

IRAM Memo 2011-2

WIFISYN: The **GILDAS** implementation of a new wide-field synthesis algorithm*

J. Pety^{1,2}, N. Rodríguez-Fernández¹

1. IRAM (Grenoble)
2. Observatoire de Paris

Jan, 19th 2011

Version 0.1

Abstract

The usual way to image wide-field interferometric observations is known as mosaicking. Different variants of mosaicking exist (*e.g.* Cornwell, 1988; Sault et al., 1996), including an interesting recent implementation of mosaicking in the uv plane (golap et al., priv. comm.). Pety & Rodríguez-Fernández (2010) revisited the theory of wide-field imaging to propose a different algorithm to image interferometric wide-field observations, based on the well-known Ekers & Rots' scheme. This algorithm is named wide-field synthesis because it *explicitely* synthesizes the wide-field spatial frequencies throughout the uv plane. This memo describes the current state of the implementation of this algorithm in a new package, named **WIFISYN**, of the **GILDAS** software suit.

*This work was mainly funded by the European FP6 "ALMA enhancement" grant. It was also partially funded by the grant ANR-09-BLAN-0231-01 from the French *Agence Nationale de la Recherche* as part of the SCHISM project.

Contents

1	Introduction	3
2	Theory	3
2.1	Observation setup and measurement space	3
2.2	Processing by explicit synthesis of the wide-field spatial frequencies	5
3	Practice	5
3.1	Simulating a wide-field observation	5
3.2	Fourier transform along α_s and β_s	7
3.2.1	Gridding	7
3.2.2	Reordering	8
3.2.3	Wide-field synthesis	9
3.3	Shifting and averaging	11
3.4	Getting the dirty image through inverse Fourier transform	11
4	Conclusion	12
A	Implementation plan	14
B	WIFISYN Language Internal Help	17
B.1	Language	17
B.2	COMPLEX	17
B.3	FFT	17
B.4	LOAD	18
B.5	READ	18
B.6	SETUP	18
B.7	UVBEAM	18
B.8	UVGRID	18
B.9	UVMAP	18
B.10	UVSYMMETRY	19
B.11	VARIABLE	19
B.12	WIFI2VISI	19
B.13	WRITE	19

1 Introduction

The usual way to image wide-field interferometric observations is known as mosaicking. Different variants of mosaicking exist (*e.g.* Cornwell, 1988; Sault et al., 1996), including an interesting recent implementation of mosaicking in the uv plane (golap et al., priv. comm.). Pety & Rodríguez-Fernández (2010) revisited the theory of wide-field imaging to propose a different algorithm to image interferometric wide-field observations, based on the well-known Ekers & Rots’ scheme. This algorithm is named wide-field synthesis because it *explicitely* synthesizes the wide-field spatial frequencies throughout the uv plane. This memo describes the current state of the implementation of this algorithm in a new package, named WIFISYN, of the GILDAS software suit. The first section summarizes the ideas underlying the proposed algorithm. The second section demonstrates the different steps taken to implement the algorithm (*i.e.* simulation, gridding, Fourier transform along the scanned sky coordinates to synthesize the wide-field visibilities, application of a shift-and-average operator to obtain the wide-field uv plane and inverse Fourier transform to yield the dirty image). We conclude on the additional needed steps to use this new algorithm on a daily basis. Appendix A includes the implementation plan written before coding WIFISYN. Appendix B includes the document of the current user interface of the WIFISYN package.

2 Theory

Figure 1 illustrates the principles underlying 1) the setup to get interferometric wide-field observations and 2) our proposition to process them. For simplicity, we display the minimum possible complexity without loss of generality. The top row displays the sky plane. The middle row represents the 4-dimensional measurement space at different stages of the processing.

2.1 Observation setup and measurement space

Panel a) displays the sky region for which we aim for estimating the sky brightness, $I(\alpha)$. The field of view of an interferometer observing in a given direction of the sky has a typical size set by the primary beam shape. In our example, this is illustrated by any of the circles whose diameter is θ_{prim} . As we aim at observing a wider field of view, *e.g.* θ_{field} , the interferometer needs to scan the targeted sky field. We assume that we scan through stop-and-go mosaicking, ending up with a 7-field mosaic.

After calibration, the output of the interferometer is a visibility function, $V(u_p, \alpha_s)$, whose relation to the sky brightness is given by the measurement equation

$$V(u_p, \alpha_s) = \int_{\alpha_p} B(\alpha_p - \alpha_s) I(\alpha_p) e^{-i2\pi\alpha_p u_p} d\alpha_p, \quad (1)$$

where V is the visibility function of 1) u_p (the spatial frequency with respect to the fixed phase center) and 2) α_s (the scanned sky angle), I is the sky brightness, and B the antenna power pattern or primary beam of an antenna of the interferometer. Panel b.1) shows the measurement space as a mosaic of single-field uv planes: The uv plane coverage of each single-field observation is displayed as a blue sub-panel at the sky position where it has been measured and which is featured by the red axes. We assume 1) that the interferometer has only 3 antennas and 2) that only a single integration is observed per sky position. This implies only 6 visibilities per single-field uv plane.

