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 . . . . . Tuesday 19th April, 2016 09:37
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 **4**

List of symbols **5**

List of figures **10**

List of tables **11**

Summary **12**

Applicable and reference documents **13**

1 Introduction **16**
- 1.1 Purpose of the document 16
- 1.2 Scope of the document 16

2 Overview of MSMFS **17**

3 Serial **23**

4 Partitioning **26**

5 A projection **27**

6 Wide-band behaviour **28**

7 Modelling MSMFS in the SDP Performance Manager **29**

8 Resource usage **30**

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 symbols

b	Length of a complex number in bytes
b	Length of a complex number in bytes
b	Length of a real number in bytes
B	Baseline length / GPU memory bandwidth
B_{max}	Maximum baseline length
B_{mem}	Memory bandwidth
B_{PCIe}	PCI-e bandwidth
C	Compression factor of visibilities for grdding for facets This symbol is not used in the document, consider removing.
$\mathbb{D}$	Matrix of observed visibilities
D_{A}	Extent of aperture
D_{s}	Diameter of antennas/stations
f_{ref}	Reference frequency
F_{cc}	Convolution kernel computational cost
F_{ci}	Factor between bandwidth and FLOPS (bytes/operations)
F_{grid}	Gridding cost
f_{patch}	Fraction of the maximum baseline below which the uv coverage is almost filled
F_{pr}	Cost for phase rotation
$\mathbb{G}$	2×2 block diagonal matrix of complex gains
$\mathbb{M}$	Matrix of model sky visibilities
$M_{\text{buf,skymod}}$	Size of the sky model buffer
$M_{\text{buf,vis}}$	Size of the visibility buffer
$M_{\text{buf,vis}}^{\text{spec}}$	Size of the visibility buffer needed for spectral line processing
$M_{\text{buf,vis}}^{\text{cont}}$	Size of the visibility buffer needed for continuum imaging
$M_{\text{cu,buf}}$	Size of buffer
$M_{\text{cu,pool}}$	Size of slow working memory
$M_{\text{cu,work}}$	Size of working memory of compute unit
M_{image}	Size of image (bytes)
$M_{\text{image,ax}}$	Image size along one axis
M_{target}	Target grid size (per facet)
$M_{uv,\text{grid}}$	Size of the uv grid (bytes)
M_{vis}	Size of visibility data (bytes)
$M_{\text{weight\_grid}}$	Size of weight grid (bytes)
$M_{w,\text{cache}}$	Cache size of gridding kernel

N_{bin}	Number of bins
N_{a}	Number of antennas or stations in the array
N_{A}	Number of A-team sources
N_{AA}	Number of grid points for A-kernel support (along one axis)
N_{apv}	Number of accesses per visibility
N_{avg}	Number of averages
N_{beam}	Number of beams simultaneously observed by the array
N_{bit}	Number of bits
N_{rma}	Number of flops in real multiply and add
N_{cma}	Number of flops in complex multiply and add
N_{bl}	Number of baselines
N_{bpa}	Number of bytes per access
N_{byte}	Number of bytes
$N_{\mathrm{byte,pix}}$	Number of bytes per pixel
$N_{\mathrm{byte,vis}}$	Number of bytes per individual visibility
$N_{\mathrm{ch}} XXXWANTTOUSENFHERE$	Number of channels
N_{cu}	Number of compute units
N_{datait}	Number of data items required per source (e.g. position)
N_{cvff}	Number of grid points along one axis for the gridding convolution function
N_{f}	Number of frequency channels/sub-bands
$N_{\mathrm{f,corr}}$	Number of frequency channels output by the correlator
$N_{\mathrm{f,grid}}$	Number of frequency channels to be gridded
$N_{\mathrm{f,de-grid}}$	Number of frequency channels to be de-gridded
$N_{\mathrm{f,out}}$	Number of frequency channels to be FFT-ed and reprojected (output) or output from the CSP pulsar timing backend
$N_{\mathrm{f,distribute}}$	Minimum number of frequency channels to maintain distributability in the SDP
$N_{\mathrm{f,kernel}}$	Number of convolution kernels needed to cover frequency axis for a given baseline
N_{facet}	Number of facets along one axis
N_{FLOP}	Number of floating point operations
N_{FLOPsamp}	Number of floating point operations per sample
N_{FLOPvis}	Number of floating point operations per visibility
N_{grid}	Number of pixels in the grid (i.e. the square of the linear dimension)
N_{GW}	Number of grid points for w-kernel support (along one axis)
N_{it}	Number of iterations

