile, Camino El Observatorio 1515, Las Condes, Santiago, Chile*

Abstract

The maximum entropy method (MEM) is a well known deconvolution technique in radio-interferometry. This method solves a non-linear optimization problem with an entropy regularization term. Other heuristics such as CLEAN are faster but highly user dependent. Nevertheless, MEM has the following advantages: it is unsupervised, it has an statistical basis, it has a better resolution and better image quality under certain conditions. This work presents a high performance GPU version of non-gridded MEM, which is tested using interferometric and simulated data. We propose a single-GPU and a multi-GPU implementation for single and multi-spectral data, respectively. We also make use of the Peer-to-Peer and Unified Virtual Addressing features of newer GPUs which allows to exploit transparently and efficiently multiple GPUs. Several ALMA data sets are used to demonstrate the effectiveness in imaging and to evaluate GPU performance. The results show that a speedup from 1000 to 5000 times faster than a sequential version can be achieved, depending on data and image size. This has allowed us to reconstruct the HD142527 CO(6-5) short baseline data set in 2.1 minutes, instead of the 2.5 days that takes on CPU.

Keywords: Maximum entropy, GPU, ALMA, SKA, LOFAR, Radio interferometry, Image synthesis

1. Introduction

Current operating radio astronomy observatories (e.g. ALMA, VLA, ATCA) consist of a number of antennas capable of collecting radio signals from specific sources. Each antenna's signal is correlated with every other signal to produce samples of the sky image, but on the Fourier domain (Candan et al., 2000). These samples are called visibilities and comprise a sparse and irregularly sampled set of complex numbers in the (u, v) plane. A typical ALMA sampling data set contains from 10^4 to more than 10^9 sparse samples in one or more frequency channels.

In the case where $V(u, v)$ is completely sampled, Equation 1 states the simple linear relationship between image and data. Thus the image can be recovered by Fourier inversion of the interferometric signal (Clark, 1999). In this equation, kernel $A(x, y)$ is called the primary beam (PB) and corresponds to the solid angle reception pattern of the antenna array and is modelled as a Gaussian function (Taylor et al., 1999).

$$V(u, v) = \int_{\mathbb{R}^2} A(x, y) I(x, y) e^{-2\pi i(ux+vy)} dx dy \quad (1)$$

In the real scenario of collecting noisy and irregularly sampled data, this problem is not well defined (Marechal and Wallach, 2009; Chen, 2011). To approximate the inverse problem of recovering the image from

a sparse and irregularly sampled Fourier data a process called Image Synthesis (A. Richard Thompson, 2004) or Fourier Synthesis (Marechal and Wallach, 2009) is used. Current interferometers are able to collect a large number of (observed) samples in order to fill as much as possible the Fourier domain. As an example, Figure 1 shows the ALMA 400 meter short baseline sampling for Cycle 2 observation of the HD142527 protoplanetary disc. Additionally, the interferometer is able to estimate data variance σ_k^2 per visibility as a function of the antenna thermal noise (A. Richard Thompson, 2004).

Many algorithms have been proposed for solving the image synthesis problem and the standard procedure is the CLEAN heuristic (Hogbom, 1974). This algorithm is based on the dirty image/beam representation of the problem (Taylor et al., 1999), which results in a deconvolution problem. CLEAN has been interpreted as a greedy class heuristic, particularly for point-like sources as a matching pursuit algorithm (Lannes et al., 1997). Image reconstruction in CLEAN is performed in the image space using the convolution relationship, and it is therefore quite efficiently implemented using FFTs. This algorithm is also supervised. The user could indicate iteratively in which region of the image the algorithm should focus. However, statistical interpretation of resulting images and remaining artifacts are far to be described by a well founded theory.

MEM is inspired on a maximum likelihood argumentation since interferometer measurements are assumed to be corrupted by Gaussian noise. Reconstructed im-

Email address: miguel.carcamo@usach.cl (M. Cárcamo)

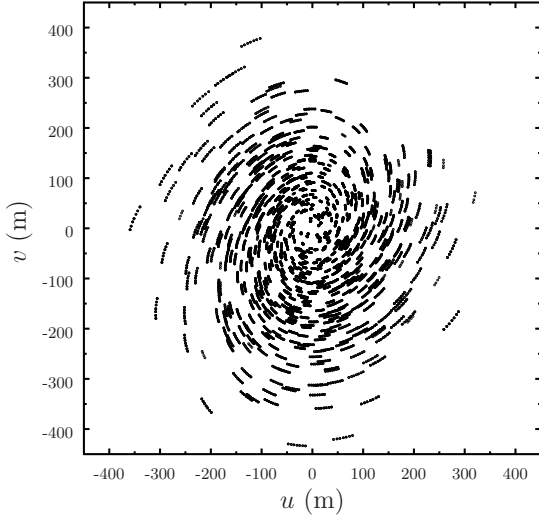


Figure 1: Short baseline uv sampling of the HD142527 protoplanetary disk.

ages with this method have been considered to have higher resolution and less artifacts than CLEAN images (Cornwell and Evans, 1985a; Narayan and Nityananda, 1986; Donoho et al., 1992). MEM was the second choice image synthesis algorithm, and mainly used for checking CLEAN bias and in images with complex structure (Neff et al., 2015; Coughlan and Gabuzda, 2013; Warmuth and Mann, 2013). However, routine use of MEM with large data sets has been hindered due to its high computational demands (Cornwell and Evans, 1985b; Taylor et al., 1999; Narayan and Nityananda, 1986). For instance, to reconstruct an image of $2^{11} \times 2^{11}$ pixels from the ALMA Long Baseline campaign dataset HL Tau Band 3 (279921600 visibilities), MEM computes 7.9×10^{16} floating-point operations per iteration, approximately. It was also claimed that MEM could not reconstruct point-source like images over a plateau e.g. (Rastorgueva, E. A. et al., 2011). However, we have found (Section 4) this not to be true using a simulated data set. In fact, the experiments show MEM can reconstruct the point source and maintain a smooth plateau.

The MEM algorithm is traditionally implemented with optimization methods (OM) based on the gradient (Cornwell and Evans, 1985b), also called first order methods. Typical examples of such methods are the conjugate gradient (Press et al., 1992), quasi-newton methods like the modified Broyden’s L-BFGS-B (Nocedal and Wright, 2006), and the Nesterov’s class of accelerated gradient descent (Nesterov, 2004). All of them require the first derivative calculation which has a computation complexity of $O(M \cdot Z)$, where Z is the number of data and M is the number of image pixels. The gradient computation is the most expensive part per iteration of first order OMs. In consequence, such OMs are equivalent per iteration from the complexity point of view. We choose to study the GPU implementation for the Conjugate Gradient (CG) in the

positive projected version due its simplicity for the case of image synthesis and large data ($Z \sim 10^9$ and $M \sim 10^7$).

