Author: R.J. Nijboer, J.E. Noordam	Date of issue: 2005-Jul-08 Kind of issue: Public	Scope: Project Documentation Doc.nr.: LOFAR-ASTRON-DOC-???
	Status: Draft Revision nr.: 0.1	File: ftp://rnijboer.astron.nl/???

Implementing the Local Sky Model in the MeqTree System

R.J. Nijboer, J.E. Noordam

Verified:			
Name	Signature	Date	Rev.nr.
J.E. Noordam	o.p.v.	2005-???-??	0.1
J.E. Noordam	....	....	

Accepted:		
Work Package Manager	System Engineering Manager	Program Manager
J.E. Noordam .... *date:*	C.M. de Vos .... *date:*	J. Reitsma .... *date:*

Distribution list:

Group:	For Information:
ASTRON R. Assendorp M. Mevius J.E. Noordam T.A. Oosterloo O.M. Smirnov NRC A.G. Willis	ASTRON K. van der Schaaf C.M. de Vos

Document revision:

Revision	Date	Section	Page(s)	Modification
0.1	2005-Jul-08	-	-	Creation

Abstract

This document describes the implementation of the Local Sky Model (LSM) in the MeqTree system. A functional description of the Local Sky Model is given in [1]. The Introduction and section 2 of this document are based on the functional description and are meant as a short overview. For more details we refer to the the original document [1]. The actual implementation of the LSM is discussed in sections 3 and 4.

Contents

1 Introduction

The Local Sky Model (LSM) is a temporary object that is used in the Self-Calibration process. It is used for updating the Sky Model and as such it stands between the Global Sky Model (GSM) and the measured data. Since it stands between the GSM and the data the LSM combines both 'source information' and 'observational information'. Having said this it is good to realize that the LSM 'lives' only in the *image domain*. Hence, it can not contain or produce any uv-data. The uv-data is generated from the LSM image data in the User Forest (the User Forest being a scripting layer that implements the Self-Cal strategy).

The LSM is for an important part a collection of sources that is relevant for the observation of a certain part of the sky (this part of the sky is determined by the *observational window* or ObsWin). As such the LSM can be considered to be a subset of the Global Sky Model (GSM). This subset contains all sources for a given region of the sky, within a given brightness range, etc. Additionally, it will contain very bright sources that will always be visible through the sidelobes, e.g. Jupiter, Cas. A, Cyg. A, etc.

All sources will be part of one of three Categories: Cat. I, Cat. II, and Cat. III. Cat. I sources are the bright sources that are used for calibration purposes. It should be possible to predict *uv*-data directly from their parameters and solve for these parameters (in the uv-domain). Cat. II sources are all the rest of the known sources. There are simply too many of them to treat them individually. Therefore, they are treated in patches. Cat. III sources are sources that are not known yet, but that are found during the processing of the data.

The LSM is needed for the processing of observed data. There are two kinds of data processing that are relevant to the LSM: Predicting / Solving of Cat. I sources and Updating of Cat. I remnants, Cat. II and Cat. III sources. These kinds of data processing are quite different and call for different LSM functionality. After the 'Predicting' / 'Solving' step and before the 'Updating' step there is the step of 'Subtracting' the LSM from the observed uv-data in order to create residual data. This step is done within the User Forest and does not need any additional LSM functionality.

The LSM has three interfaces (see figure 1):

1. In the direction of uv-data. Although the LSM does not operate in the uv-domain, it contains SubTrees for the prediction of StokesI/Q/U/V and RA and Dec for every relevant source. These SubTrees are used in the UserForest to predict uv-data. For the Cat. I sources the same SubTrees are used to solve for source parameters.
2. In the direction of images. It should be able to analyse Residual Images and update source parameters from them. This will probably also be done with SubTrees.
3. In the direction of the GSM. The LSM should be able to get a subset out of the GSM and after the data processing it should be able to update the GSM.