N_{kernel}	Linear size of combined A and w-kernels
N_{major}	Number of major CLEAN cycles (within each self-cal loop if appropriate)
N_{SelfCal}	Number of self-calibration cycles
N_{minor}	Number of minor CLEAN cycles (within each major cycle loop if appropriate)
N_{mm}	Factor to account for gridding of off-diagonal Müller terms
N_{ops}	Number of operations
$N_{\mathrm{patch,pix}}$	Number of pixels of PSF to be used in minor cycle decorrelation
N_{pix}	Number of pixels along one axis of a grid
$N_{\mathrm{pix,facet}}$	Number of pixels on one side of a facet
N_{pol}	Number of polarisation products
N_{source}	Number of sources
N_t	Number of time slots
N_{Tt}	Number of Taylor terms (in MS-MFS)
$N_{t,\mathrm{av}}$	Number of time slots that are averaged (in de-mixing)
N_{vis}	Number of visibilities in an observation
Q_{bw}	Quality factor
Q_{fcv}	Factor to allow for the reuse of functions between neighbouring frequency channels
Q_{FoV}	Quality factor defining how much bigger the diameter of the field of view is than the first zero of the Airy function
Q_{GCF}	Gridding oversampling factor
Q_{MSMF}	Factor to account for multiple subtractions (multi-frequency multi-scale CLEAN
Q_{pix}	Quality factor defining the oversampling of the synthesised beam
Q_t	Quality factor
Q_{kernel}	Convolution kernel quality factor, for kernel re-use in frequency
Q_w	Correction factor for w values (0 to 1)
R_{Earth}	Radius of the Earth (6400×10^3 m)
$\mathbb{R}$	Covariance matrix
$\mathbb{R}_{\mathrm{filt}}$	Filtered covariance matrix
R	Compute rate (FLOPS)
R^{spec}	Compute rate (FLOPS) for spectral line processing
R^{cont}	Compute rate (FLOPS) for continuum imaging
R^{fast}	Compute rate (FLOPS) for fast imaging
$R_{\mathrm{bw,mem}}$	Bandwidth to working memory
$R_{\mathrm{bw,max}}$	Maximum memory bandwidth of compute unit
R_{peak}	Peak FLOPS of compute unit

$R_{\mathrm{bw,I/O,max}}$	Maximum I/O bandwidth of compute unit to buffer
$R_{\mathrm{bw,I/O}}$	Bandwidth to buffer
$R^{\mathrm{cont}}_{\mathrm{bw,I/O}}$	Bandwidth to buffer for continuum imaging
$R^{\mathrm{spec}}_{\mathrm{bw,I/O}}$	Bandwidth to buffer for spectral line processing
R_{CCF}	Convolution kernel compute rate (FLOPS)
R_{FFT}	FFT compute rate (FLOPS)
R_{grid}	Gridding compute rate (FLOPS)
R_{PR}	Phase rotation compute rate (FLOPS)
R_{RP}	Re-projection compute rate (FLOPS)
R_{s}	Correlator output rate TRY TO REMOVE
r_{facet}	ratio for linear size increase of facets to account for overlap, typically 1.2-1.5
S	Telescope sensitivity TRY TO REMOVE
t_{dump}	Correlator dump time (sec)
t_{comp}	Computation time
$t_{\mathrm{kernel,update}}$	Update timescale for gridding kernel
t_{LSM}	Update timescale for the LSM
t_{obs}	Observation time
t_{snap}	Snapshot duration
$t_{\mathrm{snap,min}}$	Minimum duration of a snapshot in which w-correction is not dominated by the Earth's curvature
$t_{\mathrm{snap,opt}}$	Optimum snapshot duration (sec)
t_{subint}	Sub-integration time
u, v, w	Visibility coordinates
w	Visibility w coordinate
Δw_{max}	Maximum deviation of the w coordinate due to the Earth's curvature
Δw_{min}	Minimum deviation of the w coordinate due to the Earth's curvature
$\mathbb{W}_{\mathrm{filt}}$	Spatial filter matrix
Δf	Frequency resolution
t_{res}	Time resolution (sec)
t_{smear}	Smearing Time resolution (sec)
ϵ_{ω}	Fraction of maximal amplitude of envelope of convolution function
η_{comp}	Computational efficiency
η_f	Fraction of frequency band
θ_{bs}	Beam squint
θ_{FoV}	Angular diameter of the field of view (rad)
θ_{pix}	Angular resolution of pixel size (rad)