A GPU implementation that supports Bayesian Inference for Radio Observations (Lochner et al., 2015) has been proposed (Perkins et al., 2015) before. This approach uses Bayesian inference to sample a parameter set representing a sky model to propose visibilities that best match the model. Their model handles point and Gaussian sources. Our approach employs a non-linear optimization problem to directly solve the Bayesian model with a maximum entropy prior using real and synthetic data.

In this paper we present a high performance, high-throughput implementation of MEM for large scale, multi-frequency image synthesis. Our aim is to demonstrate computational performance of the algorithm making its practical for research in radio-interferometry for today data sets. The main features of the solution are:

- **GPU implementation:** With the advent of larger interferometric facilities such as SKA (Quinn et al., 2015) and LOFAR (van Haarlem et al., 2013), efficient image synthesis based on optimization cannot be delivered by nowadays multi-core computers. However, we have found that the image synthesis problem, as formulated by MEM, fits well into the SIMT paradigm, such that the solution can make efficient use of the massive array of cores of current GPUs. Also, our GPU proposal exploits features like the Unified Virtual Addressing (UVA) and the *Peer-to-peer* (P2P) communications to devise a multiple GPU solution for large multi-frequency data.
- **Unsupervised reconstruction:** Supervised algorithms could become obsolete in this high-throughput regime in favor of more automatic methods based on a fitting criteria.
- **Non-gridded approach:** A gridded implementation extrapolates scattered samples on the regular grid of Fourier space, which results in a partially filled grid and enables the use of the FFT (Taylor et al., 1999; Winkel et al., 2016). A given pixel in Fourier space typically surround several visibilities. Although this processing reduces the computation time by averaging data, it does so in detriment of further statistical fit. The use of a high performance implementation allows us to process the full dataset without gridding, and still achieve excellent computational performance.
- **Mosaic support:** Mosaic images in interferometry allows the study of large scale object in the sky. This feature is implemented as in traditional interferometry (A. Richard Thompson, 2004; Taylor et al., 1999).
- **Multi-frequency support:** Image synthesis with short-spaced multiple channels can be approximated by a constant spectral dependence. In some cases the spectral dependency introduces strong effects into image synthesis (Rau and Cornwell, 2011). In this

proposal we reconstruct datasets with several channels, but with small spectral variation.

2. Method and implementation

This section describes the mathematical formulation of the mono-frequency *Maximum Entropy Method* (MEM) and the multi-frequency MEM together with a positively constrained conjugate gradient minimization algorithm. Finally, the GPU implementation details are given.

2.1. Data description

Let $V^o = \{V_k^o\}$, $k = 0, \dots, Z - 1$, be the (observed) visibility data, let $\{\sigma_k\}$ $k = 0, \dots, Z - 1$ the estimated deviations, and let $I = [I_{m,n}]$, $m = 0, \dots, M - 1$, $n = 0, \dots, M - 1$ be the image to be reconstructed from V^o . Notice that I is a regularly sampled, and usually square image function, whereas V^o is a list of sparse points.

Visibilities are sampled in certain spectral windows, which are contiguous spectrum whose frequencies are uniformly spaced and also uniformly divided in channels. Therefore, for the case of multi-spectral, multi-frequency data the observed visibilities are indexed as $V_{w,c,k}^o$ where w corresponds to a spectral window, and c corresponds to a channel of the spectral window w .

Interferometric data also considers polarization. We simplify our approach by considering only the case of intensity polarization (I). This simplification is rather general in practice. Current radio interferometers like ALMA, VLA, or ATCA are commonly generating data on orthogonal (XX and YY) polarizations or even (LL and RR), which can enter directly in the algorithm as I polarization e.g. (Taylor et al., 1999).

2.2. Maximum Entropy Method

The deconvolution problem consists of selecting one image from many feasible. MEM is an important class of image restoration algorithms that selects the image that fits with the measured visibilities, to within the noise level, and whose pixel values satisfy the maximum entropy restriction (A. Richard Thompson, 2004; Cornwell, 1988).

In MEM, image and data are considered random variables, with known probability distributions. Let $P(V^o|I)$ be the likelihood of observing the visibilities given the image I and let $P(I)$ be an *a priori* knowledge of the image. Then, by Bayes theorem we obtain the *a posteriori* probability.

$$P(I|V^o) = \frac{P(V^o|I)P(I)}{P(V^o)} \quad (2)$$

The likelihood $P(V^o|I)$ can be approximated using the fact that visibilities are independent and identically distributed Gaussian random variables corrupted by heterocedastics Gaussian noise of mean zero and standard deviation σ_k . Then, the likelihood can be expressed as

$$P(V^o|I) \propto \exp \left\{ - \prod_k \left| \frac{V_k^o - V_k^m(I)}{\sigma_k} \right|^2 \right\} \quad (3)$$

where $V^m(I)$ denotes the model visibilities which are functions of the image estimate. This expression considers the error in estimated visibilities without using a gridding approach.

The MEM image prior is modeled considering a multinomial distribution of a discrete total intensity (Sutton and Wandelt, 2006) that covers the entire image, in analogy with a CCD camera array receiving random photons from the sky. Let M be the number of image pixels and let N_i be the number of photons collected at pixel i . Then, the prior can be expressed as

$$P(I) = \frac{N!}{M^N \prod_i N_i!} \quad (4)$$

where $N = \sum_i N_i$. The image intensity at pixel i is interpreted in this model as $I_i \propto N_i$.

After replacing Equations (3) and (4) into Equation (2), taking logarithms, dropping terms independent of I , and using Stirling's approximation for factorials, we arrive at the following objective function:

$$\Phi = \frac{1}{2} \sum_k \left| \frac{V_k^o - V_{w,c,k}^m(I)}{\sigma_k} \right|^2 + \lambda \sum_{m,n} I_{m,n} \log \frac{I_{m,n}}{G} \quad (5)$$

The previous Bayesian argument (Sutton and Wandelt, 2006) has been generalized introducing parameters G and λ . The first parameter is positive ($G > 0$) and represents the minimal signal in the image or background noise. The penalization factor $\lambda \geq 0$ controls the relative importance between data and the entropy term. When $\lambda = 0$ the problem becomes a least-square optimization problem, and as λ increases less importance is given to the data and the solution becomes smoother. We recognize Equation (5) as a typical χ^2 term plus a Shannon's entropy term S , that is,

$$\Phi(I, V^o; \lambda, G) = \chi^2(V^o, I) + \lambda S(I; G) \quad (6)$$

For multi-frequency data, the χ^2 term is expressed as:

$$\chi^2 = \frac{1}{2} \sum_{w=0}^{W-1} \sum_{c=0}^{C-1} \sum_{k=0}^{Z-1} \left| \frac{V_{w,c,k}^o - V_{w,c,k}^m(I)}{\sigma_{w,c,k}} \right|^2 \quad (7)$$

Finally, the MEM method for image synthesis consists of solving the following non-linear, constrained optimization problem:

$$\hat{I}_{\text{MEM}} = \arg \min_{I \geq G} \Phi(I; V^o; \lambda, G) \quad (8)$$

2.3. Objective function evaluation

The minimization problem (8) can be solved by a constrained Conjugate Gradient (CG) algorithm, which repeatedly evaluates the objective function and its gradient to compute the search direction and step size. Evaluation of the entropy term is straightforward, but evaluation of the χ^2 term requires some additional processing. The following steps are required for computation of model visibilities V^m required for evaluating χ^2 .

