

Fast Convolutional Resampling on Parallel Architectures

Sebastiaan van der Tol

Astron, Netherlands institute for Radio Astronomy

2nd ASTERICS-OBELICS Workshop
16-19 October 2017, Barcelona, Spain.



ASTRON

H2020-Astronomy ESFRI and Research Infrastructure Cluster
(Grant Agreement number: 653477).

Introduction

This talk is about a very specific algorithm, developed for the LOFAR radio telescope.

What are the links to the other ASTERICS ESFRI projects (the common challenges)?

- ▶ Algorithm might be applicable to data processing in other observatories where one needs to
 - ▶ Resample the data
 - ▶ and apply corrections for instrument or atmosphere
- ▶ Mapping an algorithm onto parallel architectures



Astronomy ESFRI & Research Infrastructure Cluster
ASTERICS - 653477



ASTRON

LOFAR Radio Telescope

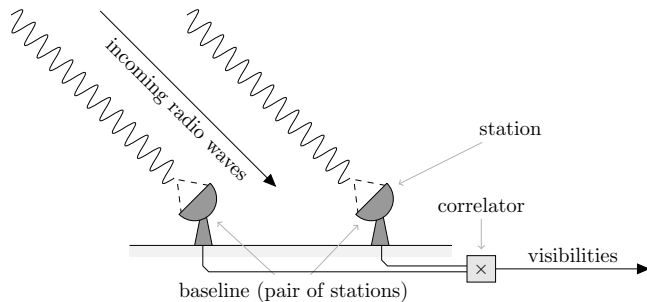


Astronomy ESFRI & Research Infrastructure Cluster
ASTERICS - 653477

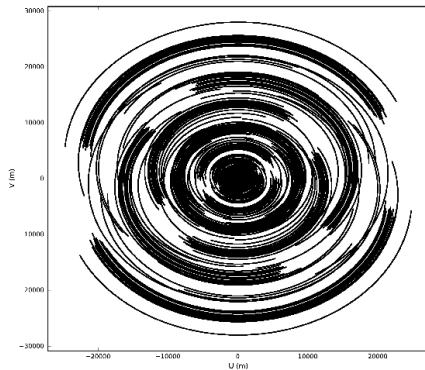


ASTRON

Radio Interferometry



Visibilities



Measurement Equation

The visibility for baseline $i - j$, channel q , timestep r is given by

$$y_{ijqr} = \iint_{lm} e^{-j2\pi(u_{ijr}l + v_{ijr}m + w_{ijr}n)/\lambda_q} g_{iqr}(l, m) g_{jqr}^*(l, m) I(l, m) dl dm$$

In discrete form

$$\mathbf{y} = \mathbf{M}\mathbf{x}$$

Direct inversion not possible, iterative methods are used.



Astronomy ESFRI & Research Infrastructure Cluster
ASTERICS - 653477



ASTRON

Iterative Imagang

Iteratively update a model image $\mathbf{x}_k$ by repeated evaluation of $\mathbf{x}_{k+1} = \mathbf{M}^H(\mathbf{y}_k - \mathbf{y})$ and $\mathbf{y}_{k+1} = \mathbf{M}\mathbf{x}_k$
For LOFAR typical sizes for the number of pixels and the number of visibilities go up to order 10^9 . Direct evaluation requires order 10^{18} operations. This can be reduced by exploiting the special form of the measurement equation. Evaluate the equations by Fast Fourier Transform



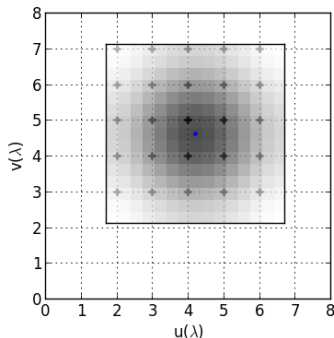
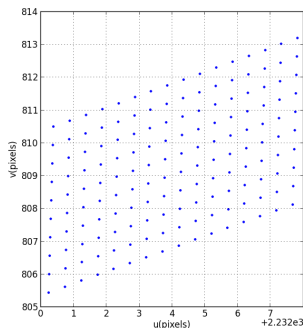
Astronomy ESFRI & Research Infrastructure Cluster
ASTERICS - 653477



ASTRON

Convolutional Resampling

To use the Fast Fourier Transform, the datapoints need to be on a regular grid



Instrumental and Atmospheric Effects

Convolution applies a multiplication (by a taper) in the image domain. Convolution kernel can be used to apply other effect that are multiplicative in the image domain, like instrumental and atmospheric effects.