$\theta_{\rm PSF}$	Angular diameter of the synthesised beam (rad)
λ	Wavelength
$\lambda_{\rm max}$	Maximum wavelength
$\lambda_{\rm min}$	Minimum wavelength
$\lambda_{\rm sub,max}$	Maximum wavelength of an imaging sub-band (channels over which image and pixel size are matched)
$\lambda_{\rm sub,min}$	Minimum wavelength of an imaging sub-band (channels over which image and pixel size are matched)
$\Omega_{\rm E}$	Angular velocity of the Earth (7.27×10^{-5} rad/sec)
ϵ_f	Fraction of a uv cell that we allow uv track to move in time- or frequency-smearing-limited averaging

List of Figures

1 Structure of Major/Minor cycle algorithm 18
2 Structure of UV-weighting Predict, as used in ASKAPsoft. 19
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. 20
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. 21
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. 22

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
[RD11]	Alexander et al., Software and Computing CoDR, Analysis of requirements derived from the DRM, WP2-050.020.010-RR-001
[RD12]	SKA-TEL-SDP-0000016, R. Bolton: SDP Archive Size Estimates
[RD13]	`http://www.astron.nl/casacore/trunk/casacore/doc/notes/229.html`
[RD14]	`https://confluence.ska-sdp.org/pages/viewpage.action?title=MS+in+UML&spaceKey=ARCH`
[RD15]	Imager.cc C++ source, CASA source code SVN revision 30821, `https://svn.cv.nrao.edu/svn/casa/trunk`
[RD16]	`http://www.lofar.org/operations/lib/exe/fetch.php?media=:public:documents:casa_image_for_lofar_0.03.00.pdf`
[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
[RD19]	Cornwell, T. J., Voronkov, M. A., Humphreys, B. (2012). Wide field imaging for the Square Kilometre Array.
[RD20]	An Efficient Work-Distribution Strategy for Gridding Radio-Telescope Data on GPUs, John W. Romein, ICS12, June 2529, 2012, San Servolo Island, Venice, Italy. Source code available at: `ftp://ftp.astron.nl/outgoing/romein/Gridding-0.2.tar.bz`
[RD21]	Mitchell, D. a. (2014). Analysis of w-projection kernel size. SKA SDP Consortium
[RD22]	Rik Jongerius, S. W. (2014). An End-to-End Computing Model for the Square Kilometre Array. IEEE Computer, volume 47, number 9
[RD23]	A parametric model for the calibration and imaging costs of SKA1, T.J. Cornwell, 12 Feb 2013, SKA Draft Memo
[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)

Reference Number
[RD25]
[RD26]
[RD27]

1 Introduction

1.1 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.

1.2 Scope of the document

2 Overview of MSMFS

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.