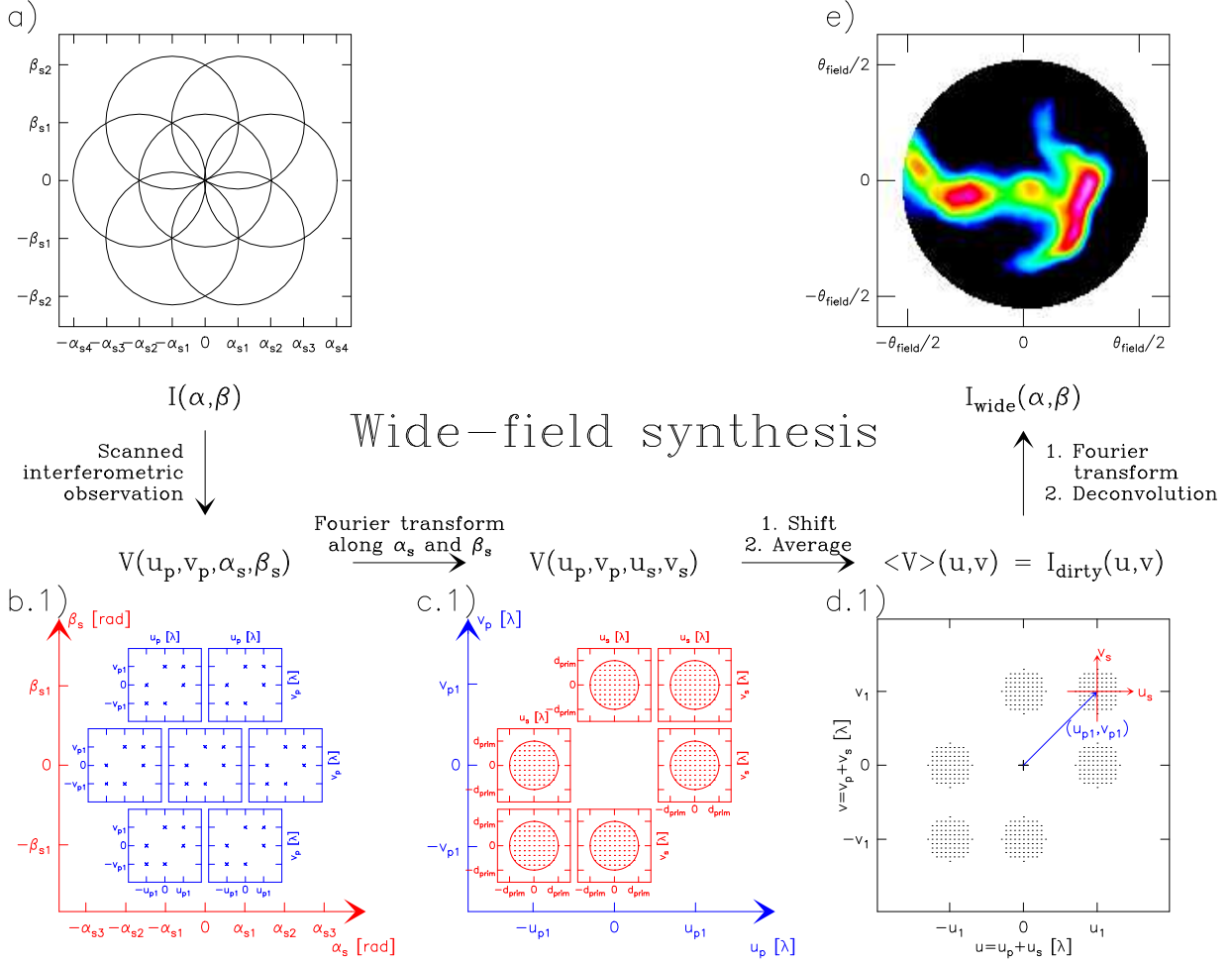


Figure 1: Illustration of the principles of wide-field synthesis, which enables us to image wide-field interferometric observations. The top row displays the sky plane. The bottom row displays the 4-dimensional visibility space at different stages of the processing. In panels b) to d), the scanned dimensions (α_s and u_s) are displayed in blue while the phased spatial scale dimensions (u_p) are displayed in red and the spatial scale dimensions (u) of the final wide-field uv plane are displayed in black. In detail, panel a) shows a possible scanning strategy of the sky to measure the unknown brightness distribution at high angular resolution: For simplicity it is here just a 7-field mosaic. Panel b.1) and b.2) sketch the space of measured visibilities: The uv plane at each of the 7 measured sky positions is displayed as a blue square box in panel b.1) and a blue vertical line in panel b.2). For simplicity, only 6 visibilities are plotted in panel b.1). Panels c.1) and c.2) sketch the space of synthesized visibilities after Fourier transform of the measured visibilities along the scanned coordinate (α_s): At each measured spatial frequency u_p (displayed on the blue axes) is associated one space of synthesized wide-field spatial frequencies displayed as one of the red squares in panel c.1) and the red vertical lines in panel c.2). The wide-field spatial scales are synthesized 1) on a grid whose cell size is related to the total field of view of the observation and 2) only inside circles whose radius is the primary diameter of the interferometer antennas. Panels d.1) and d.2) display the final, wide-field uv plane. This plane is built by application of a shift-and-average operator. Inverse Fourier transform and deconvolution methods then produce a wide-field distribution of sky brightnesses as shown in panel e).

2.2 Processing by explicit synthesis of the wide-field spatial frequencies

All the information about the sky brightness, $I(\alpha)$, is somehow coded in the visibility function, $V(u_p, \alpha_s)$. The high spatial frequencies (from d_{\min} to d_{\max}) are clearly coded along the u_p dimension. The uncertainty relation between Fourier conjugate quantities also implies that the typical spatial frequency resolution along the u_p dimension is only d_{prim} because the field of view of a single pointing has a typical size of θ_{prim} . However, wide-field imaging implies measuring all the spatial frequencies with a finer resolution, $d_{\text{field}} = 1/\theta_{\text{field}}$. The missing information must then be hidden in the α_s dimension.

Pety & Rodríguez-Fernández (2010) show that Fourier transforming the measured visibilities along the α_s dimension (*i.e.* at constant u_p) can synthesize the missing spatial frequencies, because the α_s dimension is sampled from $-\theta_{\text{field}}/2$ to $+\theta_{\text{field}}/2$, implying a typical spatial-frequency resolution of the u_s dimension equal to d_{field} . Conversely, the α_s dimension is probed by the primary beams with a typical angular resolution of θ_{prim} , implying that the u_s spatial frequencies will only be synthesized inside the $[-d_{\text{prim}}, +d_{\text{prim}}]$ range. Panel c) illustrates the effects of the Fourier transform of $V(u_p, u_s)$ along the α_s dimension. The red subpanels display the u_s spatial frequencies around each constant u_p spatial frequency.

In panel d) (*i.e.* after the Fourier transform along the α_s dimension), $\bar{V}(u_p, u_s)$ contains all the measured information about the sky brightness in a spatial frequency space. However, the information is ordered in a strange and redundant way. Indeed, we show that $\bar{V}(u_p, u_s)$ is linearly related to $\bar{I}(u_p + u_s)$. To first order, the information about a given spatial frequency u is stored in all the values of $\bar{V}(u_p, u_s)$ which verifies $u = u_p + u_s$.

A shift operation will reorder the spatial scale information and averaging will compress the redundancy (illustrated by the halving of the number of the space dimensions). The use of a shift-and-average operator thus produces a final uv plane containing all the spatial scale information to image a wide field in an intuitive form. We thus call this space the *wide-field uv plane*. Panel d) displays this space, where the minimum relevant spatial frequency is related to the total field of view, while the maximum one is related to the interferometer resolution.

Applying the shift-and-average operator to \bar{V} produces the Fourier transform of a dirty image, which is a local convolution of the sky brightness by a slowly varying dirty beam (Pety & Rodríguez-Fernández, 2010). As a result, inverse Fourier transform of $\langle \bar{V} \rangle$ and deconvolution methods will produce a wide-field distribution of sky brightness as shown in panel e) at the top right of Fig. 1.

