Image Reconstruction in Optical Interferometry: Image Reconstruction tool IRBis

Karl-Heinz Hofmann, Max-Planck-Institut for Radioastronomy Bonn

10th VLTI school, Nice/online, June 7-18 2021

Outline

- Applications of the image reconstruction tool IRBis to interferometric data

- Basic principles of IRBis

- Minimization routines in IRBis

- Additional features in IRBis: uv density weight & image quality parameter qrec

- Scan of image reconstruction parameters in IRBis

- Enough data?

- Optimal FOV and pixel grid for image reconstruction

- Image reconstruction parameter in IRBis

- Handling of IRBis: the image reconstruction script

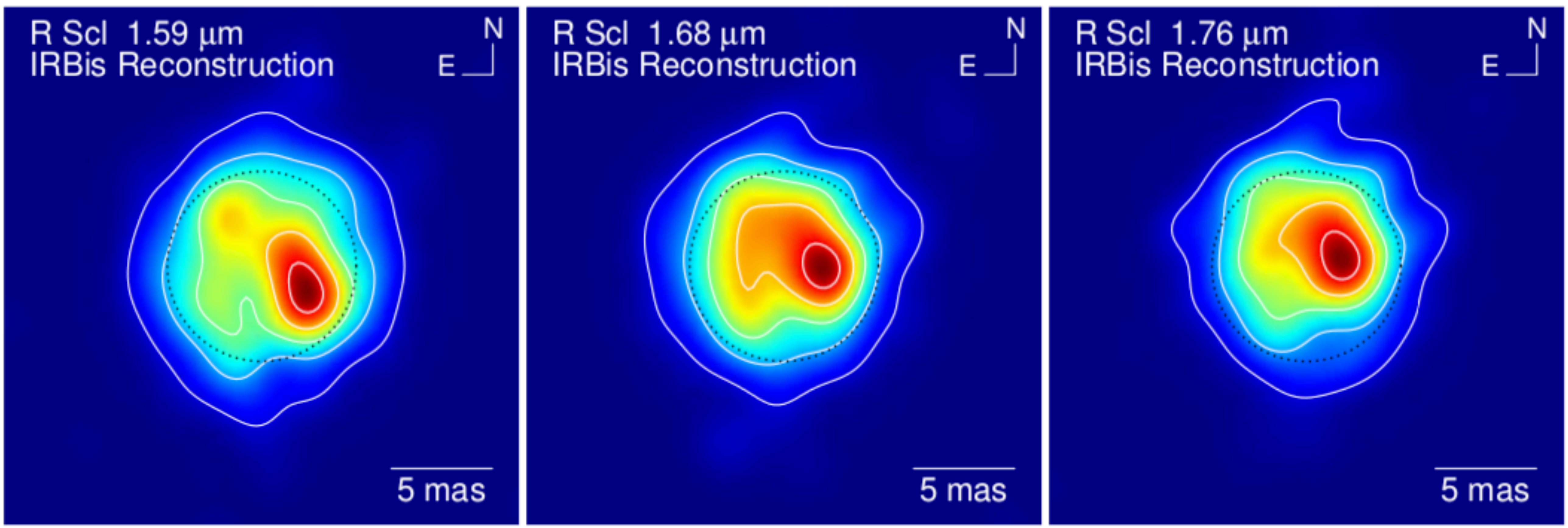
- IRBis image reconstruction example

PIONIER - IRBis
Image Reconstruction

R Scl

H-band aperture-synthesis imaging of the carbon AGB star R Scl (AT data).

- the detected complex structure within the stellar disk is caused by giant convection cells
- the dominant bright spot (western part of the disk) would be caused by a region where the stellar disk is less obscured by dust clouds



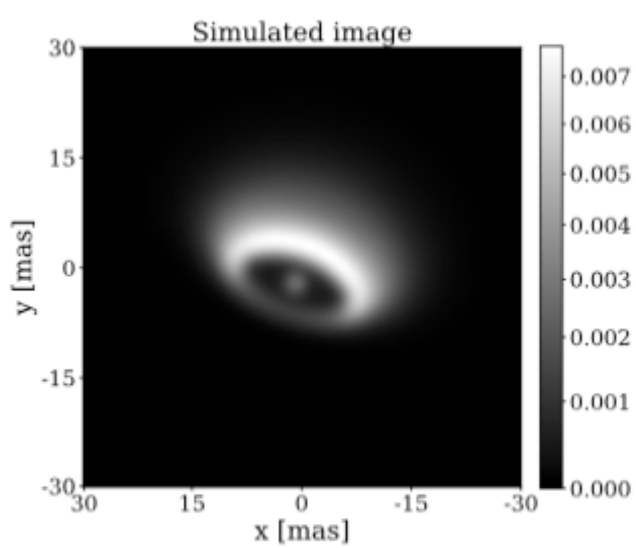
(Wittkowski + 2017)

