Tim Cornwell Consulting

Interferometry and Imaging

Implementation of the Rau-Cornwell MSMFS algorithm

Document number	TCC-SDP-151123-2
Context	(SDP.PIP)
Revision	2
Author	T.J. Cornwell
Release date	Wednesday 20th April, 2016 14:17
Document classification	Unrestricted
Status	Draft

Name	Designation	Affilitation
Name 1 - to be filled in	Designation 1 - to be filled in	Affiliation 1 - to be filled in
Signature & Date:		

Name	Designation	Affilitation
Name 2 - to be filled in	Designation 2 - to be filled in	Affiliation 2 - to be filled in
Signature & Date:		

Version	**Date of issue**	**Prepared by**	**Comments**

ORGANISATION DETAILS

Name	Science Data Processor Consortium

Table of Contents

List of abbreviations

CDR	Critical Design Review
CPU	Central Processing Unit
CSP	Central Signal Processor
CUFFT	Define it.
DFT	Only used once, in a title. Remove/expand?
DM	Define it.
EoR	Epoch of Reionisation
FFT	Fast Fourier Transform
FLOP	Floating Point Operations
FLOPS	Floating Point Operations per Second
FoV	Field of View
GPU	Graphics Processing Unit
ICD	Interface Control Document
LMA	Levenberg-Marquardt Algorithm
LOFAR	LOw Frequency ARray
LSM	Local Sky Model
MS-MFS	Multi-Scale Multi-Frequency Synthesis
NVRAM	Define it.
OCLD	Optimised Candidate List and Data
PDR	Preliminary Design Review
RFI	Radio Frequency Interference
PSF	Define it.
PSS	Pulsar Search
PST	Pulsar Timing
SDP	Scientific Data Processor
SKA	Square Kilometre Array
SKAO	SKA Organisation
WCS	Define it.

List of Figures

List of Tables

Summary

Applicable and reference documents

Applicable Documents

The following documents are applicable to the extent stated herein. In the event of conflict between the contents of the applicable documents and this document, *the applicable documents* shall take precedence.

Reference Number	Reference
[AD01]	Dewdney, P. E. (2013). SKA1 System Baseline Design. SKA Office
[AD02]	McCool, R., Cornwell, T. (2013). Miscellaneous Corrections to the Baseline Design
[AD03]	SKA Phase 1 System (Level 1) Requirements Specification, T.Cornwell, SKA-OFF.SE.ARC-SKO-SRS-001_2
[AD04]	PDR.01 SDP.ARCH document
[AD05]	SDP Costing spreadsheet
[AD06]	Cornwell, T.J. (2015). SKA1 Telescope Calibration Framework. SKA Office (draft version)
[AD07]	CSP–SDP ICD

Reference Documents

The following documents are referenced in this document. In the event of conflict between the contents of the referenced documents and this document, *this document* shall take precedence.

Reference Number	Reference
[RD01]	SKA-TEL-SDP-0000041, F. Malan: iPython Performance Model Description
[RD02]	SKA-TEL-SDP-0000027, R. Nijboer: Pipelines Element Subsystem Design
[RD03]	SKA-TEL-SDP-0000038, R. Bolton: High Priority Science Objectives: Performance Analysis
[RD04]	SKA-TEL-SDP-0000028, A.-J. Boonstra: Ingest Pipeline
[RD05]	SKA-TEL-SDP-0000029, S. Salvini: Pipelines: Calibration
[RD06]	SKA-TEL-SDP-0000030, A. Scaife: Imaging Pipeline
[RD07]	SKA-TEL-SDP-0000031, M. Johnston-Hollitt: Science Analysis Pipeline
[RD08]	SKA-TEL-SDP-0000017, S. Wijnholds: Baseline-Dependent Averaging
[RD09]	SKA-TEL-SDP-0000058, S. Salvini: Fast Fourier Transforms
[RD10]	SKA-TEL-SDP-0000057, C. Skipper: Time and Channel Averaging
[RD13]	`http://www.astron.nl/casacore/trunk/casacore/doc/notes/229.html`
[RD15]	Imager.cc C++ source, CASA source code SVN revision 30821, `https://svn.cv.nrao.edu/svn/casa/trunk`
[RD17]	Parameterized deconvolution for wide-band radio synthesis imaging, Urvashi Rao Venkata, 2010, PhD thesis
[RD18]	S. Bhatnagar, T. J. (2008). Correcting direction-dependent gains in the deconvolution of radio interferometric images. A&A, 419-429
[RD22]	Rik Jongerius, S. W. (2014). An End-to-End Computing Model for the Square Kilometre Array. IEEE Computer, volume 47, number 9
[RD24]	SDP Element Concept. Paul Alexander, Chris Broekema, Simon Ratcliffe, Rosie Bolton, Bojan Nikolic, 2013, SDP-PROP-DR-001-1 (part of SKA SDP Consortium proposal)
[RD34]	Cornwell, T. J. (2008). Multi-Scale CLEAN deconvolution of radio synthesis images. IEEE Journal of Selected Topics in Sig. Proc., 2, 793801.
[RD35]	Conway, J. E., Cornwell, T. J., and Wilkinson, P. N. (1990). Multi-Frequency Synthesis - a New Technique in Radio Interferometric Imaging. Monthly Notices of the Royal Astronomical Society, 246, 490.