3 Practice

In the real world, the visibility function is not only sampled, but this sampling is incomplete for two main reasons. 1) The instrument has a finite spatial resolution, and the scanning of the sky is limited, implying that the sampling in both planes has a finite support. 2) The uv coverage and the sky-scanning coverage can have holes caused either by intrinsic limitations (*e.g.* lack of short spacings or small number of baselines) or by acquisition problems (implying data flagging).

3.1 Simulating a wide-field observation

In order to test the implementation, we need a controlled data set. We simulated the simplest meaningful case, *i.e.*, a point source at the phase center (at the position of the famous Horsehead

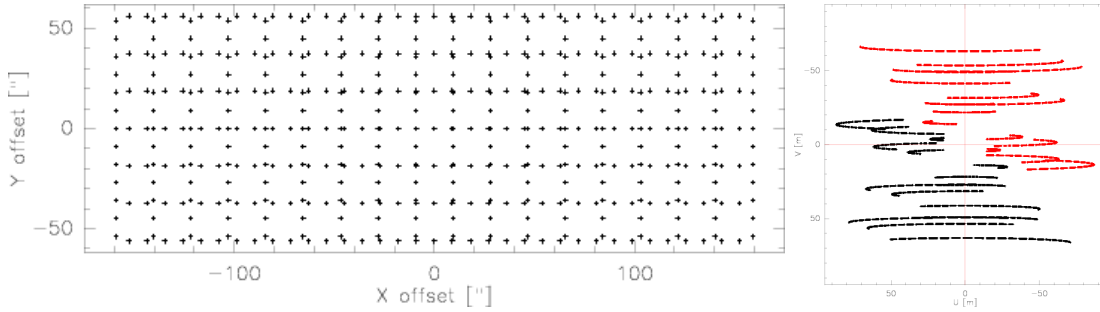


Figure 2: Sky and uv coverage of a simulated interferometric on-the-fly observation with the Plateau de Bure interferometer. **Left:** Each cross display the average sky position attributed to a set of visibilities recorded during one integration time. The center of the coordinate system is the phase center. **Right:** Each cross display the uv position of a measured visibilities whatever the associated sky position. The black crosses are measured. The red ones are deduced from the Hermitian symmetry of the visibilities.

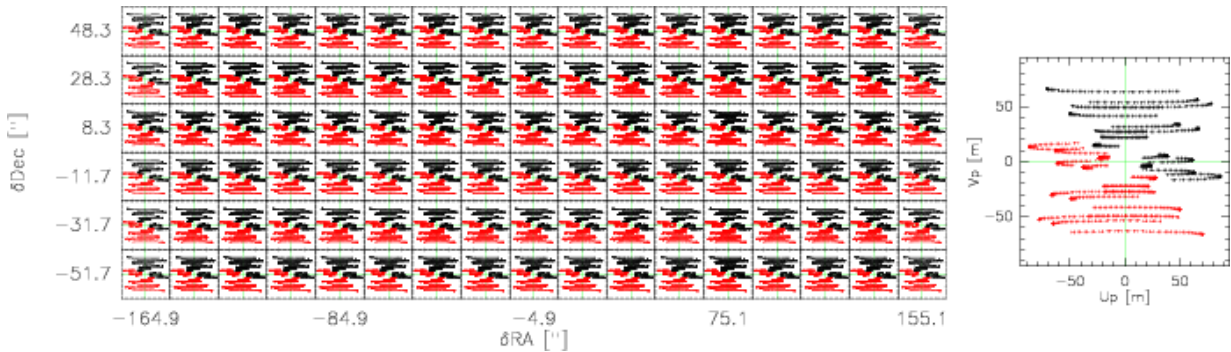


Figure 3: Chessboard display of the uv planes. **Left:** Each panel display the uv coverage of all the visibilities whose position on the sky falls in a $20''$ -range around the displayed sky coordinate. **Right:** Typical zoom of one of the uv coverages displayed left.

nebula in the Orion molecular cloud). We choose to simulate observations with the Plateau de Bure Interferometer because it allows us to clearly test the effect of incomplete instantaneous uv sampling due to the limited number of available antennas. The most compact configuration (D-configuration) was selected offering baseline length up to about 120 m. We simulated observations at the frequency of the ^{13}CO (J=1–0) line, *i.e.* 110.201 GHz or 2.72 mm. At this frequency, the primary beamwidth of 15m-antenna is $44.9''$.

We simulated On-The-Fly observations covering a total field of view of $300'' \times 100''$, which converts into the observation of about 17 independent fields. The field of view is rectangular to explore the effect of non-square pixels in the uv planes. We scanned the source along two perpendicular direction, named hereafter lambda and beta. The distance between two consecutive rows ensures Nyquist sampling, *i.e.*, $\lambda/D = 18.7''$. We sampled along the scanning direction at 5 points per primary beamwidth along the scanning direction to decrease the effect of beam elongation. We dumps at a rate of 0.5Hz, *i.e.*, one dump every two seconds. These two last considerations implies a scanning velocity of $4.5''$ per second. We observed one full track, *i.e.*,

during 9 consecutive hours alternating beta and lambda coverage in a loop with 5 minutes calibration measurements intersperse every 20 minutes. Each lambda or beta coverage lasts about 16 minutes. There were thus 47 coverages observed in total. Taking into account various overheads (*e.g.*, slew time between rows) and shadowing (about 13% of the visibilities lost), we estimate the observing efficiency to be about 60%, *i.e.* 60% of the time is spent on-source. We end up with a data set of 148 147 visibilities. If the data was imaged through standard mosaicking, it would imply between 500 and 10 000 actual fields depending on the acquisition system. Indeed, it is difficult to ensure that each scanning line always starts at exactly the same position on-sky for each coverage as it is ideally assumed in this simulation.

Fig. 2 separately displays the sky and uv coverage, each cross representing one measure. In reality, the uv coverage *a priori* depends on the position of the sky because of the Earth rotation. Fig. 3 thus display the uv coverage as a function of the position on the sky through a chessboard display. Each case of the chessboard displays the all the uv measures which falls in a $20''$ -range around the marked sky coordinate. The uv coverage is qualitatively similar over the full field of view as a result of the observing strategy. The left panel of Fig. 3 is a zoom of one of the uv coverages shown on the chessboard. The uv coverage displayed on Fig. 2 is much densely populated than any of the individual uv coverages.