1. The attenuation image $A_{m,n}$ or primary beam represents the discrete version of the reception pattern of the telescope in Equation 1. This image is modelled as a Gaussian and depends on the radio-interferometer (Taylor et al., 1999) scaling linearly with the frequency. For ALMA the FWHM is 21 arcsec at 300GHz for antennas of 12m diameter.
2. A gridded version of the model visibilities V^g is obtained by applying a 2D Discrete Fourier Transform. For the case of multi-frequency data, an attenuation matrix $A_{w,c}$ is built for each channel of each spectral window, resulting in a set of gridded visibilities, $V_{w,c}^g$. Thus, the Fourier modulation and the interpolation step are carried out independently for each frequency channel.

$$V^g = \mathcal{F}_{2D}\{A \cdot I\}$$

In case of a mosaic measurement, for each pointing in the sky an attenuation image is calculated and centered in the corresponding field of view.

3. The phase-tracking of the object has a center according to the celestial sphere (Taylor et al., 1999). This implies that the image is shifted according to that center. A modulation factor is applied to the gridded visibilities as follows:

$$\hat{V}^g(u, v) = V^g(u, v) \exp\{2\pi i(ux_c + vy_c)/M\}$$

where (u, v) are the uniformly spaced grid locations, and (x_c, y_c) are the direction cosines (Taylor et al., 1999) of the phase-tracking center. In case of mosaic image, data is divided into several pointing in the sky shifting each field of view according to its location.

4. Model visibilities V^m are approximated from gridded visibilities \hat{V}^g by using an interpolation method. A bilinear interpolation method (Burger and Burge, 2010) was chosen for approximating observed visibilities surrounded by four grid point.
5. Residual visibilities $V^R = \{V_k^R\}$ are calculated from $V_k^R = V_k^o - V_k^m$ in order to compute the χ^2 term.

It is worth emphasizing that our algorithm computes the error term for each visibility at their exact uv locations, and do not apply any type of Fourier data gridding to reduce the amount of computation.

2.4. Gradient evaluation

Gradient computation is required for first order optimization methods. Notice that before gradient host func-

tion is called, the objective function is always executed. Thus, residual visibilities V^R are available for gradient calculation. As well as the objective function the first computations has to do with the entropy gradient. Therefore, every coordinate of the entropy gradient (∇S) has a value that follows Equation 9.

$$[\nabla S]_{m,n} = 1 + \log \frac{I_{m,n}}{G} \quad (9)$$

The χ^2 gradient term is calculated from the derivative of the Direct Fourier Transform of the image, as shown in Equation 10, where $X_{m,n} = (x, y)_{m,n}$ is the image coordinate according to the direction cosines of the phase tracking center, and U_k is the (u, v) sampling coordinate of the k th visibility.

$$[\nabla \chi^2]_{m,n} = \sum_{k=0}^{Z-1} \frac{\text{Re}(V_k^R \exp(2\pi i X_{m,n} \cdot U_k))}{\sigma_k^2} \quad (10)$$

2.5. SIMT and implementation

We have implemented a positive constrained CG algorithm to solve Equation 8. The general principles that guide the SIMT solution are:

1. Write simple and short kernels to use a small number of registers per thread, increasing warp number per Streaming Multiprocessor.
2. Write kernels with spatial locality access to increase memory transfer and cache hit rate.
3. Keep most of the data in global memory to avoid slow transfers from host to device and vice versa.

The algorithm's main iteration loop is kept in host, while compute intensive functions are implemented in GPUs. To minimize data transfer, most vectors and matrices are allocated and kept in device global memory. Only the image estimate is transferred back and forth between host and device. At convergence, model visibilities are also sent back to host to compute final residuals.

As an example, here we show implementation details of the two most compute intensive functions, namely Φ and $\nabla \Phi$. Algorithm 1 shows the χ^2 host function that invokes 1D and 2D kernels to accomplish its goal. Lines 2 to 5 apply correction factors and the Fourier transformation steps. Although not shown here, each kernel invocation has an associated grid on which threads are run. For instance, the attenuation and modulation kernels on line (2) and (4), respectively, use a $M \times M$ kernel grid, while the interpolation kernel at line 5 uses a 1D grid of T threads, where T is the next higher power of 2, greater than or equal to Z . All 2D grids are organized in blocks of 32×32 threads, while 1D grids are organized in blocks of 1024×1 threads. The last two steps correspond to the residuals vector computation (line 6) and its global reduction sum (line 7).

Algorithm 1 χ^2 host function

```
1: Chi2( $I, V^o, x_c, y_c, \Delta u, \Delta v$ )
2:  $I_a = \mathbf{KAttenuation}(I, A)$ 
3:  $V^g = \mathbf{KcudaFFT}(I_a)$ 
4:  $\hat{V}^g = \mathbf{KModulation}(V^g, x_c, y_c)$ 
5:  $V^m = \mathbf{KInterpolation}(\hat{V}^g, u/\Delta u, v/\Delta v)$ 
6:  $C = \mathbf{KChi2Res}(V^m, V^o)$ 
7:  $\mathbf{KReduce}(C)$ 
```

Algorithm 2 χ^2 1D Kernel

```
1:  $\mathbf{KChi2}(V^m, V^o)$ 
2:  $j = \text{blockDim.x} * \text{blockIdx.x} + \text{threadIdx.x}$ 
3: if  $j < Z$  then
4:    $V_j^R = V_j^o - V_j^m$ 
5:    $\chi^2(j) = W(j) \cdot (\text{Re}(V^R(j))^2 + \text{Im}(V^R(j))^2)$ 
6: end if
```

Algorithm 2 shows the $\mathbf{KChi2Res}$ kernel. First, each thread computes its index into the grid (line 2), and then computes the corresponding contribution to the χ^2 term (lines 4 and 5).