1 Introduction

2 Purpose of the document

The purpose of this document is to describe the Multi-Scale MultiFrequency Synthesis algorithm, the variant implementations in CASA and ASKAPsoft, and the scaling for insertion in the SDP Performance Model. We conclude with some recommendations.

3 Scope of the document

4 Background

MS-MFS was developed by Urvashi [RD17], melding together the concepts from multi-scale clean [RD34] and multi-frequency synthesis e.g. [RD35], with the goal of improving the reconstruction of sky brightness from radio interferometric data. The explicit model used for the sky brightness is a collection of blobs of varying scale sizes and strengths, each with spectral behaviour described by a power law expanded into a Taylor series.

The algorithm is an elaboration of the Multiscale CLEAN. It consists of two parts:

Major cycle The image residuals for the current model are calculated by Fourier transform of the images constituting the power law expansion and degridding to obtain the predicted visibility, subtraction from the observed visibility, then gridding, and Fourier transformation to the image plane. There are two ways of performing this cycle depending on whether the Taylor series is calculated in image or uv space. Typically 5 - 10 major cycles are required.

Minor cycle The minor cycle is a greedy algorithm which identifies the dominant candidate scale and removes that using the appropriate PSF centered on that component. The minor cycle repeats this until convergence. Typically hundreds or thousands of iterations are required.

A canonical major/minor cycle algorithm is shown in Figure **??**. The two functions Predict and MakeImage are sub-pipelines. ASKAPSoft and CASA both follow this form but with different ways of performing Predict (2 and 3) and MakeImage (4 and 5)

The major cycle returns to the visibility data each iteration, whereas the minor cycle is entirely image-based.

MSMFS is relatively high in complexity because the usual CLEAN process is coupled over both scale (MSClean) and frequency or Taylor terms (MFS). The minor cycle must conduct a search in scale and Taylor term for each peak search. The major cycle requires calculation of the residual Taylor terms, which implies gridding and degridding all relevant frequency information. Memory use is high since a large number of images must be kept and updated as the iteration proceeds.