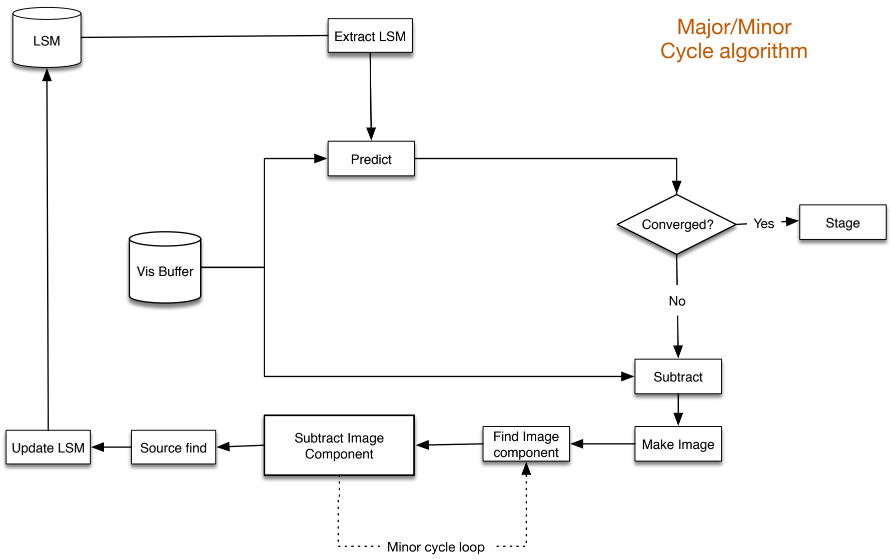

Figure 1: Structure of Major/Minor cycle algorithm

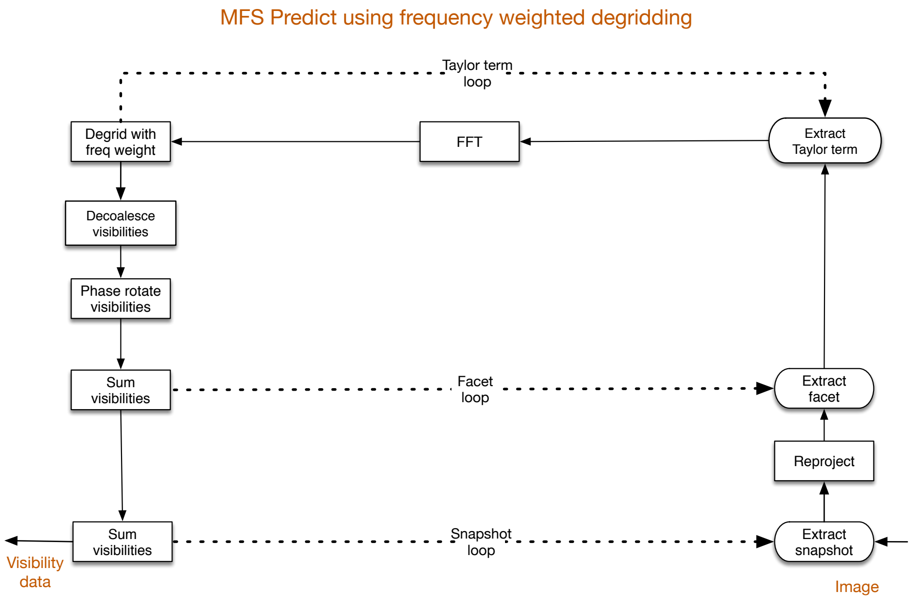

Figure 2: Structure of UV-weighting Predict, as used in ASKAPsoft.

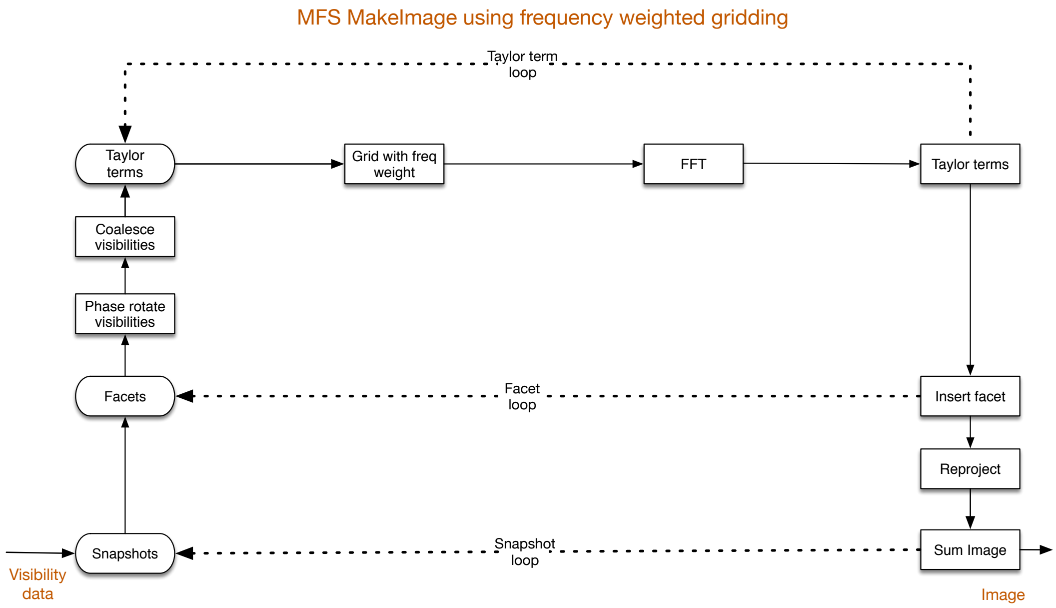

Figure 3: Structure of UV-weighting MakeImage, as used in ASKAPsoft. Note that the three loops (denoted by the rectangle with curved ends) can be permutated at will.

