Image Domain Gridding with WSClean on a CEP4 GPU node

Sebastiaan van der Tol, Bram Veenboer, André Offringa

What is Image Domain Gridding?

New gridding algorithm, but effectiv ely just an efficient implementation of A-Projection. A-Projection applies correct ions for direction dependent effects (DDEs). The effectively applied correction is a sinc interpolation o ver the sample points, in contract to the faceting approach, where a piecewise constant correction is applied.

Why a new algorithm? We would like to use A-projection but the computation of gridding kernels is expensive, especially when de DDEs vary quickly over time, for example ionospheric effects. Computing the equations in the image domain circum vents the need to compute uv domain kernels.

Properties of the algorithm

- Computationaly a bit more expensive, mostly due to sin-cos evaluations
- Highly parallel, less memory access => Well suited for GPU implementation, especially GPUs with hardware sin-cos support (NVIDIA)

Implementation

Source code can be found at https://gitlab.com/astron-idg/ (https://gitlab.com/astron-idg/)
It contains

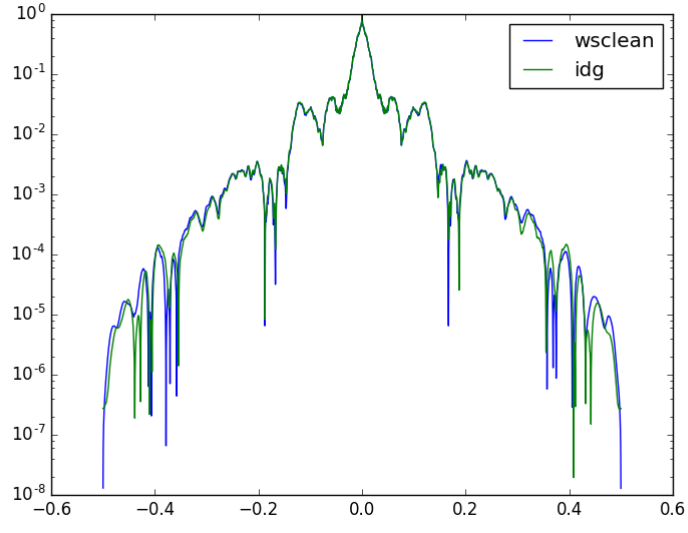
- idg-lib - core library
- idg-bin - C++ and python example code
- idg-api - higher level interface

Implementation for CPU, CUDA (NVIDIA GPU), OpenCL (other GPUs)

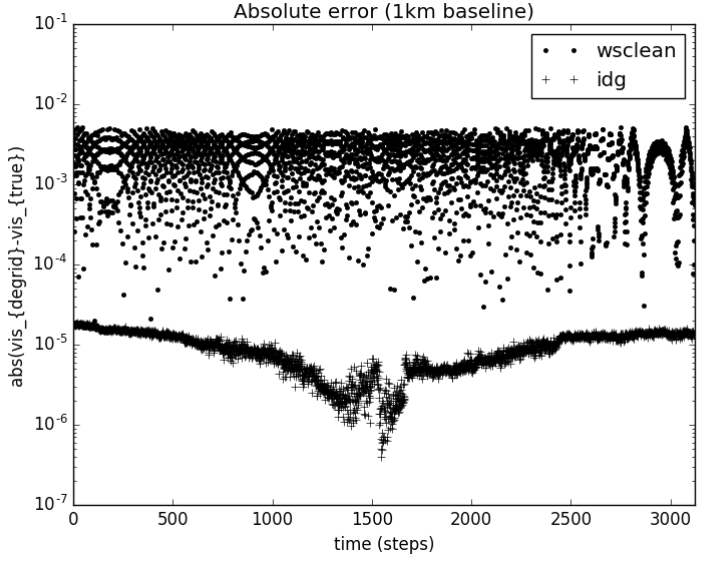
Best performance with CUDA implementation. The CPU version works best with Intel compiler, compared to GCC (6x speedup)