Algorithm 3 lists the pseudo-code for the calculation of the gradient from Equation 10, which is the most compute intensive kernel. This kernel uses a $M \times M$ grid. The first lines (2 and 3) compute the index into the 2D grid. Then every thread calculates their corresponding contribution to a cell of the gradient, for every visibility. Recall that sine and cosine functions are executed in the special function units (SFU) of a GPU. Therefore, if Tesla Kepler GK210B has 32 SFU and 192 cores per SM, an inevitable hardware bottleneck occurs.

Algorithm 3 $\nabla\chi^2$ 2D Kernel

```
1:  $\mathbf{KGradChi2}(V^R, W, x_c, y_c, \Delta x, \Delta y)$ 
2:  $k = \text{blockDim.x} * \text{blockIdx.x} + \text{threadIdx.x}$ 
3:  $i = \text{blockDim.y} * \text{blockIdx.y} + \text{threadIdx.y}$ 
4:  $x = (k - x_c) \cdot \Delta x$ 
5:  $y = (i - y_c) \cdot \Delta y$ 
6:  $\nabla\chi^2(N \cdot i + k) = 0$ 
7: for  $j = 0$  to  $Z - 1$  do
8:    $\nabla\chi^2(N \cdot i + k) += W(j) * [\text{Re}(V^R(j)) \cdot \cos(2\pi \langle (u_j, v_j), (x, y) \rangle) - \text{Im}(V^R(j)) \cdot \sin(2\pi \langle (u_j, v_j), (x, y) \rangle)]$ 
9: end for
```

2.6. Multi-GPU Strategy

The main idea of using several GPUs comes from Equation 7. Since the contribution of every channel to the χ^2 term can be calculated independently from any other channel, it is possible to schedule different channels to different devices, and then do the sum reduction in host. Generally, the number of channels is larger than the number of devices and it is therefore necessary to distribute channels as equal as possible to achieve a balanced workload.

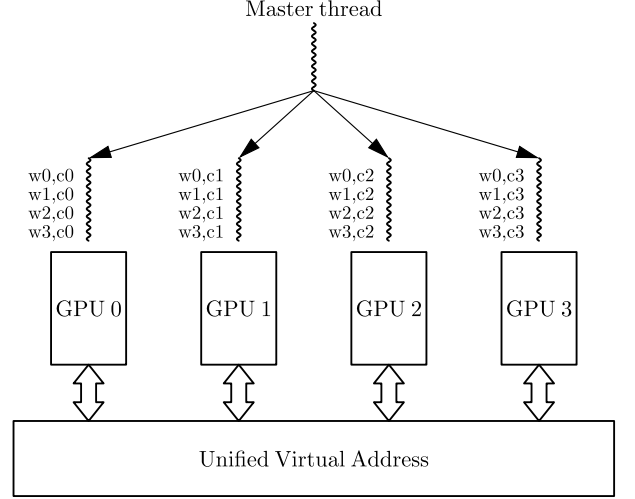


Figure 2: Illustration of the multi-thread strategy for multi-GPU reconstruction. The figure shows an example of how spectral windows (w_0, w_1, w_2, w_3) and channels (c_0, c_1, c_2, c_3) are distributed with a round-robin scheduling. Data is copied to the UVA before kernel invocation.

Kernels executing on 64-bits systems and on devices with compute capability 2.0 and higher (CUDA version higher than 4.0 is also required) can use *Peer-to-peer* (P2P) to reference a pointer whose memory pointed to has been allocated in any other device connected to the same PCIe root node. This, together with the Unified Virtual Addressing (UVA), which maps host memory and device global memory into a single virtual address space can be combined to access memory on any device transparently. In other words, the programmer does not need to explicitly program memory copies from one device to another. However, since memory referencing under P2P and UVA has to be done in the same process, OpenMP is used (OpenMP Architecture Review Board, 2015) to create as many host threads as devices to distribute channels.

Figure 2 illustrates the multi-GPU reconstruction strategy for the case of data with four spectral windows and four channels each, on a four device system. A master thread creates a team of four worker threads which are assigned channels in a round-robin fashion with portions of one channel at a time. Channel data is copied from host to the corresponding device by the master thread. Algorithm 4 shows the multi-frequency host function pseudo code that implements this idea. In line 2 a global variable is defined for the χ^2 term, and in line 3 the number of threads is set, always equal to the number of devices. Each thread invokes the $\mathbf{Chi2}$ host function (Algorithm 1) with the appropriate channel data, in parallel. Once the function is done, a critical section guarantees exclusive access to workers to update the shared variable. Since the update consists of adding a scalar value to the global sum, we have decided to keep this computation on host.

Computation of $\nabla\chi^2$ follows the same logic as before (see Algorithm 5). The major difference now is that the

$\nabla\chi^2$ is a vector allocated on GPU number zero, and a kernel is used to sum up partial results. Recall that with P2P and UVA, kernels can read and write data from and to any GPU.

Algorithm 4 Host multi-frequency χ^2 function

```

1: ParChi2( $I$ )
2:  $\chi^2 = 0$ 
3: set_num_threads( $N_{Devices}$ )
4: #pragma omp parallel for
5: for  $i = 0$  to  $TOTALCHANNELS - 1$  do
6:   cudaSetDevice( $i \% N_{Devices}$ )
7:    $\chi_i = \mathbf{Chi2}(I, V_i^o, x_c, y_c, \Delta u, \Delta v)$ 
8:   #pragma omp critical
9:    $\chi^2 = \chi^2 + \chi_i^2$ 
10: end for

```

Algorithm 5 Host multi-channel $\nabla\chi^2$ function

```

1: KParGradChi2( $V^R$ )
2:  $\nabla\chi^2 = 0$ 
3: set_num_threads( $N_{Devices}$ )
4: #pragma omp parallel for
5: for  $i = 0$  to  $TOTALCHANNELS - 1$  do
6:   cudaSetDevice( $i \% N_{Devices}$ )
7:    $\nabla\chi_i^2 = \mathbf{KGradChi2}(V_i^R, W_i, x_c, y_c, \Delta x, \Delta y)$ 
8:   #pragma omp critical
9:    $\mathbf{KSumGradChi2}(\nabla\chi^2, \nabla\chi_i^2)$ 
10: end for

```

3. Experimental Settings

In this section, we test our GPU version of MEM with synthetic and ALMA observatory data sets. We use them to demonstrate the effectiveness of the algorithm and to measure computational performance. As we pointed out in the introduction, this paper is not meant to be a comparison between MEM and CLEAN. This task would require an extensive imaging study which is beyond the scope of this work. Nevertheless, we find useful to display CLEAN images as a reference and because it is the most commonly method used today.