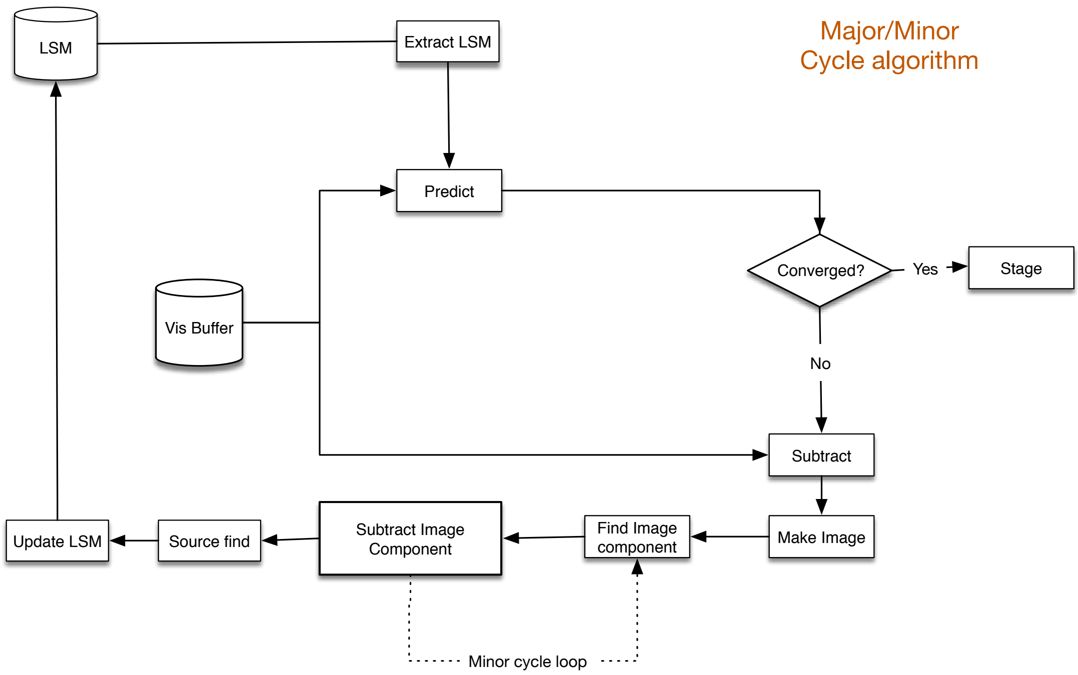

Figure 1: Structure of Major/Minor cycle algorithm

5 Overview of MSMFS

5.1 Mathematical description

The MSMFS algorithm decomposes the image into a set of blobs, using a greedy algorithm in which the peak (in scale space) is found by searching for the peak in the reference frequency residual images each convolved with scale. For bright pixels, the principal solution is calculated by applying the inverse Hessian of the PSF convolved with scale. For faint pixels, just the peak in the reference frequency image is used.

$$\vec{I}^{model} = \sum_{s=0}^{N_s-1} \vec{I}_s^{shp} \star \vec{I}_s^{sky,\delta} \tag{1}$$

where N_s is the number of spatial scales used to construct the image, and $\vec{I}_s^{sky,\delta}$ represents a collection of δ-functions that describe the locations and integrated amplitudes of flux components of scale s in the image. $\vec{I}_s^{shp}$ is a tapered truncated parabola of width proportional to s. The symbol $\star$ denotes convolution.

$$\vec{I}_\nu^{model} = \sum_{t=0}^{N_t-1} w_\nu^t \vec{I}_t^{sky} \quad \text{where} \quad w_\nu^t \quad = \quad \left(\frac{\nu - \nu_0}{\nu_0}\right)^t \tag{2}$$

where N_t is the order of the Taylor series expansion, and the I_t^m represent multi-scale Taylor coefficient images.

The image flux model at each frequency can be written as a linear sum of coefficient images at different spatial scales.

$$\vec{I}_\nu^{model} = \sum_{t=0}^{N_t} \sum_{s=0}^{N_s} w_\nu^t \left[\vec{I}_s^{shp} \star \vec{I}_{\frac{s}{t}}^{sky}\right] \quad \text{where} \quad w_\nu^t = \left(\frac{\nu - \nu_0}{\nu_0}\right)^t \tag{3}$$

Here, N_s is the number of discrete spatial scales used to represent the image and N_t is the order of the series expansion of the spectrum. $\vec{I}_{\frac{s}{t}}^{sky}$ represents a collection of δ-functions that describe the locations and integrated amplitudes of flux components of scale s in the image of the t^{th} series coefficient.