WSClean (https://sourceforge.net/projects/wsclean/ (https://sourceforge.net/projects/wsclean/)) can be linked against idg-api. It then provides a new option -use-idg

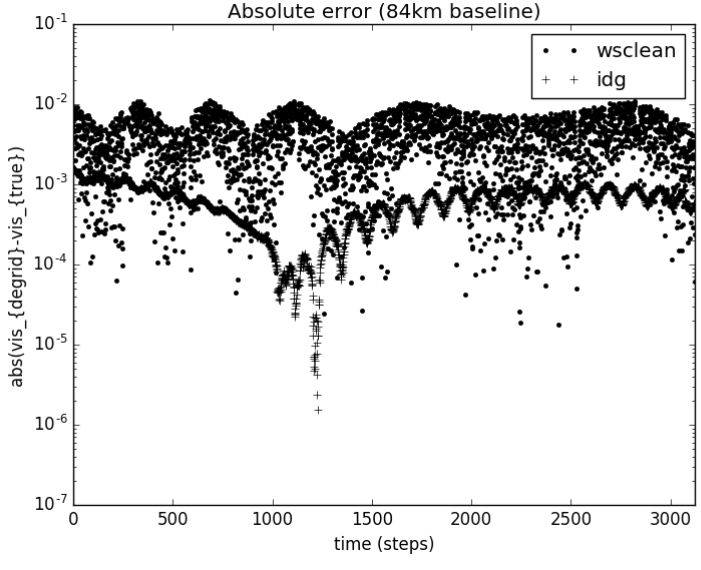
Validation PSF



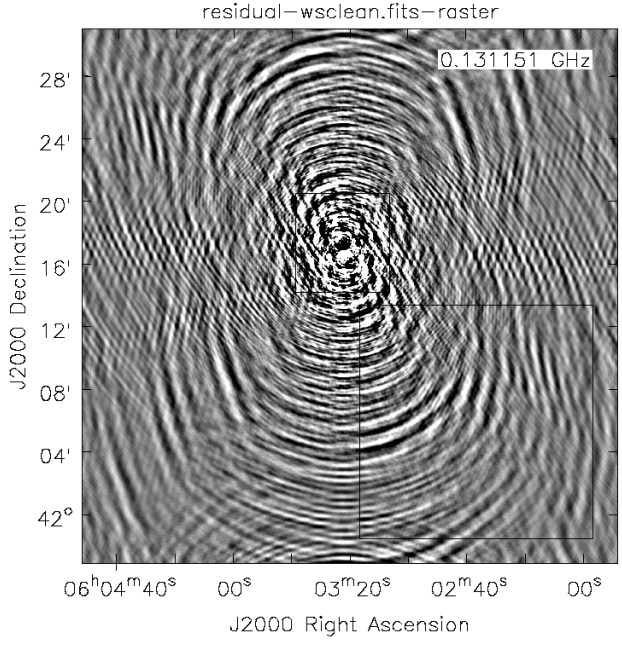
Validation - degridded visibilities



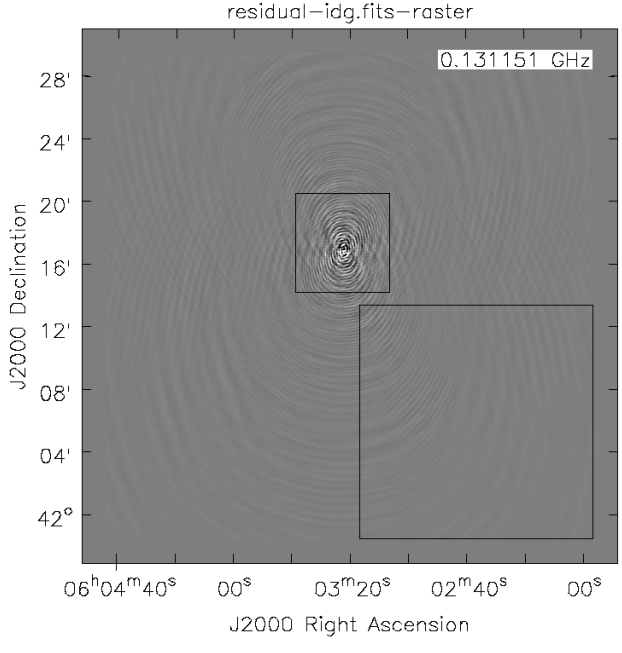
Validation - degridded visibilities



Validation - residual map wsclean



Validation - residual map idg



Performance

Raw gridding speed on a Titan X consumer NVIDIA GPU (retail price ~ €600) is > 100MVisibilities/s for a subgrid size of 32x32. Each visibility is 4 polarization x 2 complex components x 4 bytes (single precision) = 32 bytes. Data rate is 3 GB/s (~6x SSD reading speed).

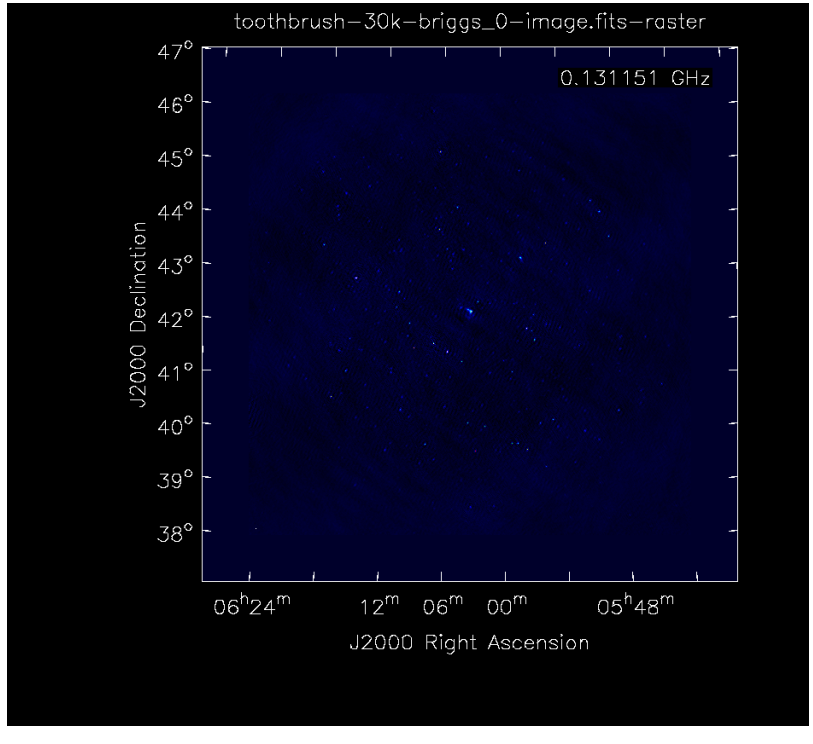
In practice

- Overhead in reading and handling the data
- For larger images W-projection needs to be combined with W-stacking
- Large grids do not fit in GPU memory, subgrids need to be added to master grid by the CPU.

Running on CEP4

CEP4 has 4 GPU nodes with 4 Tesla K40c cards per node

- 20m per major cycle for a single MS containing 20 channels 10 SB of 2 channels each



Current Status & Future

Current Status

- It is now feasible to make 30k x 30k pixel images with WSClean (on a machine with sufficient memory + GPU)
- For smaller images (< 8k x 8k) speed is about the same, for larger images (> 20k x 20k) -use-idg is a lot faster, but still far from away from the raw gridding speed
- Running over multiple MS still inefficient
- Too simplistic trade off between number of layers in W-stack and subgrid size.
- CPU adder is bottleneck in Hybrid (GPU + CPU) gridder.
- DDE correction is still filled with identity matrices

Future

- Fix performance issues
- Rerun on a set 25 MS. Goal: <1m per MS per major cy cle.
- Use results of direction dependent calibr ation