As stated the LSM combines both 'source information' and 'observational information'. This is reflected

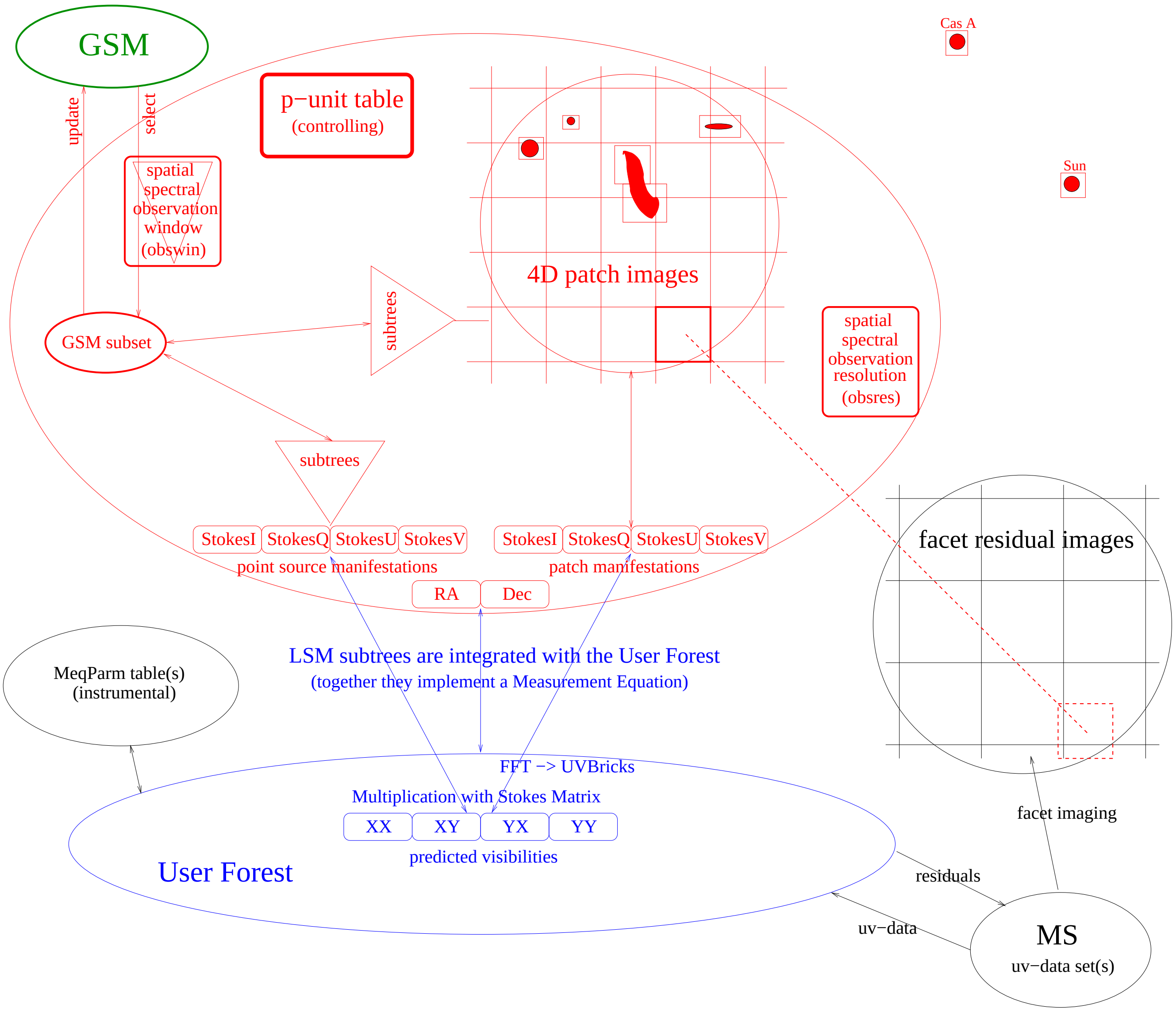


Figure 1: The Local Sky Model

in the objects it contains:

- A subset of the GSM containig information of all relevant sources in the sky.
- An Observational Window, which is an idealized Station Power Beam. It determines which part of the sky is observed.
- An Observational Resolution, both spatial and spectral.
- A set of Source Patches. A Source Patch is an image of an extended source or of multiple (point) sources. A Source Patch is treated as a single Source element in the subsequent data processing steps.
- The P-Unit List. A P-Unit is a Peeling or Predict Unit. This list controls the interfaces between the LSM and the User Forest.

2 Using the Local Sky Model

In order to get some idea how the Local Sky Model may be used, we give a description of some general procedure. Of course, with more experience in using the LSM this procedure will be altered and / or extended. However, for the moment we could think of the following.