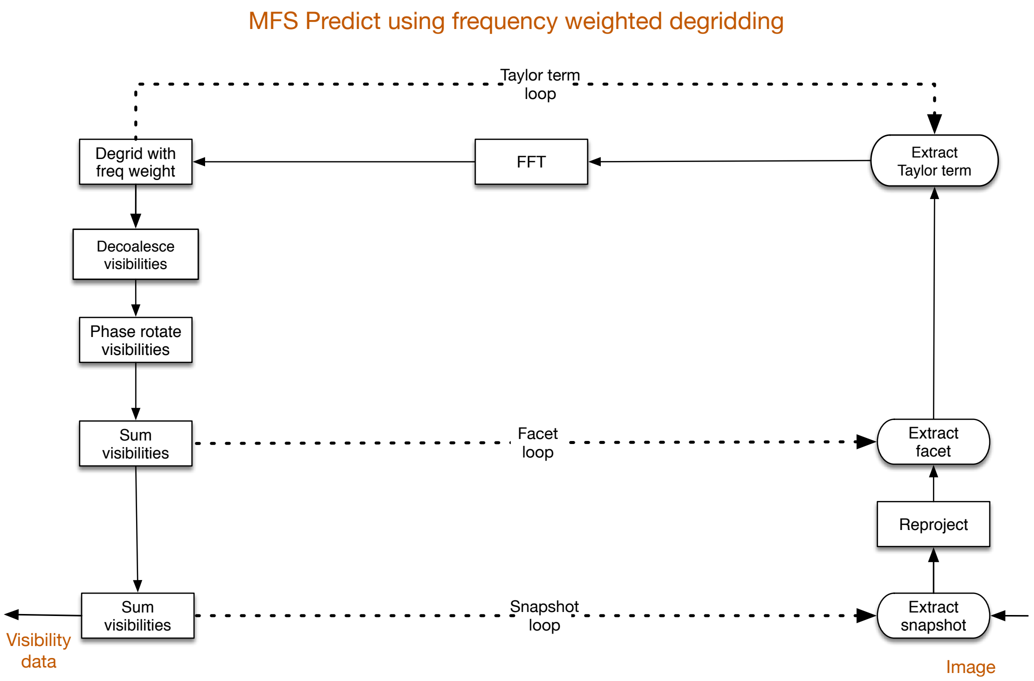

Figure 2: Structure of UV-weighting Predict, as used in ASKAPsoft.

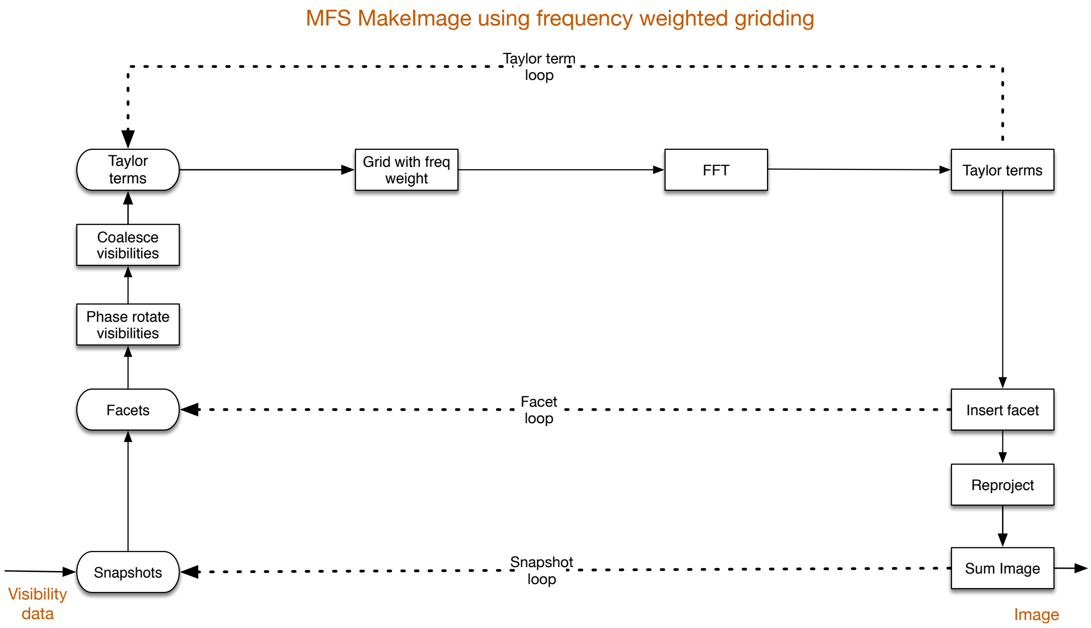

Figure 3: Structure of UV-weighting MakeImage, as used in ASKAPsoft. Note that the three loops (denoted by the rectangle with curved ends) can be permutated at will.