3.2 Fourier transform along α_s and β_s

We want to Fourier transform the raw visibilities along the sky dimension (α_s and β_s) at some constant value of u_p and v_p . The raw data, however, is sampled on an irregular grid in both the uv and sky planes. We need to grid the measured visibilities in both the uv and the sky planes before Fourier transformation for different reasons. First, the gridding in the uv plane will handle the variation in the spatial frequency as the sky is scanned, *i.e.*, the difficulty and perhaps the impossibility of Fourier-transforming at a completely constant (u_p, v_p) values. Second, the gridding along the sky dimension enables the use of Fast Fourier Transforms.

3.2.1 Gridding

As usual, we grid through convolution and regular resampling. The visibilities must be convolved in 4 dimensions (α_s , β_s , u_p and v_p). For simplicity sake's, the convolution kernel is just the product of 4 one-dimensional functions along each convolution dimension. We used the standard spheroidal functions (cf. Fig. 4), whose Fourier transforms die off as quickly as themselves. This is a desirable mathematical property in processing interferometry data because it limits aliasing.

Fig. 5 displays the amplitude of the grided 4D visibility cube using a chessboard display, *i.e.* a set of uv images positionned at their right place in the ($\delta RA, \delta Dec$) sky coordinate system. The first obvious visual impression when the color scale is linear is that the visibility amplitude decreases isotropically as the distance from the point source position (at the center of the coordinate system) increases. Indeed, the decrease in amplitude mimic the primary beam shape which was used to simulate the visibilities, *i.e.* a Gaussian of $44.9''$ -beamwidth. The same chessboard is also displayed with a logarithmic color scale to emphasize edge effects. The gridded visibilities “abruptly” go to zero following an almost rectangular pattern. This is just the consequence of gridding along rectangular axes a limited field of view (the observed one) while the Gaussian primary beam has an unlimited support.

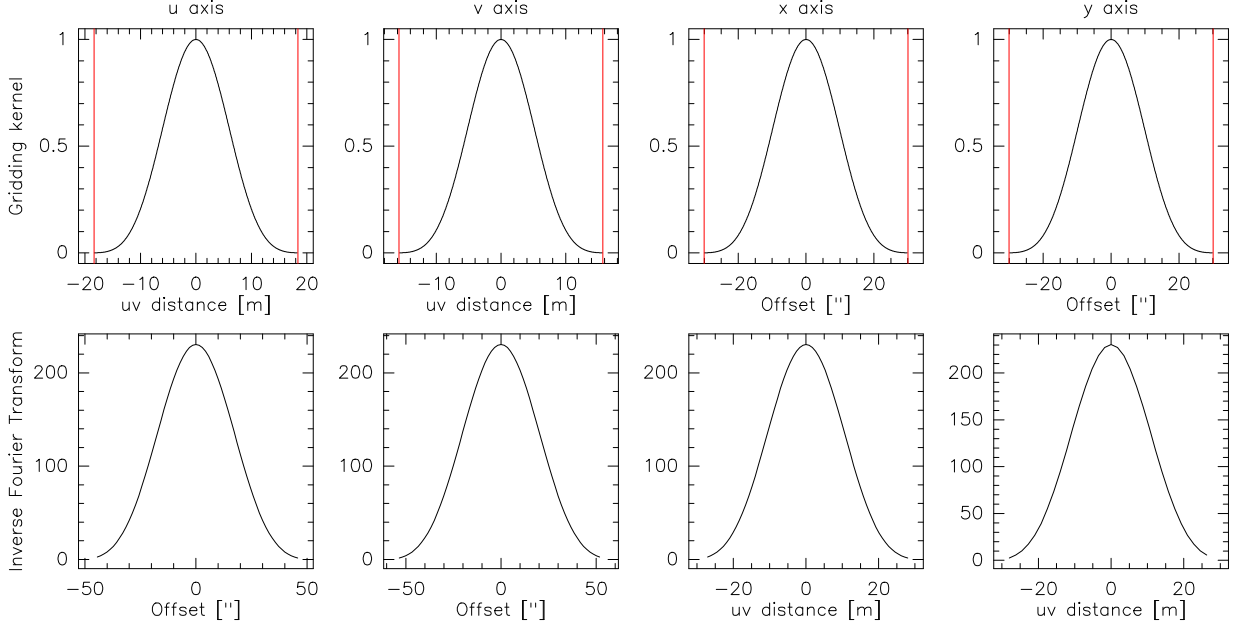


Figure 4: Convolution kernels used to grid the visibilities along the u_p (first column), v_p (second column), α_s (third column) and β_s (fourth column) dimension. The top row displays the used kernels while the bottom row displays their Fourier transform. The vertical red lines show the size of the support over which the kernels are computed and used.

The left panel is a zoom of the image displayed at the point source position on the chessboard. The visibility amplitude results here from the combination of the true visibility function (*i.e.*, a Gaussian for a point source at the phase center) and of the density and quality of the measurements at the considered position on the sky (because we grid the weighted visibilities, wV , and not just the visibilities themselves, V). The images on the chessboard resembles each other to first order because the uv coverage is similar over the whole field of view.

3.2.2 Reordering

The chessboards in Fig. 5 display the u_p and v_p visibility planes at constant values of α_s and β_s . However, we want to Fourier transform the grided visibilities along the sky dimensions (α_s and β_s) at constant value of u_p and v_p . We thus reorder in Fig. 6 the chessboards so that they display the α_s and β_s visibility planes at constant values of u_p and v_p . Nothing is actually done on the visibility cube itself, it is just a change in the way it is displayed. We note that we now recognize on the chessboard display the footprint of the uv plane sampling. Each image of the chessboards looks like the two zooms, which clearly show the Gaussian shape of the beam centered on the point source position. In particular, the images of the chessboards here are rectangular¹ (to keep the ratio aspect of the $\alpha_s\beta_s$ planes).

¹This is better seen on the chessboard with a logarithmic color scale.