In practice

- ▶ compute response on a grid (Nyquist sampled)
- ▶ zero-pad by the oversampling factor
- ▶ FFT gives the oversampled convolution kernel

Becomes very expensive for quickly varying effects (atmosphere)



Astronomy ESFRI & Research Infrastructure Cluster
ASTERICS - 653477



ASTRON

Image Domain Gridding

Idea: Instead of transforming the kernel from image to spatial frequency domain, apply the kernel directly in the image domain

- ▶ no need to compute an oversampled kernel
- ▶ while gridding, no need to do lookups in oversampled kernel
- ▶ but instead do (a lot) more computations

Memory Gap

The extra computations might affect the performance that much because computations are cheap compared to memory access

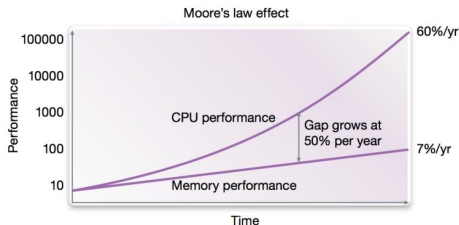


Image Domain Gridding

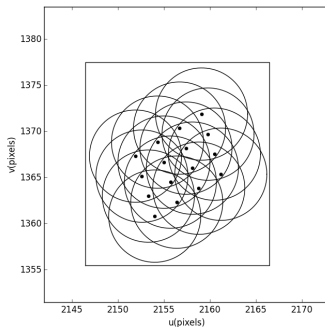
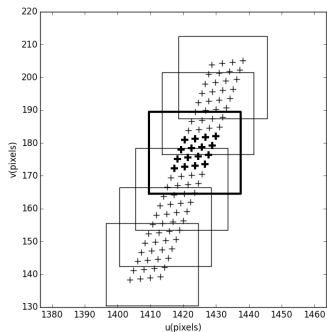


Image Domain Gridding

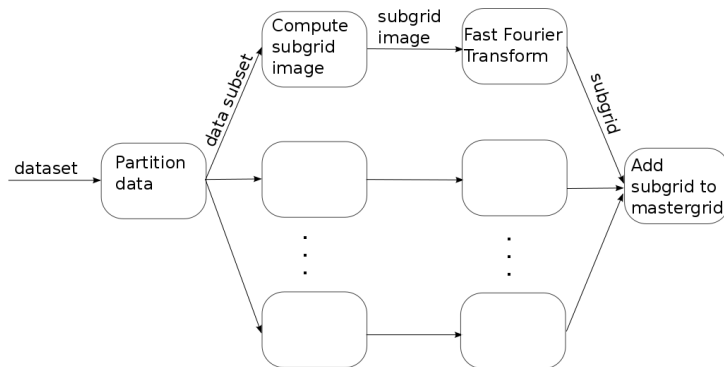


Image Domain Gridding

```
▷ iterate over data blocks
for p in 0...M-1 do
  ▷ iterate over data within block
  for k in 0...Kp-1 do
     $u \leftarrow u_{pk} - u_{0p}$ 
     $v \leftarrow v_{pk} - v_{0p}$ 
     $w \leftarrow w_{pk} - w_{0p}$ 
    ▷ iterate over pixels in subgrid
    for q in 0...Lp-1 do
      for r in 0...Lp-1 do
         $l \leftarrow -\frac{S}{2} + \frac{q}{M_p-1} \frac{S}{2}$ 
         $m \leftarrow -\frac{S}{2} + \frac{r}{M_p-1} \frac{S}{2}$ 
         $n \leftarrow \sqrt{1 - l_q^2 - m_r^2}$ 
         $\hat{I}_p[q,r] \leftarrow \hat{I}_p[q,r] +$   

                   $e^{2\pi i(ul+vm+wn)} c_{pk}^*[l,j] y_{pk}$ 
    ▷ Transform subgrid to uv domain
     $\hat{V}_p \leftarrow \text{FFT}(\hat{I}_p)$ 
```

Implementation

Implementation was done by Bram Veenboer, working for the DOME project, on

- ▶ CPU, Intel Haswell en Xeon Phi
- ▶ GPU, NVIDIA & AMD
- ▶ FPGA (currently ongoing)

Optimizations included

- ▶ Profiling
- ▶ Further partitioning of the computations
- ▶ Overlap I/O and compute

"Image-Domain Gridding on Graphics Processors", Bram Veenboer, Matthias Petschow, John W. Romein, Parallel and Distributed Processing Symposium (IPDPS), 2017 IEEE International