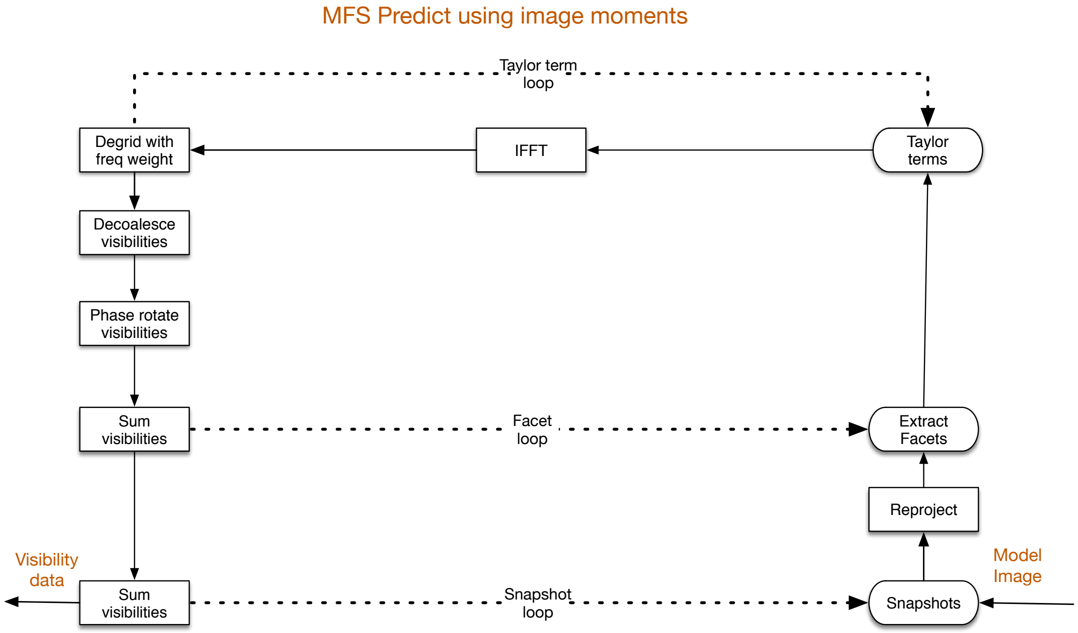

Figure 4: Structure of Image plane-weighting Predict, as used in CASA. Note that the three loops (denoted by the rectangle with curved ends) can be permutated at will.

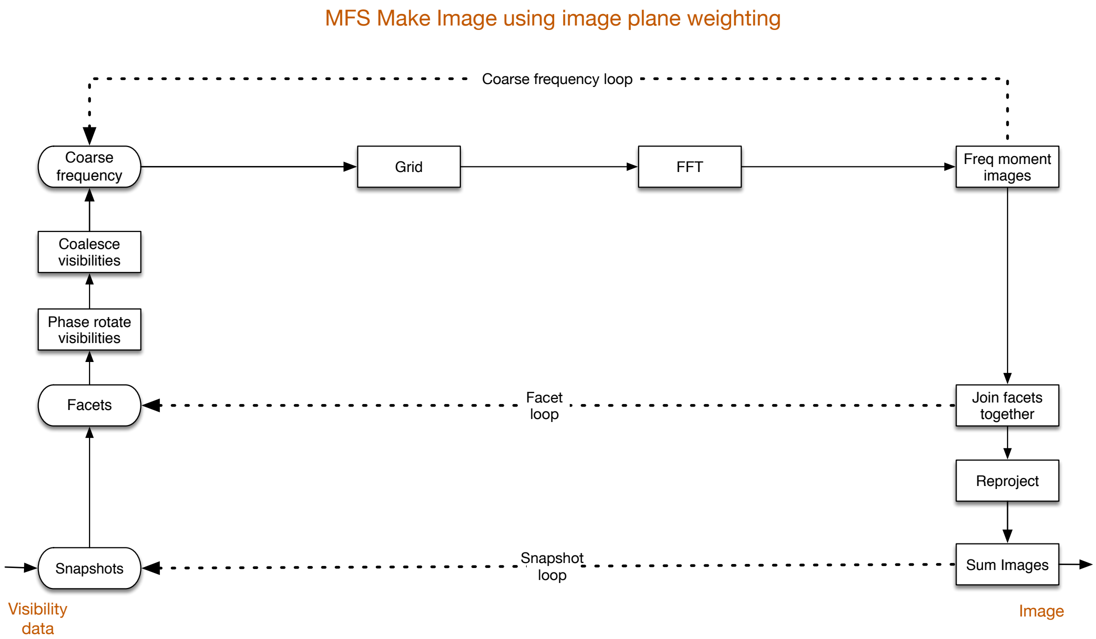

Figure 5: Structure of Image plane-weighting MakeImage, as used in CASA. Note that the three loops (denoted by the rectangle with curved ends) can be permuted at will.