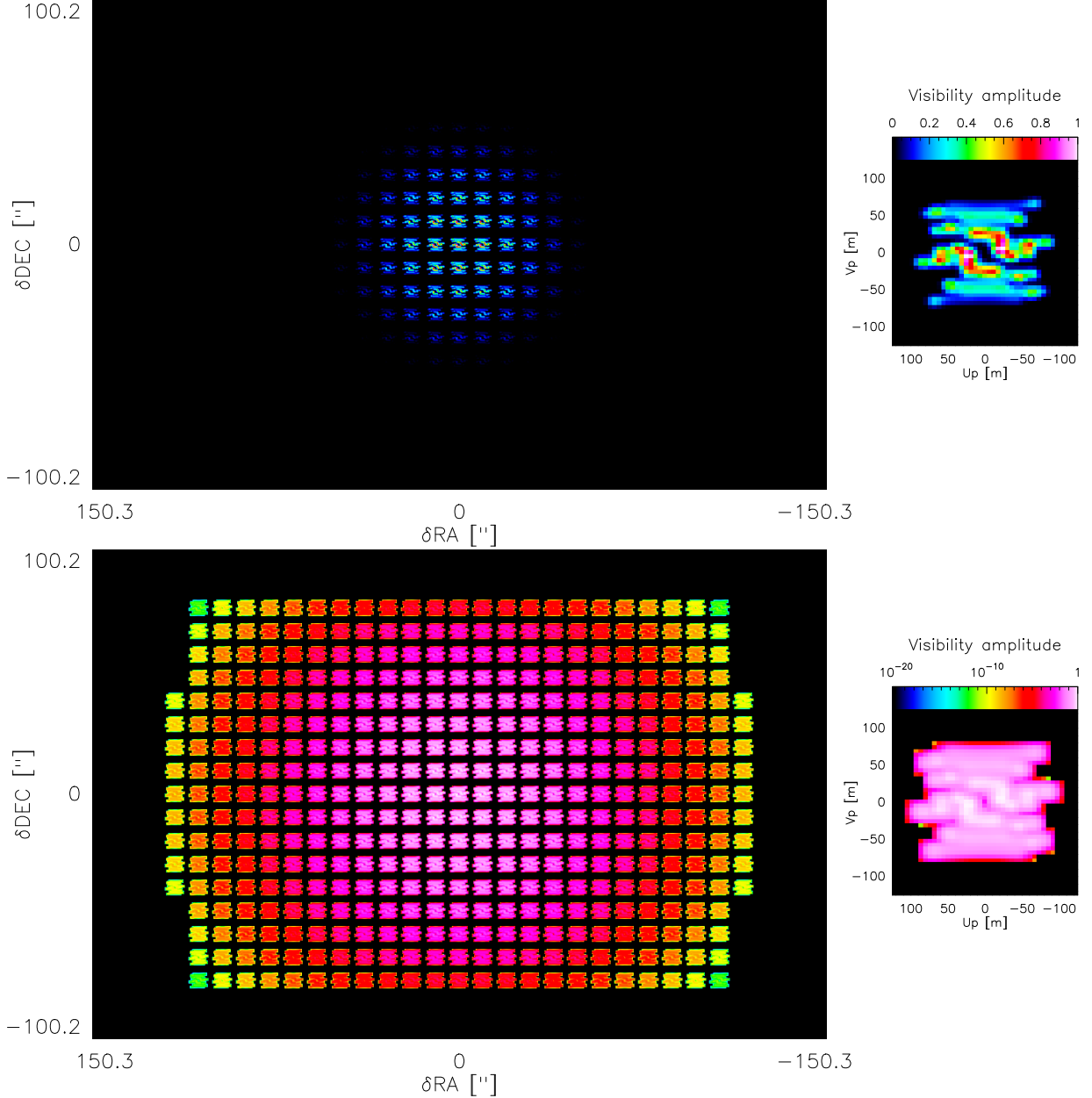


Figure 5: Chessboard display of the visibility amplitudes after the 4-dimensional gridding. The amplitudes were arbitrarily normalized to get a maximum value of 1 because this allows the reader to easily quantify the dynamic at which potential artifacts arise. **Left:** Each panel display the image of the visibility amplitude at constant value of the sky coordinates (α_s, β_s) and as a function of the uv coordinates (u_p, v_p) . **Right:** Typical zoom of one of visibility amplitude images. **Top:** The color scale is linear. **Bottom:** The color scale is logarithmic.

3.2.3 Wide-field synthesis

The chessboards in Fig. 9 display the visibility amplitude after Fourier transform along the (α_s, β_s) dimensions at constant (u_p, v_p) . We thus obtain a chessboards of (u_s, v_s) planes at their positions

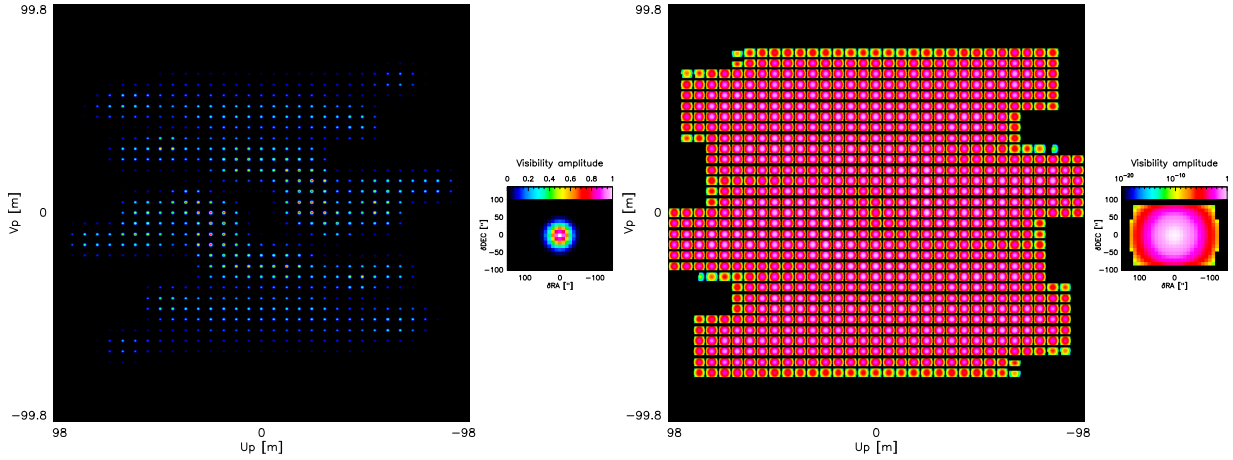


Figure 6: Same as Fig. 5 after reordering of the axes, *i.e.* each panel display the image of the visibility amplitude at constant value of the uv coordinates (u_p, v_p) and as a function of the sky coordinates (α_s, β_s) .