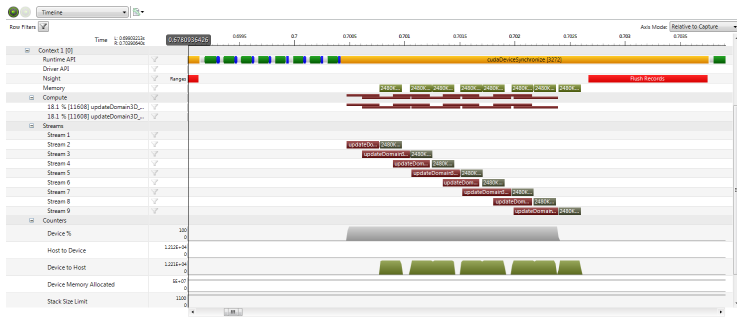


Astronomy ESFRI & Research Infrastructure Cluster
ASTERICS - 653477

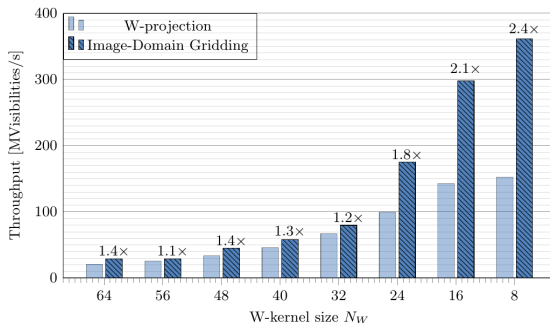


ASTRON

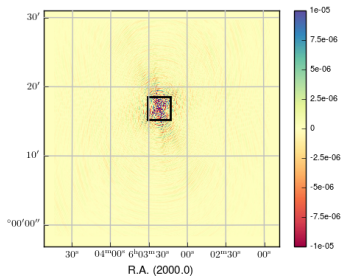
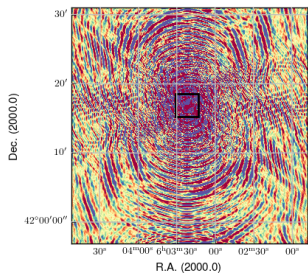
Profiling



Performance - Speed



Performance - Accuracy



Results

We wanted to eliminate the cost of kernel computation at the expense of increasing the gridding cost by hopefully not too much.

We got

- ▶ a (somewhat) higher throughput,
- ▶ eliminated cost of computation of convolution kernel
- ▶ no sampling of the kernel => more accurate



Astronomy ESFRI & Research Infrastructure Cluster
ASTERICS - 653477



ASTRON

Integration in WSClean

The algorithm has been integrated into WSClean.

- ▶ For larger images already an order of magnitude faster
- ▶ Still an order of magnitude less than raw gridding speed
- ▶ Lot of time spend in overhead, not all avoidable, but we expect significant improvement is possible

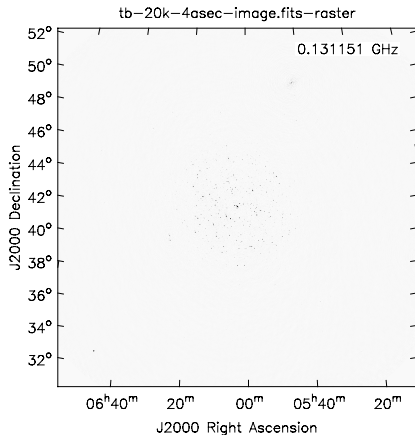


Astronomy ESFRI & Research Infrastructure Cluster
ASTERICS - 653477



ASTRON

Toothbrush field



20000 x 20000 pixels
8h observing time
10 subbands x 2 channels
4 major cycles
in 1h20m



Astronomy ESFRI & Research Infrastructure Cluster
ASTERICS - 653477



ASTRON

Repositories

- ▶ IDG <https://gitlab.com/astron-idg>
 - ▶ idg <https://gitlab.com/astron-idg/idg>
 - ▶ idg-lib <https://gitlab.com/astron-idg/idg-lib>
 - ▶ idg-bin <https://gitlab.com/astron-idg/idg-bin>
 - ▶ idg-util <https://gitlab.com/astron-idg/idg-util>
 - ▶ idg-api <https://gitlab.com/astron-idg/idg-api>
- ▶ WSClean <https://sourceforge.net/projects/wsclean/>



Astronomy ESFRI & Research Infrastructure Cluster
ASTERICS - 653477



ASTRON

Lessons learned

- ▶ Optimization starts at algorithm level
- ▶ Combine different fields of expertise (Astronomy, Signal Processing, Computer Science)
- ▶ Performance strongly influenced by match between algorithm and hardware
- ▶ Get the structure right (limit I/O, memory access) do not worry (too much) about number of computations
- ▶ Toolkit (drivers, compilers, profilers) matters. Best hardware (in theory) does not always give best performance, if the toolkit does not help you to make the best use of the hardware



Astronomy ESRI & Research Infrastructure Cluster
ASTERICS - 653477



ASTRON