6 Serial and Parallel implementations

6.1 Serial

The serial algorithm is as shown in Figure 6.

Algorithm 1: MS-MFS, as implemented in CASA.

Data: calibrated visibilities: $\boldsymbol{V}_\nu^{\rm obs}\ \ \forall\nu$
Data: uv-sampling function and weights: $[\mathsf{S}_\nu]$, $[\mathsf{W}_\nu^{\rm im}]$
Data: input: number of Taylor-terms $N_{\rm t}$, number of scales $N_{\rm s}$
Data: input: image noise threshold, $\sigma_{\rm thr}$, loop gain g
Data: input: scale basis functions: $\boldsymbol{I}_s^{\rm shp}\ \forall s \in \{0, N_{\rm s}-1\}$
Data: input: reference frequency ν_0 to compute $w_\nu = \left(\frac{\nu-\nu_0}{\nu_0}\right)$
Result: model coefficient images: $\boldsymbol{I}_t^{\rm m}\ \forall t \in \{0, N_{\rm t}-1\}$
Result: intensity, spectral index and curvature: $\boldsymbol{I}_{\nu_0}^{\rm m}, \boldsymbol{I}_\alpha^{\rm m}, \boldsymbol{I}_\beta^{\rm m}$

for $t \in \{0, N_{\rm t}-1\}, q \in \{t, N_{\rm t}-1\}$ **do**
  Compute the spectral Hessian kernel $\boldsymbol{I}_{tq}^{\rm psf} = \sum_\nu w_\nu^{t+q}\boldsymbol{I}_\nu^{\rm psf}$
  for $s \in \{0, N_{\rm s}-1\}, p \in \{s, N_{\rm s}-1\}$ **do**
    Compute scale-spectral kernels $\boldsymbol{I}_{\substack{sp\\tq}}^{\rm psf} = \boldsymbol{I}_s^{\rm shp} \star \boldsymbol{I}_p^{\rm shp} \star \boldsymbol{I}_{tq}^{\rm psf}$
  end
end
for $s \in \{0, N_{\rm s}-1\}$ **do**
  Construct $[\mathsf{H}_s^{\rm peak}]$ from $mid(\boldsymbol{I}_{\substack{s,s\\t,q}}^{\rm psf})$ and compute $[\mathsf{H}_s^{\rm peak^{-1}}]$
end
Initialize the model $\boldsymbol{I}_t^{\rm m}$ for all $t \in \{0, N_{\rm t}-1\}$