3.1. Data-sets

A 1024×1024 phantom image of a 7.4×10^{-4} (Jy/beam) point source located at the image center and surrounded by a plateau was used to show that MEM can indeed reconstruct isolated point sources. Image pixel size was set to 0.003 arcsec and the plateau’s magnitude was of 3.8×10^{-4} (Jy/beam). Synthetic interferometer data was created with `uvsim` (Casassus et al., 2015b) tool, and the HL Tauri uv coverage (explained below). Image resolution was measured by the Full-Width-Half-maximum (FWHM) for MEM and CLEAN images. For this particular example, $\lambda = 0.1$ was used in MEM.

Four ALMA data sets were used in our experiments. The first one was the gravitationally lensed galaxy SDP 8.1 on Band 7 with four spectral windows, four channels and a total of 121322400 non-flagged visibilities (Tamura et al., 2015). Channel 1 of spectral window 3 was completely flagged, so there were no visibilities from this particular channel. MEM images of 2048×2048 pixels were reconstructed, with $\lambda = 0.0$. The CLEAN image was downloaded from the ALMA Science Verification site, and had 3000×3000 pixels.

The second data set was the CO(6-5) emission line of the HD142527 protoplanetary disc, on band 9 with one channel and 107494 visibilities (Casassus et al., 2015a). This data set was used only for computational performance evaluation. Since this is the smallest data set of all, it was used to measure GPU occupancy. Larger data sets could not be used because current GPU hardware use 32 bits long counters, which are overflowed by bigger reconstructions.

To test mosaic support the Antennae Galaxies Northern mosaic Band 7 data set was used. This data had 23 fields, 1 channel, and 149390 visibilities. Once again, images of 512×512 pixels were reconstructed, but $\lambda = 0.1$ for MEM reconstruction.

Finally, the HL Tau Band 6 long baseline data set that corresponds to the observations of the young star HL Tauri surrounded by a protoplanetary disk was used (ALMA Partnership et al., 2015; Pinte et al., 2016; Tamayo et al., 2015). Spectral window zero (four channels) of this data set was used to measure speedup factor for varying image size. Data was time averaged on windows of 300 seconds, producing only 835360 visibilities. To stress multi-GPU compute capacity, a MEM image of 2048×2048 pixels was reconstructed from HL Tau full Band 6 with four spectral windows, four channels each and a total of 96399248 visibilities. The CLEAN image was downloaded from the ALMA Science Verification site and had 1600×1600 pixels.

3.2. Measures and additional processing

Additionally, we use CASA (McMullin et al., 2007) to reconstruct CLEAN images that were not downloaded from the ALMA Science verification site and also to use the `mstransform` command to time average specific spectral windows of the HL Tau Band 6 dataset.

Since MEM separates signal from noise (A. Richard Thompson, 2004; Taylor et al., 1999), CASA software was also used to join each MEM image and its residuals, as follows. Firstly, MEM’s model image was convolved with CLEAN’s beam, which effectively changes MEM image units from Jy/pixel to Jy/beam, but also changes MEM image resolution. Secondly, a CLEAN image was produced from MEM’s residuals. Finally, these two last images were added to generate what is called a MEM restored image.

Computational performance was measured by the speedup factor between one GPU and one CPU (single

Table 1: Average time per iteration (min) for one CPU and a single GPU versions of MEM, for two data sets and different images sizes.

Data	CPU time			GPU time		
	1024x1024	2048x2048	4096x4096	1024x1024	2048x2048	4096x4096
CO(6-5)	131.13	348.60	360.32	0.08	0.11	0.13
HL Tau B6w0	2645.21	10338.82	55008.72	0.60	2.39	9.86

thread) wall-clock execution time versions of MEM, for three images sizes, namely 1024×1024 , 2048×2048 , and 4096×4096 . Since the number of MEM iterations depended on the data set, and some of the sequential reconstructions took excessively long, the average execution time per iteration, was used as timing measure. All speedup results were for short-spaced channels data sets. The GPU platform consisted of a cluster of four Tesla K80, each one with two Tesla GK210B GPUs, with 2496 streaming processors (CUDA cores) and 12 Gbytes of global memory. It is important to highlight that this system had two PCI-Express ports on which two GPU were connected to each port. Also, it had a 2x10 Core Intel Xeon E5-2640 V2 2GHz processor on which sequential versions of MEM were run.

Unfortunately, speedup cannot be measured in function of GPU cores. Speed factor is usually measured according to a certain number of processing elements like threads or processors. However, GPU programmers do not have control over the number of cores used in every kernel, but only over the grid dimensions. Grid size affects the streaming multiprocessors (SM) occupancy which is defined as:

$$O = \frac{\text{Number of Active Warps}}{\text{Total Number of Warps}} \quad (11)$$

Thus, occupancy measures how efficiently the SM are being used. Any program inefficiencies like inappropriate grid and block sizes, uncoalesced memory access, thread divergence or unbalanced workload will decrease the number of active warps and therefore SM occupancy and speedup. We used the NVIDIA Profiler (NVIDIA Corporation, 2016b) to measure kernel's occupancy, floating-point operations, compute bound and memory bound kernels.

4. Results

4.1. Computational performance

When using CPU timers, it is critical to remember that all kernel launches and API functions are asynchronous, in other words, they return control back to the calling CPU thread prior completing their work. Therefore, to accurately measured the elapsed time it is necessary to synchronize the CPU thread by calling `cudaDeviceSynchronize()` which blocks the calling CPU thread until all previously issued CUDA calls by the thread are completed (NVIDIA Corporation, 2016a).

Table 2: Average speedup per iteration for two data sets and different image sizes.

Data	Speedup		
	1024x1024	2048x2048	4096x4096
CO(6-5)	1681.14	3169.12	2688.93
HLTau B6w0	4379.49	4322.25	5580.11

Table 1 shows average time per iteration in minutes, for the sequential and single GPU versions of MEM. Notice that for the HL Tau case, execution times increased proportionally with image size. Thus, going from a 1024×1024 to a 4096×4096 image, GPU execution time increased by a factor of 4, approximately. A similar behavior can be observed for the CPU case. However, the results for the CO(6-5) data sets are different and there cannot be found a proportionality between execution time and image size increase. There must be remembered that this is the smallest data set used and neither CPU nor GPU are stressed to the their maximal computational capacity. In any case, the time differences are significant. A single iteration for the 4096×4096 HL Tau sequential case took 38.2 days to finish, while the same reconstruction in GPU took only 9.85 minutes. Considering that the algorithm converged in 147 iterations in GPU, it is clear why we could not measure total execution time of the sequential version of MEM. Speedup factors for all cases are presented in Table 2. The smallest and largest speed factor achieved were 1681.14 and 5580.11, respectively. The smallest speedup corresponded to the short baseline CO(6-5) with smallest image size. CPU took 2.5 days to reconstruct, and GPU only 2.1 minutes.