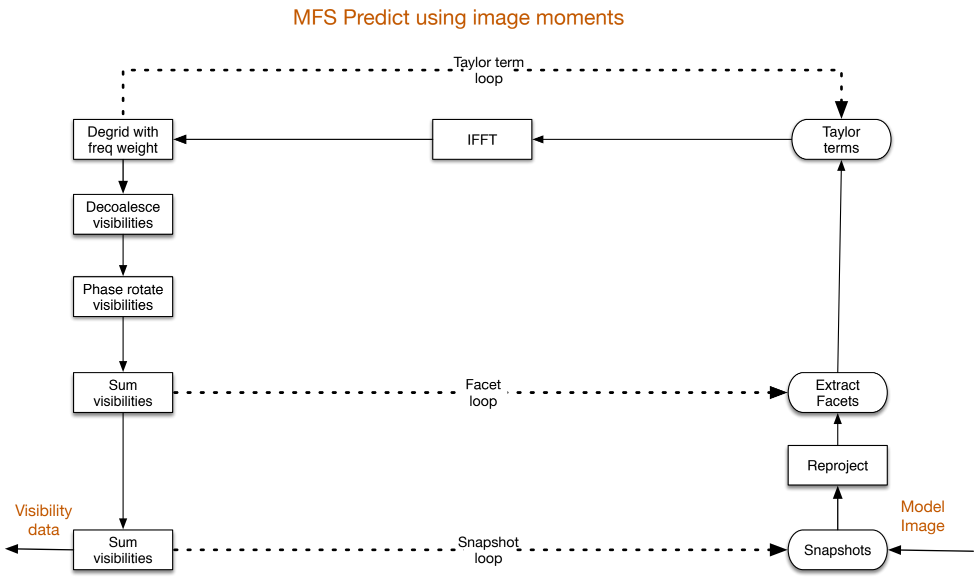

Figure 4: Structure of Image plane-weighting Predict, as used in CASA. Note that the three loops (denoted by the rectangle with curved ends) can be permutated at will.

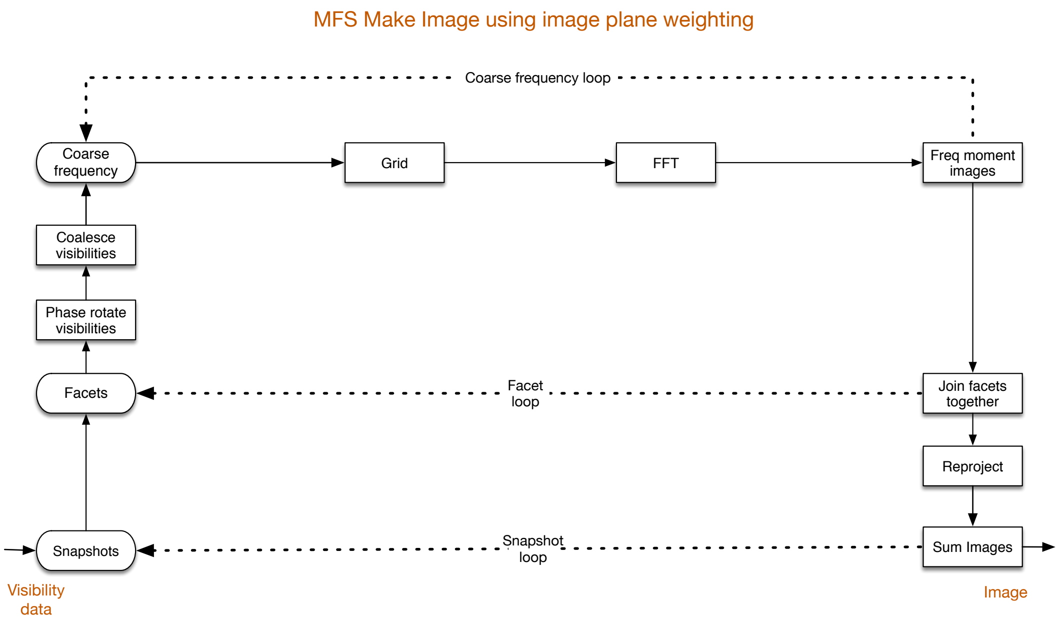

Figure 5: Structure of Image plane-weighting MakeImage, as used in CASA. Note that the three loops (denoted by the rectangle with curved ends) can be permuted at will.

3 Serial

The serial algorithm is as shown in Algorithm **??**.

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

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.

In words, 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.