repeat `/* Major Cycle */`
  for $t \in \{0, N_{\rm t}\text{-}1\}$ **do**
    Compute $\boldsymbol{I}_t^{\rm res} = \sum_\nu w_\nu^t \boldsymbol{I}_\nu^{\rm res}$ from residual visibilities $\boldsymbol{V}_\nu^{\rm res}$
    for $s \in \{0, N_{\rm s}\text{-}1\}$ **do**
      Convolve with sth scale-function $\boldsymbol{I}_{\substack{s\\t}}^{\rm res} = \boldsymbol{I}_s^{\rm shp} \star \boldsymbol{I}_t^{\rm res}$
    end
  end
  Calculate minor-cycle threshold $f_{\rm limit}$ from $\boldsymbol{I}_{\substack{0\\0}}^{\rm res}$
  repeat `/* Minor Cycle */`
    for $s \in \{0, N_{\rm s}\text{-}1\}$ **do**
      foreach *pixel* **do**
        Construct $\boldsymbol{I}_s^{\rm rhs}$, an $N_{\rm t} \times 1$ vector from
        $\boldsymbol{I}_{\substack{s\\t}}^{\rm res}\ \ \forall\ t \in \{0, N_{\rm t}\text{-}1\}$
        Compute principal solution $\boldsymbol{I}_s^{\rm sol} = [\mathsf{H}_s^{\rm peak^{-1}}]\boldsymbol{I}_s^{\rm rhs}$
        Fill solution $\boldsymbol{I}_s^{\rm sol}$ into model images $\forall t$: $\boldsymbol{I}_{\substack{s\\t}}^{\rm m,sol}$
      end
      Choose $\boldsymbol{I}_{\substack{p\\t}}^{\rm m} = max\{\boldsymbol{I}_{\substack{s\\t=0}}^{\rm m,sol},\ \forall\ s \in \{0, N_{\rm s}\text{-}1\}\}$
    end
    for $t \in \{0, N_{\rm t}-1\}$ **do**
      Update the model image: $\boldsymbol{I}_t^{\rm m} = I_t^{\rm m} + g\,[\boldsymbol{I}_p^{\rm shp} \star \boldsymbol{I}_{\substack{p\\t}}^{\rm m}]$
      for $s \in \{0, N_{\rm s}\text{-}1\}$ **do**
        Update the residual image:
        $\boldsymbol{I}_{\substack{s\\t}}^{\rm res} = \boldsymbol{I}_{\substack{s\\t}}^{\rm res} - g\ \sum_{q=0}^{N_{\rm t}-1}[\boldsymbol{I}_{\substack{sp\\tq}}^{\rm psf} \star \boldsymbol{I}_{\substack{p\\t}}^{\rm m}]$
      end
    end
  until *Peak residual in* $\boldsymbol{I}_{\substack{0\\0}}^{\rm res} < f_{\rm limit}$
  Compute model visibilities $\boldsymbol{V}_\nu^{\rm m}$ from $\boldsymbol{I}_t^{\rm m}\ \forall t \in \{0.N_{\rm t}-1\}$
  Compute residual visibilities $\boldsymbol{V}_\nu^{\rm res} = \boldsymbol{V}_\nu^{\rm obs} - \boldsymbol{V}_\nu^{\rm m}$
until *Peak residual in* $\boldsymbol{I}_0^{\rm res} < \sigma_{\rm thr}$

Restore the model Taylor-coefficients $\boldsymbol{I}_t^{\rm m}\ \forall t \in \{0.N_{\rm t}-1\}$
Calculate $\boldsymbol{I}_{\nu_0}^{\rm m}, \boldsymbol{I}_\alpha^{\rm m}, \boldsymbol{I}_\beta^{\rm m}$ from $\boldsymbol{I}_t^{\rm m}\ \forall t \in \{0.N_{\rm t}-1\}$
If required, remove average primary beam:
$\boldsymbol{I}_{\nu_0}^{\rm new} = \boldsymbol{I}_{\nu_0}^{\rm m}/\boldsymbol{P}_{\nu_0}; \boldsymbol{I}_\alpha^{\rm new} = \boldsymbol{I}_\alpha^{\rm m} - \boldsymbol{P}_\alpha; \boldsymbol{I}_\beta^{\rm new} = \boldsymbol{I}_\beta^{\rm m} - \boldsymbol{P}_\beta$

Figure 6: The MSMFS algorithm as implemented in CASA [RD??]

6.2 Parallel processing

When moving the MSMFS algorithm to a distributed (slow interconnect) and parallel (fast interconnect) architecture such as the SDP architecture, there are a large number of important factors that must be considered.

Order of iterations

Coupling across partitions e.g. is the same pixel addressed in two different partitions? If so, is there a satisfactory approach to reconciling the two values?

Synchronisation points e.g. is a global synchronisation point required so that the deconvolutions can be made consistent? Is there an acceptable algorithm for reconciling separate facets?

Amount of CU memory

Loading of CU memory backplane

Access to visibility store

6.2.1 Minor cycle

6.2.2 Partitioning

The SDP pipeline framework [RD:??], allows partitioning across multiple axes. In this case, the natural partitions and typical values are:

Polarisation Only Stokes I can be modelled using MSMFS

Frequency O(500) coarse channels are required to avoid bandwidth smearing

Sub-bands O(1-3) partitions of the coarse channels are required to ensure that the power law approximation is adequate.