1. Select a measured uv-data set.
2. Create the LSM Observational Window (ObsWin).
3. Create the LSM Observational Resolution (ObsRes).
4. Select a GSM subset using the ObsWin.
5. Create the P-Unit list. Define which sources are Cat. I and which sources are Cat. II. Also define which sources are to be combined in a Source Patch.
6. Create a set of Source Patches from the P-Unit list and the ObsRes.
7. Generate the 6 SubTrees for every P-Unit.

Now the LSM is ready for use. We generate a User Forest (for example using the Tree Defintion Language, TDL), attach the measured data and Instrumental Parameters to the User Forest and execute. First the Cat. I sources are Predicted, Solved for, and Subtracted. After prediction and subtraction of Cat. I sources, Residual Images are created from the Residual uv-data. Using the ObsRes these images are deconvolved and the P-Units are updated. Several Predict / Subtract / Update iterations may take place, and after the last iteration all LSM Parameters and Images are updated. Finally, the GSM is updated from the LSM.

3 Implementing the Local Sky Model

In this section the implementation of the Local Sky Model is discussed. As discussed in [1] the LSM is a temporary object and will be implemented as an object. Furthermore, the LSM object contains the five sub-objects: the ObsWin, the ObsRes, the GSM Subset, the P-Unit list, and the Set of Source Patches. Together with these objects we have to implement methods in order to have LSM functionality. This functionality will grow with time and it will be implemented in stages. This staged implementation is discussed in the next section.

The LSM can probably be implemented in Python using TDL (Olegs Tree Definition Language) to define the Trees. Especially in the uv-data interface this should be possible, since all can be done using Trees. For the Image data interface it is still uncertain. Probably AIPS++ is needed for the deconvolution of the residual images. It is not clear whether this can (already) be done using Python. [Note: if it is not possible to use Python in the Image Interface, we might set up a separate Glish branch dealing with this interface. This would be temporary until AIPS++ has the needed Python interfaces.]

The LSM has its own access to the MeqTree kernel. In this way the Trees that are inside the LSM can be used in the construction of the different objects. For instance in order to determine the Apperent Brightness of a source first the Intrinsic Brightness may be obtained through the Source Tree.

A sketch of the current stage of implementation of the LSM is shown in Figure 2.

3.1 The LSM Object

Methods

- constructor, destructor
- add source
- visualization

3.2 The ObsWin

The Observational Window (ObsWin) is the Average Station Power Beam. Every Station has a (slightly?) different power beam. The ObsWin is the Power Beam averaged over all stations. It lives in the Image plane and includes Beam Polarization effects. Hence, the ObsWin can be described by an Image for StokesI/Q/U/V. The ObsWin is described in local l, m coordinates relative to the phase centre of the observation.

Obviously, the ObsWin is frequency dependent, but we will allow for the option that it is also time dependent. This is perhaps too computationally demanding for LOFAR right now (it will generate time dependent uv-data!), but it may be of great use for smaller telescopes (e.g. the CLAR).

Implementation: The ObsWin is implemented by a SubTree. Its Tree takes MeqParms as leaf nodes. Given a Request (*l, m* / time / freq) it produces a 5D Result (*l, m*, Stokes, freq, time). The resolution and the extent of the ObsWin will depend on the Request. For use in Apparent Mode the ObsWin is just a Top Hat function.

Questions

- Can we have *l, m* in the Request? Note that *l, m* are *local* coordinates!
- Does the ObsWin change with changing phase center? This may be of relevance to changing Patches and / or Facets.
- How do we deal with observations using stations with completely different station beams? E.g. WSRT & WHAT. Do we have multiple ObsWins?

Methods

- constructor, destructor

3.3 The ObsRes

The Observational Resolution (ObsRes) gives the spatial and spectral resolution of an observation. As such it contains

- The spectral range and resolution;
- The minimun and maximum baseline;
- An (approximated) Point Spread Function (PSF). The PSF is determined by the discrete uv-coverage and is used in the deconvolution of Residual Images. By having just the 'main lobe' of the PSF a Clark-CLEAN deconvolution can be used.

Implementation: Where the ObsWin is basically just one image, the ObsRes is a collection of data. We could either implement them as separate objects or as a Record of ObsRes objects. The PSF is parameterized and given by a Tree. Hence, resolution and extent depend on the Request. The PSF will be frequency dependent, but not time dependent.

Questions

- For the implementation of the ObsRes PSF we have the same questions as for the ObsWin.