Results for multi-GPU reconstruction are displayed in Figure 3. This graph shows average execution time per iteration as a function of number of GPUs, for the two largest data sets. The number of visibilities of these data sets were similar, but the number of visibilities per channel were different. Also the SDP 8.1 data had one channel with zero visibilities. Both curves display a typical fixed workload timing behavior, where execution time does not decrease linearly as more GPUs are used. Due to the fact that the number of visibilities in each channel varies, some load imbalance among GPUs may be causing this behavior. However, it is well known in parallel computing that fixed workload always has a maximum speedup possible.

Finally, Table 3 shows the number of registers per thread

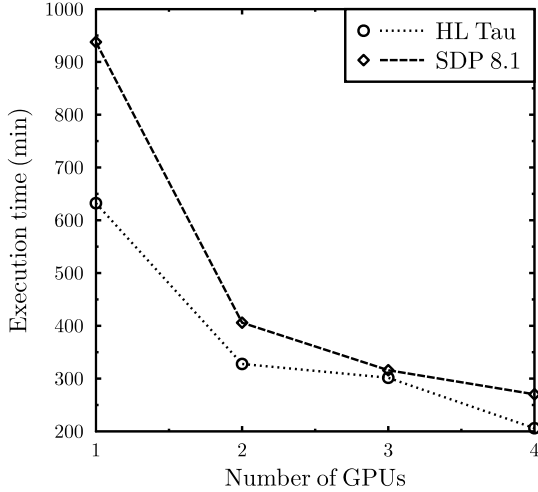


Figure 3: Average time per iteration using multi-GPU, SDP 8.1 Band 7 and HL Tau Band 6 data sets.

of the most important kernels and the achieved occupancy, for the CO(6-5) data set. The **KGradChi2** kernel is the only kernel that achieved 100% occupancy for any image size. This is probably due to the fact that it is the most compute intensive of all the kernels and it is able to maintain the GPU busy without many context-switches. Even though the other kernels require a smaller number of register per thread than the **KGradChi2**, which means they can have more active warps per SM, they do not achieve maximum occupancy. Another observation it can be made, is that occupancy varies differently for different kernels with image size. For instance, the **KGradPhi** kernel has a consistent 76% to 78% occupancy, but the **KReduce** kernel occupancy improves with image size, showing that the kernel itself is not big enough to fully exploit the GPU.

4.2. Image reconstruction

Figure 4 depicts restored MEM and CLEAN image profiles of the point source phantom. On one hand, CLEAN achieved a higher point source intensity, but on the other hand MEM performed a smoother fit of the plateau. The FWHM was 0.072 and 0.052 arcsec for MEM and CLEAN, respectively. Notice that CLEAN overestimated the point source intensity and MEM underestimated the value. CLEAN tends to fit the point source and plateau overestimating the peak, but MEM is known to lose flux in recovered images (Narayan and Nityananda, 1986).

Image reconstruction results are shown in Figure 5. First column displays MEM model images, second column shows MEM restored images, and third column shows CLEAN images. First and second rows corresponds to the SDP 8.1 on Band 7 and HL Tau on Band 6, respectively. As it has been said both data are long baseline data sets, and therefore, a high angular resolution and detail is expected. Third row shows the Antennae Galaxies North on Band 7. This dataset is a short baseline and also a mosaic. Model images (first column) show images with only signal

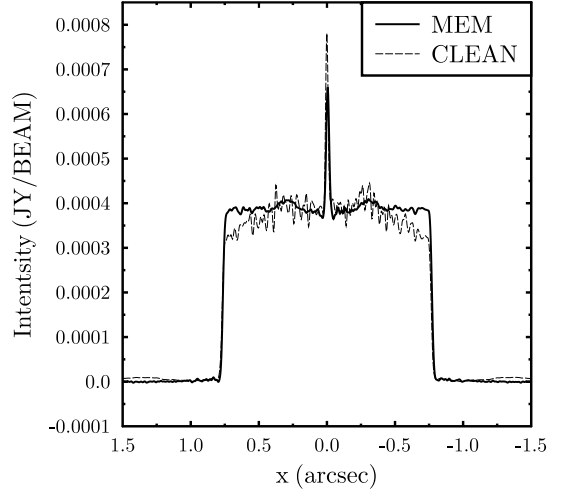


Figure 4: Reconstructed image profile of a point source in the center of a flat circular plateau, with MEM (restored), $\lambda = 0.1$ and CLEAN. The point source and plateau intensities of the simulated image are 7.4×10^{-4} (Jy/beam) and 3.8×10^{-4} (Jy/beam), respectively.

as MEM separates it from noise, and in all cases restored images are of lower resolution than model images.

5. Conclusions

We have developed a high performance computing solution for maximum entropy image reconstruction in interferometry. The solution is based on GPU for single channel data and on multiple GPUs for multi-spectral data. The implementation uses a host algorithm that runs the main iteration loop and orchestrates kernel calls. We have decided to write small kernels to improve data locality and minimize thread divergence. Most data is kept on device memory which also minimizes memory moves between host and devices. Overall, we have found that the algorithm renders naturally into a SIMT paradigm and makes it a good candidate for successful GPU implementation.

The resulting code achieves a speedup of approximately 1681 times for our smallest data set, which means that CO(6-5) can be reconstructed in 2.1 minutes, instead of 2.5 days in single CPU. Long baseline HL Tau Band 6, spectral window zero and 1024^2 image size, takes 58 minutes approximately. However for full band, multi-spectral data sets, like SDP 8.1 and HL Tau, non-gridded MEM reconstruction is still a challenging task. For instance, full data-set band 6 HL Tau reconstruction still requires 200 minutes per iteration, which is unacceptable for routine use.

The results are encouraging, but there is still several challenges that need to be overcome. Most of the algorithm's time is spent on gradient computation, which is the most demanding kernel. But, for nearly-black objects, gradient evaluation in pixels far from the central object carry little or zero contribution to the signal. Therefore, we are currently modifying our code to include a focus-of-attention improvement, which allows to select a region