Notes:

Line 13

Algorithm 1: MS-MFS Algorithm :

Data: calibrated visibilities : $\vec{V}_{\nu}^{corr}\ \ \forall\nu$
Data: uv-sampling function : S_{ν}
Data: image noise threshold and loop gain σ_{thr}, g_s
Data: scale basis functions : $\vec{I}_s^{shp}\ \forall s \in \{0, N_s - 1\}$
Result: model coefficient images : $\vec{I}_q^m\ \forall q \in \{0, N_t - 1\}$
Result: spectral index and curvature : $\vec{I}_{\alpha}^m, \vec{I}_{\beta}^m$

for $t \in \{0, N_t - 1\}, q \in \{t, N_t - 1\}$ **do**
  Compute the spectral PSF $\vec{I}_{tq}^{psf}$;
  for $s \in \{0, N_s - 1\}, p \in \{s, N_s - 1\}$ **do**
    Compute the scale-spectral PSF $\vec{I}_{\substack{sp\\tq}}^{psf} = \vec{I}_s^{shp} \star \vec{I}_p^{shp} \star \vec{I}_{tq}^{psf}$;
  end
end
for $s \in \{0, N_s - 1\}$ **do**
  Construct $[H_s^{peak}]$ from $mid(I_{\substack{s,s\\t,q}}^{psf})$ and compute $[H_s^{peak^{-1}}]$;
end
Initialize the model $\vec{I}_t^m$ for all $t \in \{0, N_t - 1\}$ and compute $f_{sidelobe}$;
repeat `/* Major Cycle */`
  for $t \in \{0, N_t\text{-}1\}$ **do**
    Compute the residual image $\vec{I}_t^{res}$; `/* Makeimage */`
    for $s \in \{0, N_s\text{-}1\}$ **do**
      Compute $\vec{I}_{s,t}^{res} = \vec{I}_s^{shp} \star \vec{I}_t^{res}$
    end
  end
  Calculate f_{limit} from $\vec{I}_{0,0}^{res}$;
  repeat `/* Minor Cycle */`
    for $s \in \{0, N_s\text{-}1\}$ **do**
      if *Peak of* $\vec{I}_{s,0}^{res} > 10\ \sigma_{thr}$ **then**
        foreach *pixel* **do**
          Construct I_s^{rhs}, an $N_t \times 1$ vector from $I_{s,t}^{res}\ \ \forall\ t \in \{0, N_t\text{-}1\}$;
          Compute principal solution $I_s^{sol} = [H_s^{peak^{-1}}] I_s^{rhs}$;
        end
        Choose $I^{sol} = max\{I_{t=0}^{sol},\ \forall\ s \in \{0, N_s\text{-}1\}\}$;
      else
        Find the location of the peak in $\vec{I}_{s,0}^{res},\ \forall\ s \in \{0, N_s\text{-}1\}$;
        Construct I_s^{rhs}, from $I_{s,t}^{res}$ for the chosen s, at this location;
        Compute $I^{sol} = [H_s^{peak^{-1}}] I_s^{rhs}$ at this location;
      end
    end
    for $t \in \{0, N_t - 1\}$ **do**
      Update the model image : $I_t^m = I_t^m + g_s\ I_{s_i}^{shp} \star I_t^{sol}$;
      for $s \in \{0, N_s\text{-}1\}$ **do**
        Update the residual image : $I_{s,t}^{res} = I_{s,t}^{res} - g\ \sum_{p=0}^{N_s-1} \sum_{q=0}^{N_t-1} [I_{\substack{sp\\tq}}^{psf} \star I_q^{sol}]$;
      end
    end
  until *Peak residual in* $\vec{I}_{0,0}^{res} < f_{limit}$;
  Compute model visibilities V_{ν}^m from $I_t^m\ \forall t \in \{0.N_t - 1\}$; `/* Predict */`
  Compute a new residual image I^{res} from residual visibilities $V_{\nu}^{corr} - V_{\nu}^m$; `/* Makeimage */`
until *Peak residual in* $\vec{I}_0^{res} < \sigma_{thr}$;
Calculate $\vec{I}_{\nu_0}^m, \vec{I}^{\alpha}, \vec{I}^{\beta}$ from $I_t^m\ \forall t \in \{0.N_t - 1\}$ and restore the results ;
If required, remove average primary beam : $\vec{I}_{\nu_0}^{new} = \vec{I}_{\nu_0}^m / \vec{P}_{b\nu_0}, \vec{I}_{\alpha}^{new} = \vec{I}_{\alpha}^m - \vec{P}_{b\alpha}, \vec{I}_{\beta}^{new} = \vec{I}_{\beta}^m - \vec{P}_{b\beta}$;

4 Partitioning

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

Frequency

Time

Facet

Scales

5 A projection

6 Wide-band behaviour

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

7 Modelling MSMFS in the SDP Performance Manager

8 Resource usage