Methods

- constructor, destructor
- The ObsRes is used in deconvolvin Residual Images
- The ObsRes (and/or the ObsWin) are used in determining the size of Patch Images

3.4 A subset of the GSM

The subset of the GSM contains a LSM source list, MeqParm Tables, 4D GSM Images and Template SubTrees. Here, the LSM source list is the binding factor. For every source it has an entry containing: source name, MeqParm Table reference / GSM Image reference, SubTree reference. This list connects for every source in the LSM its parameters or underlying Image to a Predict SubTree.

Implementation:

- The LSM Source List is implemented as a Table having three columns: Name, TreeType, and TableName. All three entries are strings; e.g. Name may be '3C343.1', TreeType may be 'Point_NEWSTAR', and TableName may be 'Table1'.
- The MeqParm table is an AIPS++ Table. The LSM is shielded from it, since it is automatically accessed through the MeqParm nodes of the Tree. This is all C++ implementation within the MeqParm.
- The Template Trees are implemented as a Table having 7 columns: TreeType, RA Tree, Dec Tree, StokesI Tree, StokesQ Tree, StokesU Tree, and StokesV Tree. The 6 trees are called a sixpack.

Methods

- constructor, destructor
- add source, delete source
- add Tree Template, delete Tree Template
- visualization of Source Table

3.5 The P-Unit List

The P-Unit list is the controlling object for the data processing. Every P-Unit (Peeling / Predict Unit) is treated as a single unity, although it can contain (many) different sources.

Implementation: The P-Unit list is implemented as a Table having the following columns:

- P-Unit name (e.g. CasA, Sun, 3C273, Patch23);
- P-Type: 'Point' or 'Patch';
- P-Unit Source List: List of source names (strings) that are contained in the P-Unit. This could be 1 up to very many;
- Cat. I: True or False. Treat the P-Unit as a Cat. I source or not.
- Apperent Brightness: real value. This allows an ordering for the Peeling algorithm.
- Tree Root node reference: sixpack (6 Trees) for the P-Unit.
- FOV Distance: real value. This gives the distance to the phase center of the observation. If the value is larger than 1 the source is outside the Field of View (like CasA, the Sun, Jupiter). Values smaller than 1 indicate that the source is within the FOV.

Questions

- How and when do we decide which sources go into a Patch?

Remark The P-Unit Source list is the only place where we know which sources are contained in which Patch. This is of importance for the updating process of the LSM source parameters (of the GSM Subset).

Methods

- constructor, destructor
- construct the SubTrees for every P-Unit

3.6 A set of Source Patches

Somewhere in the LSM we have to store the actual Source Patches. Note that these patches have the correct resolution and size, since they were created using the ObsRes.

Implementation: A Source patch is implemented in an Image VellSet. We could have Prediction SubTrees from the GSM subset into a Source Patch. Furthermore, there are Prediction SubTrees from the Source Patch to the User Forest.

Remark

- We still need to figure out a few things about the actual implementation. This has to do with closing the loop of uv-data predict and residual image update.

Questions

- Is it possible to not always pass Results/Requests down/up the Tree? We would like to first iterate through parts of the Forest in order to get Updated Patches and only after that update the underlying Source parameters. Hence the connection to the GSM SubTrees must be allowed to be (temporarily) blocked.

Methods

- constructor, destructor

4 A staged implementation

In this section the implementation of the LSM is discussed in all its stages. First the uv-data interface is developed. This is done in Stages 1 to 5. In these stages an initial ObsWin based on a modelled WSRT beam is used. In Stage 6 the ObsWin is considered in full and this may lead to the first GSM interface. Although we might still decide to postpone the actual interface to a later stage. In Stage 7 the updating from Residual Images is considered, so here the Image date Interface is being developed.

Since no GSM will be available in the early stages, the source list will have to be constructed by hand or script. Maybe some automatic source finding will be considered before Stage 7 and this would mean some early Image Interface.

4.1 Stage 1: 3C343.1 & 3C343

Prediction of Point Sources.

The field of 3C343.1 and 3C343 is relatively simple which is ideal for setting up the basic system. This is particularly important in order to get the interface between the P-Unit list and the User Forest right.