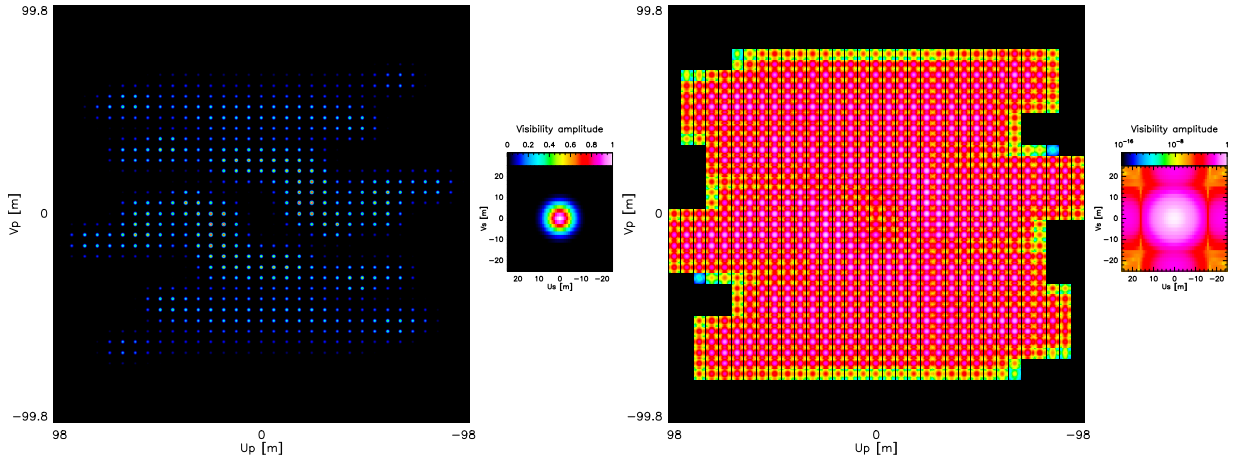


Figure 7: Chessboard display of the amplitudes of the synthesized wide-field visibilities after direct Fourier transform along the α_s and β_s axes at constant (u_p, v_p) values. The amplitudes were arbitrarily normalized to get a maximum value of 1. **Left:** Each panel display the image of the visibility amplitude at constant value of the wide scale uv coordinates (u_s, v_s) and as a function of the narrow scale uv coordinates (u_p, v_p) . **Right:** Typical zoom of one of visibility amplitude images. **Top:** The color scale is linear. **Bottom:** The color scale is logarithmic.

in the (u_p, v_p) plane. The Fourier transform of a Gaussian is a Gaussian. The zoom panels thus display to first order a Gaussian shape of full width at half maximum of about 15 m (*i.e.* the bare antenna diameter). The zoom panel using a logarithmic color scale clearly shows departure from the Gaussian shapes at the edges of the image because of the presence of “abrupt” edges in Fig. 6. This is at this step that the wide-field visibility are synthesized, although they are ordered in an unnatural way.

3.3 Shifting and averaging

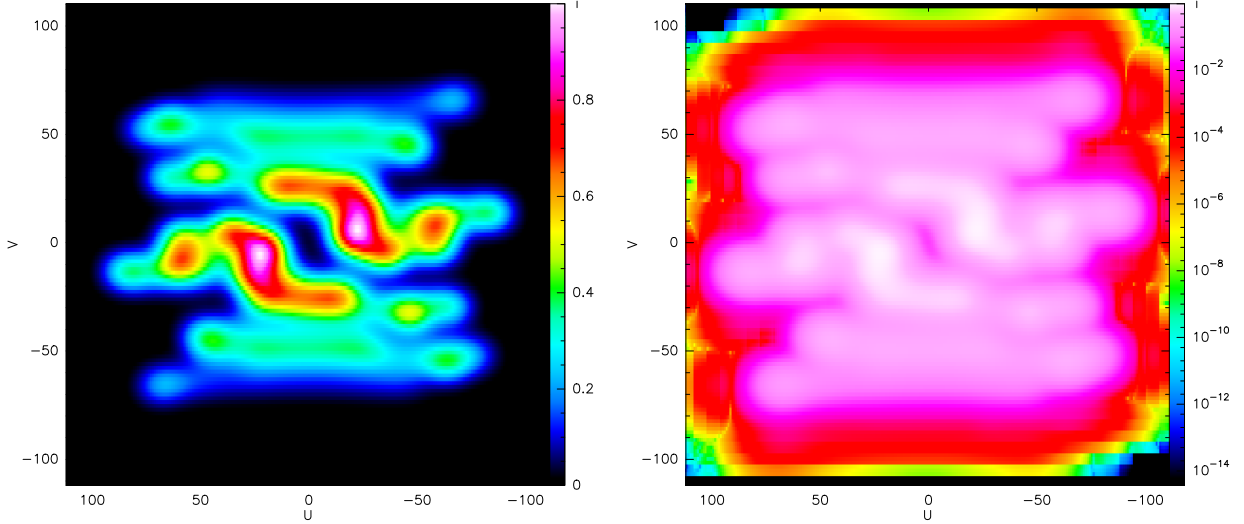


Figure 8: Wide-field 2D uv plane after shifting-and-averaging the wide-field 4D uv plane displayed in Fig. 7. The amplitudes were arbitrarily normalized to get a maximum value of 1. **Right:** The color scale is linear. **Left:** The color scale is logarithmic.

After the wide-field synthesis step, we obtain a 4D wide-field visibility cube for which each final uv spatial frequency (u,v) is measured for every (u_p, v_p, u_s, v_s) such that $u = u_p + u_s$ and $v = v_p + v_s$. The use of a shift-and-average operator allows us to get an intuitive wide-field uv plane because this first reorders the spatial frequencies at their right place and this then compresses them.

Fig. 8 display the wide-field uv plane, which results from this operation. It can be shown that the properties of the measurement equations imply that this wide-field uv plane must comply with the Hermitian symmetry, *i.e.* $V(u, v) = V^*(-u, -v)$. This is equivalent to state that the dirty image must be real. However, this Hermitian symmetry is only approximately enforced when the shift-and-average operator is used blindly over the 4D wide-field visibility cube, probably due to rounding errors. We thus enforces the Hermitian symmetry through 1) computation of only half of the wide-field uv plane (the one with negative values of v) and 2) deduction of the other half using the Hermitian symmetry.

3.4 Getting the dirty image through inverse Fourier transform

Once the wide-field uv plane is available, the dirty image is obtained by taking the real part of the 2D Fourier transform of this plane. The top panel of Fig. 9 displays this dirty image, which clearly shows a point source at the phase center. As this image is the response of the wide-field synthesized interferometer to a point source located at the phase center, it can be interpreted at the dirty beam at the phase center. In this framework, the side-lobes peaks at about 10% and there is a negative bowl surrounding it because of the missing zero-spacing. The bottom panel shows the absolute value of the dirty image with a logarithmic color scale to display the levels at which appear different kind of artefacts. Aliasing replications appear between 10^{-2} and 10^{-5}

of the peak (yellow-red replications of the main blob along the two main axes) and they become pronounced between 10^{-5} and 10^{-7} of the peak (green strips).