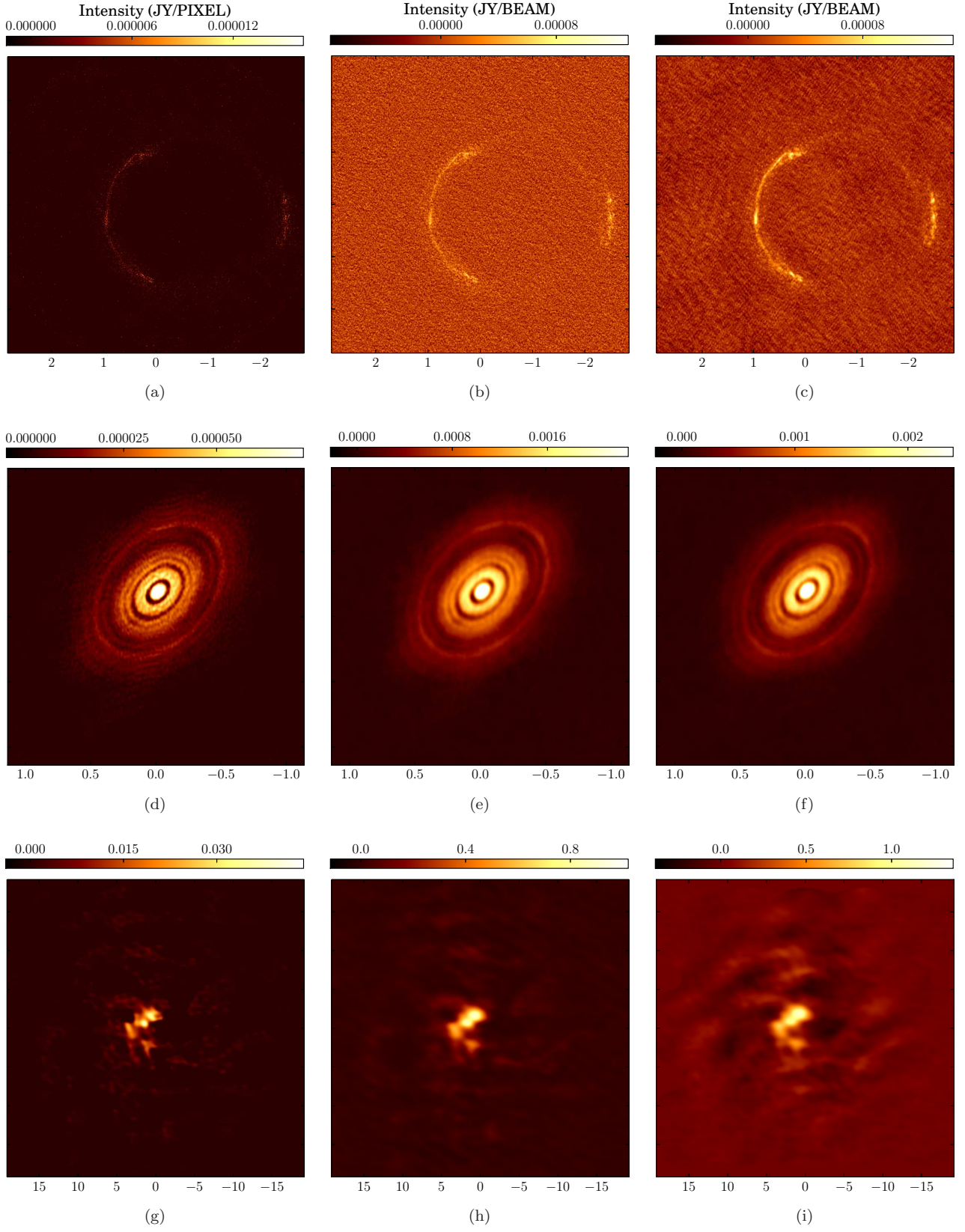


Figure 5: Image synthesis results using MEM and CLEAN. First column shows MEM model images. Second column shows MEM Restored images, and third column shows CLEAN images. (a)-(c) SDP 8.1 on Band 7, (d)-(f) HL Tau on Band 6. (g)-(i) Antennae Galaxies Northern mosaic on Band 7.

Table 3: Occupancy of the most important kernels using a 1024 thread block size and different image sizes.

Kernel	Registers per thread	Achieved Occupancy (%)		
		1024x1024	2048x2048	4096x4096
KGradPhi	10	77	76	78
KChi2	12	81	79	80
KGradChi2	30	100	100	100
KEntropy	13	86	81	79
KGradEntropy	13	87	81	80
KInterpolation	32	92	90	88
KModulation	18	89	89	88
KReduce	13	75	88	96

on the image where reconstruction should take place, thus reducing the number of free parameters substantially. We are also exploring the use of multiple GPUs for multiple fields and single channel data sets. Additionally, we are currently working on a solution to reconstruct data sets with complex polarizations and long-spaced multiple channels, also called multi-frequency synthesis.

Finally, we would like to emphasize that although this work implements the MEM algorithm using the conjugate gradient algorithm, many other first order optimization methods based on gradient can be used. Also, the design permits the use of other well behaved regularization method to penalize the solution. In consequence we demonstrate that unsupervised methods based on a fitting criteria and implemented in GPU can be used in practice for research in astronomy.

Acknowledgements

This paper makes use of the following ALMA data: ADS/JAO.ALMA #2011.0.000015.SV, ADS/JAO.ALMA #2011.0.00465.S, ADS/JAO.ALMA #2011.0.00003.SV, ADS/JAO.ALMA #2011.0.000016.SV, ADS/JAO.ALMA #2013.1.00305.S. ALMA is a partnership of ESO (representing its member states), NSF (USA) and NINS (Japan), together with NRC (Canada), NSC and ASIAA (Taiwan), and KASI (Republic of Korea), in cooperation with the Republic of Chile. The Joint ALMA Observatory is operated by ESO, AUI/NRAO and NAOJ. Also, the calculations used in this work were performed in the Belka cluster, financed by Fondecip project EQM140101 and housed at MAD/Cerro Calan. F. R. Rannou and M. Carcamo were partially funded by DICYT project 061519RF, Universidad de Santiago de Chile. P. Roman was partially funded by postdoctoral Fondecyt projects 3140634 and Basal PFB-03, Universidad de Chile.