The field consists of two dominant point sources: one in the field center and the other at the half power beam. They need separate calibration parameters. Apart from these two sources there are about 20 faint background sources. Since there are still relatively few background sources we might still predict them by DFT (note that they are too faint to solve for) and they could share instrumental parameters in groups. Since Michiel Brentjes already processed the data we know how to use the MeqTree system for Self-Calibration of this field.

The 3C343.1 & 3C343 field allows for the following simplifications:

- There are only Points Sources
- The data processing is done in apparent mode or using a simple ObsWin Tree based on a WSRT beam model
- No GSM selection needs to be made (a LSM source list will be made by hand)

The simplifications have the following implications on the implementation:

ObsWin: A simple ObsWin Tree is needed. This Tree implements a model for the WSRT beam.

ObsRes: Not needed yet, since we do not update from Residual Images.

GSM subset: The Source List is needed, together with 6 template Trees using one type of Point Source parametrization (using varying spectral index). The parameters are stored in a AIPS++ ParmTable. (Since there are only few sources to be considered we could use all the actual trees instead of template trees. The template trees then enter in Stage 3.)

P-Unit list: The P-Unit list is needed. Since this is the controlling object it probably means that it is implemented in full.

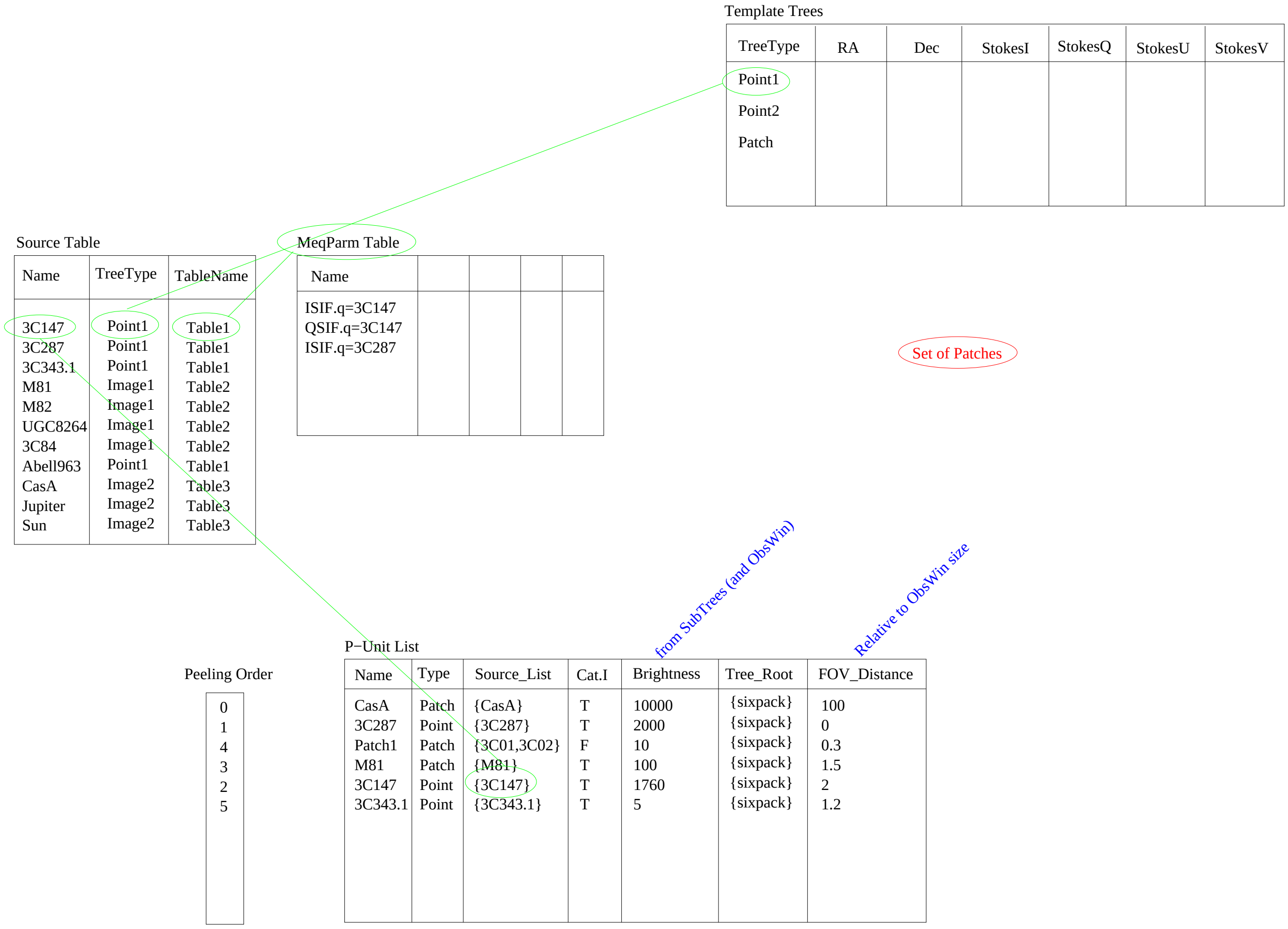
Set of Source Patches: Not needed yet.