Time O(43200) O(1s) correlator samples are required to avoid smearing on the longest baselines

Facet O(100) facets to limit the size of the AW projection kernel

Scales O(5) scales are typically required to model extended structure

Taylor terms O(5) terms are required to represent the spectral behaviour

According to the logic of the MSMFS algorithm, Sub-bands can be defined as the limit of the power law approximation for brightness so consequently can always be used as the coarsest partition. For Predict and MakeImage the remaining choices are between (most rapid first):

Frequency and Taylor terms are tight coupled and so to avoid excess inter-CU communication, the order should be [Taylor, Frequency]. The remaining choices are then between the

Table 1: Possible ordering of partitions UV weighting version of Predict and MakeImage

Frequency	Time	Facet	Sub-Band
Frequency	Facet	Time	Sub-Band
Time	Facet	Frequency	Sub-Band
Time	Frequency	Facet	Sub-Band
Facet	Time	Frequency	Sub-Band
Facet	Frequency	Time	Sub-Band

two orderings of Time and Facet. Since deconvolution is non-linear, we prefer that all times be represented in a single facet rather than the other way around.

Hence according to this level of analysis, the best ordering is [Frequency, Time, Facet, Sub-Band]. Parallelisation over Compute Nodes starts from the left to the right, Distribution over Compute Islands works from the right to the left (see Figure 7):

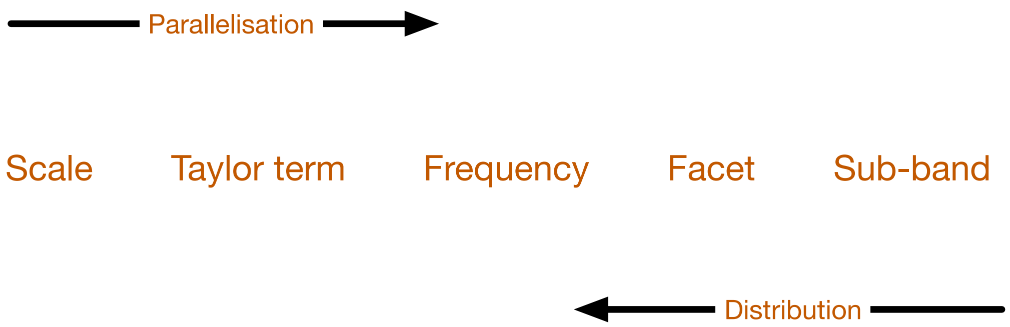

Figure 7: Preferred order of parallelisation and distribution for MSMFS algorithm

In this approach, the facets will almost certainly be processed on different compute islands. Note we have a dilemma on how to deal with the deconvolution of the facets.

- Construct the facets with some padding, deconvolve each separately, and reconcile after deconvolution by facet to facet broadcast.
- Construct the facets without padding, send all facets to one compute island, perform deconvolution, distribute models to facets.

The first approach will inevitably introduce edge effects in the final image revealing the facet grid, while the second will block processing while deconvolution proceeds on the different sub-band compute nodes. We would then require two different types of nodes:

Predict and MakeImage Requires the visibility data but not many scratch images. Stalled during MinorCycle.

MinorCycle Requires a substantial number of scratch images, but not the visibility data. Stalled during Predict and MakeImage.

6.3 Minor Cycles

6.3.1 CASA

6.3.2 ASKAPsoft

The core algorithm is in C++ Template DeconvolverMultiTermBasisFunction.tcc. The basis function is abstracted and can be any class having the interface DeconvolverMultiTermBasisFunction. There are two forms present: one for point sources, and one for the same blobs used in CASA - a truncated upside down parabola.

The expansion in Taylor terms is generalised to any form obeying an equation like: $D = B(0) * I(0) + B(1) * I(1) + B(2) * I(2)$, where D is the dirty image, B(?) are the spectral dirty beams, and I(?) are the images for each term. The coupling matrices between the terms are calculated for each basis function. Once a suitable peak component is found, the inverse of the coupling matrix is used to decouple the different terms for the optimum scale. The optimum blob location and scale is found by using one of a number of criteria:

MAXBASE0 The peak of term 0 across after decoupling in term.

MAXCHISQ The peak of chisquared across scale after decoupling in term.

The second criterion is expensive to compute and usually MAXBASE0 is used.

Once the optimum blob scale and location is found, the vector in term-space is found by decoupling. The model images (one for each term) are then updated, and the effects of this blob removed from all the cached images.

7 A projection

8 Wide-band behaviour

- Does naive broadband work?
- Bhatnagar et al WB algorithm

9 Modelling MSMFS in the SDP Performance Manager

10 Resource usage