References

- A. Richard Thompson, James M. Moran, G., 2004. Interferometry and Synthesis in Radio Astronomy. WILEY-VCH Verlag GmbH & Co. KGaA, Weinheim.
- ALMA Partnership, et al., 2015. The 2014 ALMA Long Baseline Campaign: First Results from High Angular Resolution Observations toward the HL Tau Region. *Astrophysical Journal Letters* 808, L3.
- Burger, W., Burge, M., 2010. Principles of Digital Image Processing: Core Algorithms. Undergraduate Topics in Computer Science, Springer London.
- Candan, C., Kutay, M., Ozaktas, H., 2000. The discrete fractional fourier transform. *Signal Processing, IEEE Transactions on* 48, 1329–1337.
- Casassus, S., Marino, S., Perez, S., Roman, P., Dunhill, A., Armitage, P.J., Cuadra, J., Wootten, A., van der Plas, G., Cieza, L., Moral, V., Christiaens, V., Montesinos, M., 2015a. Accretion kinematics through the warped transition disk in hd142527 from resolved co(65) observations. *The Astrophysical Journal* 811, 92.
- Casassus, S., Wright, C., Marino, S., Maddison, S.T., Wootten, A., Roman, P., Perez, S., Pinilla, P., Wyatt, M., Moral, V., Menard, F., Christiaens, V., Cieza, L., van der Plas, G., 2015b. A compact concentration of large grains in the hd142527 protoplanetary dust trap. *The Astrophysical Journal* 0, 10.
- Chen, W., 2011. The ill-posedness of the sampling problem and regularized sampling algorithm. *Digit. Signal Process.* 21, 375–390.
- Clark, B.G., 1999. Coherence in Radio Astronomy, in: Taylor, G.B., Carilli, C.L., Perley, R.A. (Eds.), *ASP Conf. Ser.* 180: Synthesis Imaging in Radio Astronomy II, pp. 1+.
- Cornwell, T.J., 1988. Radio-interferometric imaging of very large objects. *Astronomy and Astrophysics* 202, 316–321.
- Cornwell, T.J., Evans, K.F., 1985a. A simple maximum entropy deconvolution algorithm. *AAP* 143, 77–83.
- Cornwell, T.J., Evans, K.F., 1985b. A simple maximum entropy deconvolution algorithm. *Astronomy and Astrophysics* 143, 77–83.
- Coughlan, C.P., Gabuzda, D.C., 2013. Imaging VLBI polarimetry data from Active Galactic Nuclei using the Maximum Entropy Method, in: *European Physical Journal Web of Conferences*, p. 07009.
- Donoho, D.L., Jhonstone, I.M., Koch, J.C., Stern, A.S., 1992. Maximum entropy and the nearly black object. *Journal of the Royal Statistical Society* 54.
- Hogbom, J.A., 1974. Aperture Synthesis with a Non-Regular Distribution of Interferometer Baselines. *Astron. Astrophys. Suppl. Ser.* 15, 417–426.
- Lannes, A., Anterrieu, E., Marechal, P., 1997. Clean and Wipe. *Astronomy and Astrophysics* 123, 183–198.
- Lochner, M., Natarajan, I., Zwart, J.T.L., Smirnov, O., Bassett, B.A., Oozeer, N., Kunz, M., 2015. Bayesian inference for radio observations. *Monthly Notices of the Royal Astronomical Society* 450, 1308–1319. [arXiv:1501.05304](https://arxiv.org/abs/1501.05304).
- Marechal, P., Wallach, D., 2009. Fourier synthesis via partially finite convex programming. *Mathematical and Computer Modelling* 49, 2206 – 2212. *Trends in Application of Mathematics to Medicine-Trends in Application of Mathematics to Medicine*.
- McMullin, J.P., Waters, B., Schiebel, D., Young, W., Golap, K., 2007. Astronomical data analysis software and systems xvi, in: Shaw, R.A., Hill, F., Bell, D.J. (Eds.), *ASP Conf. Ser.*, San Francisco, CA. p. 127.
- Narayan, R., Nityananda, R., 1986. Maximum entropy image restoration in astronomy. *Annual Review of Astronomy and Astrophysics* 24, 127–170.
- Neff, S.G., Eilek, J.A., Owen, F.N., 2015. The complex north transition region of centaurus a: Radio structure. *The Astrophysical Journal* 802, 87.
- Nesterov, Y., 2004. *Introductory Lectures on Convex Optimization*. Springer-Verlag, New York, NY, USA.
- Nocedal, J., Wright, S.J., 2006. *Numerical Optimization*. Springer-Verlag, New York, NY, USA.
- NVIDIA Corporation, 2016a. *CUDA C Best Practices Guide*. NVIDIA Corporation. Version 8.0.
- NVIDIA Corporation, 2016b. *CUDA Profiler User's Guide*. NVIDIA Corporation. Version 8.0.
- OpenMP Architecture Review Board, 2015. *OpenMP openmp application programming interface version 4.5*. URL: <http://www.openmp.org/mp-documents/openmp-4.5.pdf>.
- Perkins, S., Marais, P., Zwart, J., Natarajan, I., Smirnov, O., 2015. Montblanc: Gpu accelerated radio interferometer measurement equations in support of bayesian inference for radio observations. *Astronomy and Computing* 12, 73–85.
- Pinte, C., Dent, W.R.F., Menard, F., Hales, A., Hill, T., Cortes, P., de Gregorio-Monsalvo, I., 2016. Dust and gas in the disk of hl tauri: Surface density, dust settling, and dust-to-gas ratio. *The Astrophysical Journal* 816, 25.
- Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., 1992. *Numerical Recipes in C (2Nd Ed.): The Art of Scientific Computing*. Cambridge University Press, New York, NY, USA.
- Quinn, P., Axelrod, T., Bird, I., Dodson, R., Szalay, A., Wicenec, A., 2015. Delivering SKA science. *CoRR abs/1501.05367*.
- Rastorgueva, E. A., Wiik, K. J., Bajkova, A. T., Valtaoja, E., Takalo, L. O., Vetukhnovskaya, Y. N., Mahmud, M., 2011. Multi-frequency vlba study of the blazar s5 0716+714 during the active state in 2004. *Astronomy and Astrophysics* 529.
- Rau, U., Cornwell, T.J., 2011. A multi-scale multi-frequency deconvolution algorithm for synthesis imaging in radio interferometry. *Astronomy and Astrophysics* 532, A71.
- Sutton, E.C., Wandelt, B.D., 2006. Optimal image reconstruction in radio interferometry. *The Astrophysical Journal Supplement Series* 162, 401.
- Tamayo, D., Triaud, A.H.M.J., Menou, K., Rein, H., 2015. Dynamical stability of imaged planetary systems in formation: Application to hl tau. *The Astrophysical Journal* 805, 100.
- Tamura, Y., Oguri, M., Iono, D., Hatsukade, B., Matsuda, Y., Hayashi, M., 2015. High-resolution ALMA observations of SDP.81. I. The innermost mass profile of the lensing elliptical galaxy probed by 30 milli-arcsecond images 67, 72.
- Taylor, G.B., Carilli, C.L., Perley, R.A. (Eds.), 1999. *Synthesis Imaging in Radio Astronomy II. volume 180 of Astronomical Society of the Pacific Conference Series*. Astronomical Society of the Pacific, San Francisco.
- van Haarlem, M.P., Wise, M.W., Gunst, A.W., et al., 2013. LOFAR: The LOW-Frequency ARray. *Astronomy & Astrophysics* 556, A2.
- Warmuth, A., Mann, G., 2013. Thermal and nonthermal hard X-ray source sizes in solar flares obtained from RHESSI observations. I. Observations and evaluation of methods. *Astronomy and Astrophysics* 552, A86.
- Winkel, B., Lenz, D., Flöer, L., 2016. Cygrid: A fast Cython-powered convolution-based gridding module for Python. *Astronomy and Astrophysics* 591, A12.