A sketch of the implementation of the LSM in this stage is given in Figure 2.

4.2 Stage 2: 3C286

Polarized Point Source in the center of the field.

3C286 is a polarized WSRT calibrator source. Jan Noordam has polarization processing Trees and this stage is meant to test the polarization part of the LSM Trees. Since these Trees are already implemented in Stage 1, there is no additional implementation needed here. Only testing is needed.



Template Trees

TreeType	RA	Dec	StokesI	StokesQ	StokesU	StokesV
Point1						
Point2						
Patch						

Source Table

Name	TreeType	TableName
3C147	Point1	Table1
3C287	Point1	Table1
3C343.1	Point1	Table1
M81	Image1	Table2
M82	Image1	Table2
UGC8264	Image1	Table2
3C84	Image1	Table2
Abell963	Point1	Table1
CasA	Image2	Table3
Jupiter	Image2	Table3
Sun	Image2	Table3

MeqParm Table

Name				
ISIF.q=3C147				
QSIF.q=3C147				
ISIF.q=3C287				

Peeling Order

0
1
4
3
2
5

P−Unit List

Name	Type	Source_List	Cat.I	Brightness	Tree_Root	FOV_Distance
CasA	Patch	{CasA}	T	10000	{sixpack}	100
3C287	Point	{3C287}	T	2000	{sixpack}	0
Patch1	Patch	{3C01,3C02}	F	10	{sixpack}	0.3
M81	Patch	{M81}	T	100	{sixpack}	1.5
3C147	Point	{3C147}	T	1760	{sixpack}	2
3C343.1	Point	{3C343.1}	T	5	{sixpack}	1.2

Figure 2: Implementation of The Local Sky Model for Stages 1 and 2

4.3 Stage 3: 3C147; LFFE & WHAT

Prediction of Patches of Point Sources.

3C147 is another WSRT Calibrator source (unpolarized). Apart from 3C147 the field contains about 2000 Point Sources. So the LSM should be able to deal with this. This means prediction of multiple Point Sources through Patch Images.

3a: LFFE We start with using the WSRT telescopes only and do the processing using an ObsWin based on the WSRT model beam.

3b: WHAT Once the 'traditional' result is obtained we include the WHAT station and do the processing all over, using the 'traditional' result. We may then solve for the WHAT beam (relative to the WSRT beam model.

If the WHAT station is included we might want to be able to solve for station parameters, escpecially the beam. However, at this moment we do not want to apply Image Plane effects to Image Patches (UVBricks). So we might want to discuss how to deal with this.

4.4 Stage 4: Abell 963

Prediction of slightly Extended Sources.

Abell 963 is a field that contains some slightly extended sources. In this stage we have to deal with sources that are stored in images and further test the UVBrick functionality. The following simplifications are kept:

- The data processing is done in apparent mode, hence we do not apply beam shape effects in the UVInterpol. Instead constant gain error over the (small) extended sources is assumed.
- No GSM selection needs to be made (a LSM source list will be made by hand)

4.5 Stage 5: M81 / M82

Prediction of Extended Sources.

In this stage the extension of the Extended Sources is increased. No additional LSM functionality is needed. This field is basically used to test the scaling of the Patch Images.

4.6 Stage 6: UGC 8264

Solve for (relative) beamshapes using the bright point sources in the field. Now we are working in intrinsic mode, so here the full ObsWin enters the game.

4.7 Stage 7: 3C84

Cat II Update.

At this stage we are going to update from Residual Images. Hence, now the ObsRes is needed.

References

[1] J.E. Noordam, R.J. Nijboer *Functional description of the Local Sky Model*, LOFAR document (in progress) 2005.