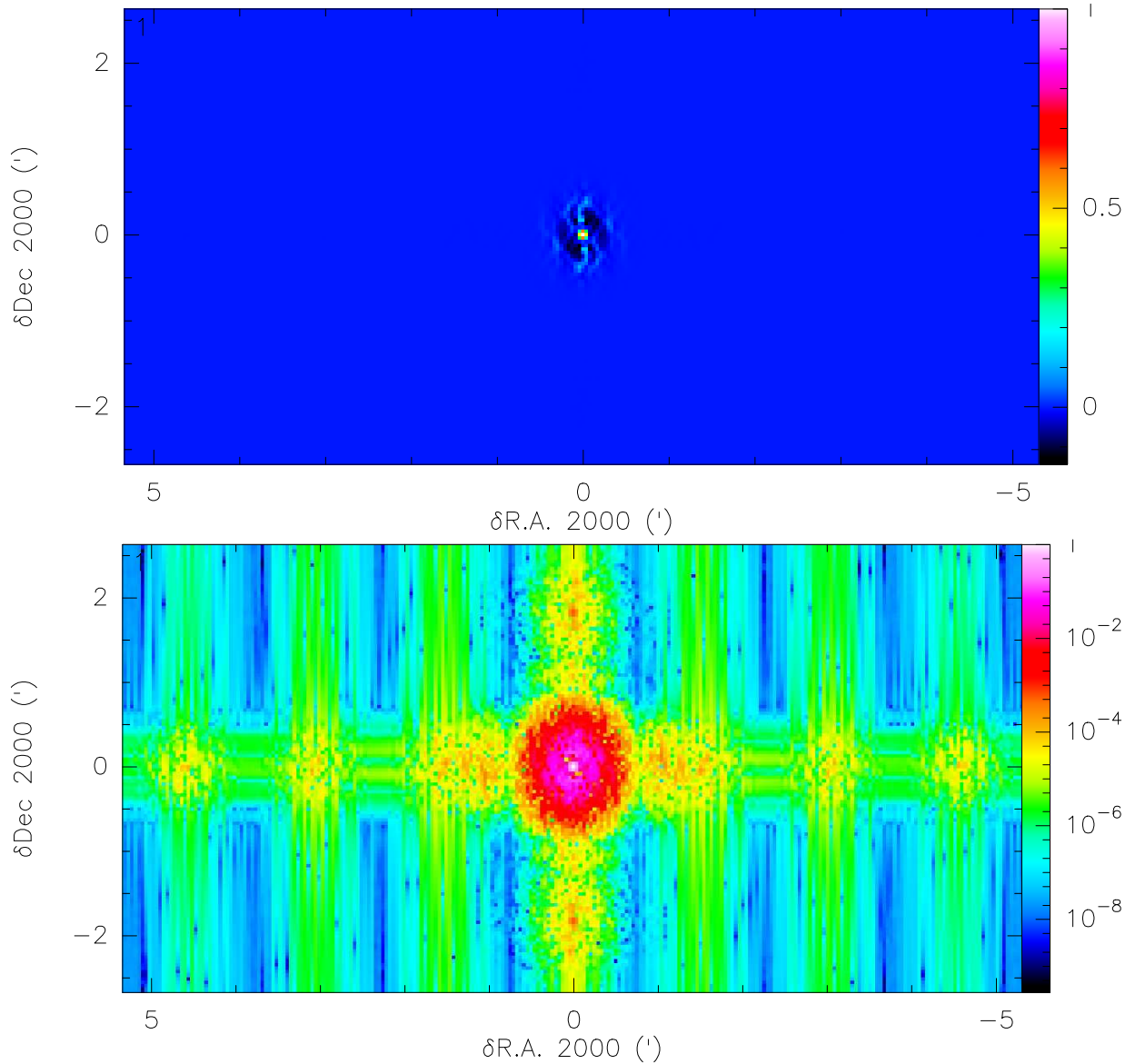


Figure 9: Dirty image after inverse Fourier transform of the wide-field 2D uv plane displayed in Fig. 8. The images were arbitrarily normalized to get a maximum value of 1. **Top:** The dirty image is displayed with a linear color scale. **Bottom:** The absolute value for the dirty image is displayed with a logarithmic color scale.

4 Conclusion

This memo describes the first implementation in the GILDAS software suit of the wide-field synthesis imaging algorithm, proposed in Pety & Rodríguez-Fernández (2010). It shows good promise

to process the large datasets produced by the on-the-fly interferometric observing mode. The current implementation does not yet include deconvolution, a mandatory step to get clean images compatible with the sky brightness distribution of the observed source. As a first step in this direction, we computed the sets of wide-field dirty beams associated with the observation. Indeed, the dirty beam slowly varies with the position of the sky because of the shift-variant nature of interferometric wide-field observations. Using these dirty beams, standard CLEAN deconvolution algorithms will be adapted to our imaging algorithm.

Acknowledgments. The authors thank S. Guilloteau for useful discussions at early stages of the WIFISYN implementation.

References

- Cornwell, T. J. 1988, A&A, 202, 316
- Pety, J. & Rodríguez-Fernández, N. 2010, A&A, 517, A12+
- Sault, R. J., Staveley-Smith, L., & Brouw, W. N. 1996, A&AS, 120, 375

A Implementation plan

This appendix displays the implementation plan written when the GILDAS prototype was started.

- * Step #1: Reading the uv and the xy (short-spacings) tables
 - Substeps (READ OTF)
 - + Read uv if available, results:
 - o Measured uv and sky positions (4 columns: up, vp, as, bs)
 - o Measured weights (nf columns)
 - o Measured visibilities (2*nf columns)
 - + Read xy if available, results:
 - o Measured sky positions (2 real columns: as, bs
2 virtual columns: up=0, vp=0)
 - o Measured weights (nf columns)
 - o Measured brightness (2*nf columns)
 - + If uv and xy available then
 - o Check consistency (spatial and frequency coordinates) of UV and XY tables
 - o Crop XY table to a reasonable size
 - o Convert XY into jansky
 - o Merge both tables (The origin of the data must be kept as additional columns)
 - Results
 - + Measured uv and sky positions (4 columns: up, vp, as, bs)
 - + Measured weights (nf columns)
 - + Measured visibilities (2*nf columns)
 - Comments
 - + We probably want to read several uv table coming from different instruments (e.g. ALMA + ACA). The difficulty is not so much in reading and merging the tables but in assigning the correct transfer function to the correct visibilities.
- * Step #2: Analysis of the data and definition of the tasks
 - Substeps:
 - + Sorting the visibilities in 3D if not already done
 - + Definition of the grid axes
 - + Definition of the kernels
 - Results:
 - + Sorted uv table
 - + Grided uv and sky axes (4 axes: upg, vpg, asg, bsg)
 - + Gridding Kernels (spheroidals)
 - Comments:
 - + The gridding function should depend on the kind of antennas
- * Step #3: Gridding
 - Substeps:

- + Convolution of weighted visibilities
 - + Convolution of weights
 - Results:
 - + Grided weights (5D cube: upg,vpg,asg,bsg,nu)
 - + Grided visibilities (5D cube: upg,vpg,asg,bsg,nu)
 - Comments:
 - + No gridding correction
- * Step #4: 2D DFFT along sky dimensions
- Substeps:
 - + 2D DFFT of grided visibilities
 - + Transformed weights (Hyp: independent weights before 2D FFT)
 - Results:
 - + Synthesized visibilities (5D cube: upg,vpg,usg,vsg,nu)
 - + Transformed weights (5D cube: upg,vpg,usg,vsg,nu)
- * Step #5: Shift-and-average
- Substeps:
 - + Definition of the weighting function (using the transformed weights, 5D cube: upg,vpg,usg,vsg,nu)
 - + Shift-and-average data
 - Results:
 - + Wide-field visibilities (3D cube: u,v,nu)
- * Step #6: 2D IFFT
- Substeps:
 - + 2D IFFT of wide-field visibilities
 - Results:
 - + Dirty image (3D cube: a,b,nu)
- * Step #7: Computation of the dirty beams
- Substeps:
 - + Computation of the transfer functions of the single-dish and interferometer antennas
 - + Computation of a uv table corresponding to point sources at position where the dirty beams must be computed
 - $$Sft(up, vp, as, bs, us, vs, nu) = S(up, vp, as, bs, nu) \cdot T(us, vs, nu) \cdot \exp(-i \cdot 2 \cdot \pi \cdot us \cdot as)$$
 - Results: A set of uv tables
 - + Loop over steps 3 to 6 on each table and store the dirty beams
 - + Fit of clean beams on the dirty beams
 - Results:
 - + Set of dirty beams (5D cube: a',b',a'',b'',nu)
 - + Set of clean beams (4D cube: (major,minor,angle),a'',b'',nu)
 - Comments:
 - + If only the width of the beam and not its shape varies with the frequency, then we can apply a dilatation of the coordinates

as a function of frequency, implying the computation of a single dirty beam for all the frequencies.

B WIFISYN Language Internal Help

This appendix describes the user interface of the WIFISYN package.

B.1 Language

```

COMPLEX      : Transfer complex internal buffers into SIC real images and
               vice-versa
FFT          : Compute direct or inverse FFT of internal buffers
LOAD        : Load an internal buffer into a SIC variable
READ        : Read the input files in internal buffers
SETUP       : Setup and display the imaging parameters for the internal
               uvxy table
UVBEAM      : Compute the dirty beams associated to the internal uvxy
               table
UVGRID      : Grid the internal uvxy table
UVMAP       : Compute the dirty image associated to the internal uvxy
               table
UVSHIFT     : Shift the phase center of the internal uvxy table
               (Not working yet)
UVSORT      : Sort the internal uvxy table
               (Not working yet)
UVSWAP      : Swap the UVT and UVR buffers
               (Not working yet)
UVSYMMETRY  : Check the Hermitian symmetry of the internal buffers
VARIABLE    : Map the imaging internal parameters onto the WIFI% SIC
               structure
WIFI2VISI   : Shift-and-average the wide-field hypercube to get a
               ''standard'' visibility plane
WRITE       : Write internal buffers into output files

```

B.2 COMPLEX

```
[WIFISYN\]COMPLEX VISICUBE|WIFICUBE TO|FROM AMPLITUDE|PHASE|RE-
AL|IMAGINARY
```

Transfer complex internal buffers (visicube or wificube) into SIC real images for display. The inverse operation is possible to set or to modify the complex internal buffers.

B.3 FFT

```
[WIFISYN\]FFT VISICUBE|WIFICUBE /DIRECT|INVERSE /UV|XY
```

Compute the direct or inverse 2D Fast Fourier Transform of internal complex buffers (visicube or wificube). When the FFT is applied to the wificube buffer, the 2D FFT can be applied along either the uv or the xy

axes.

B.4 LOAD

[WIFISYN\] LOAD Buffer Varname

Load the specified internal buffer (VISI, WEIGHT, BEAM, DIRTY) into the specified SIC variable (Varname).

B.5 READ

[WIFISYN\]READ Buffer File [/PLANE First Last]

Read the specified internal buffer (UV, BEAM, DIRTY) from the specified input File. Default extensions are respectively .tuv, .beam, .lmv. A range of channels can be specified using the /PLANE option.

B.6 SETUP

[WIFISYN\]SETUP

Setup and display the imaging parameters for the internal uvxy table.

B.7 UVBEAM

[WIFISYN\]UVBEAM

Compute the dirty beams associated to the internal uvxy table.

B.8 UVGRID

[WIFISYN\]UVGRID [/NOCONV]

Grid the internal uvxy table into a wificube. The /NOCONV option creates the wificube of the right dimension and initializes it so that the user can fill it through the COMPLEX command.

B.9 UVMAP

[WIFISYN\]UVMAP

Compute the dirty image associated to the internal uvxy table. This command is equivalent to the following sequence: SETUP, UVGRID, FFT WIFI /DIRECT /XY, WIFI2VISI, FFT VISI /INVERSE, COMPLEX VISI TO REAL.

B.10 UVSYMMETRY

[WIFISYN\]UVSYMMETRY

Check the Hermitian symmetry of the wificube (or visicube???) buffer.

B.11 VARIABLE

[WIFISYN\]VARIABLE

Map the imaging internal parameters onto the WIFI% SIC structure.

B.12 WIFI2VISI

[WIFISYN\]WIFI2VISI

Shift-and-average the wide-field hypercube to get a ''standard'' visibility UV plane, which can then be FFTed to get the dirty image.

B.13 WRITE

[WIFISYN\]WRITE Buffer File [/PLANE First Last]

Write the specified output File from the specified internal buffer (UV, BEAM, DIRTY). Default extensions are respectively .tuv, .beam, .lmv. A range of channels can be specified using the /PLANE option.