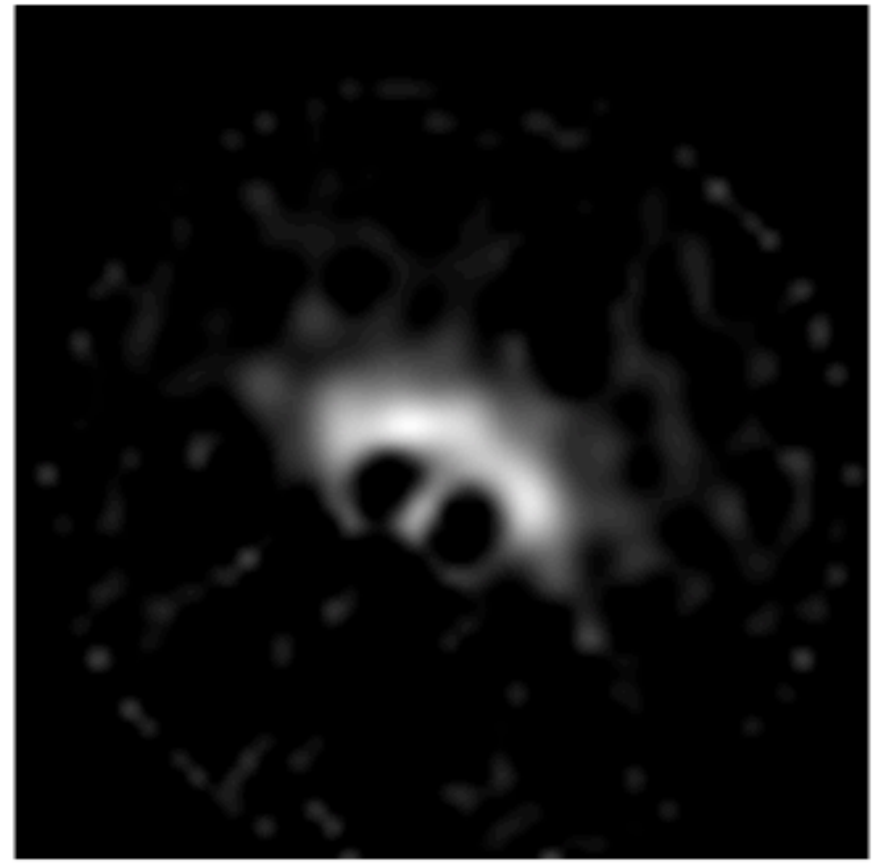
MATISSE - IRBis
Image Reconstruction

FS CMa

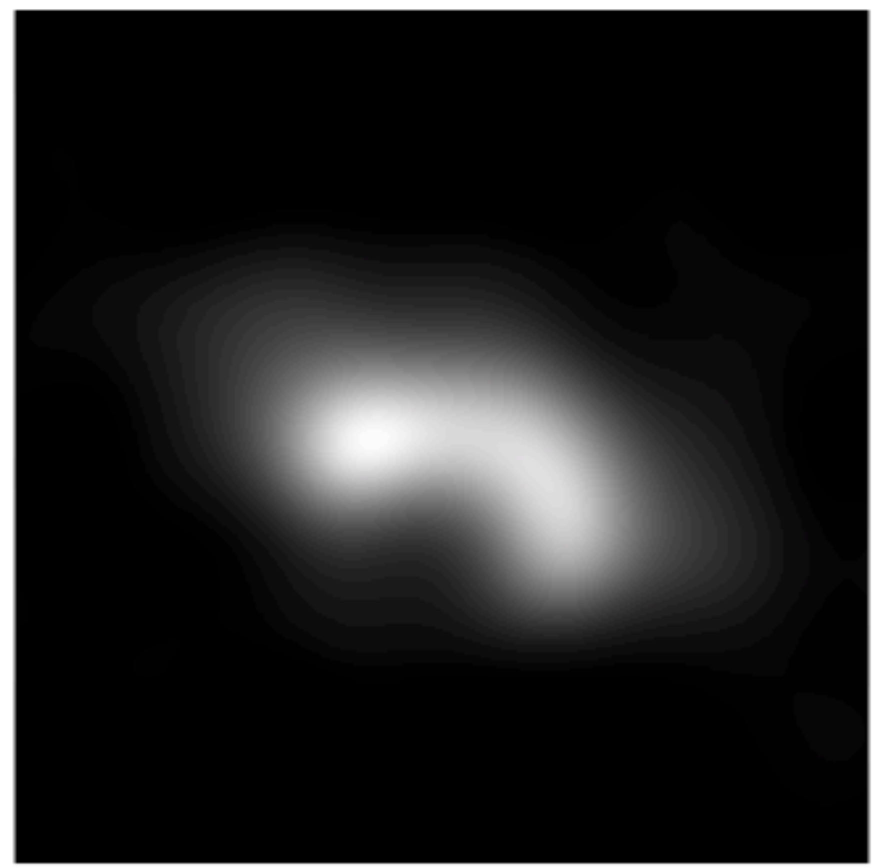
L- and N-band aperture-synthesis imaging of the unclassified B[e] star FS CMa (AT data)

- L-band image shows the inner rim of the inclined disk and the central star
- N-band disk is more extended than the L-band disk
- L-band RT model





- 3.4 - 3.8 mu
- FOV = 64 mas
- resolution ~ 2.7 mas



- 8.6 - 9.0 mu
- FOV = 64 mas
- resolution ~ 6.6 mas

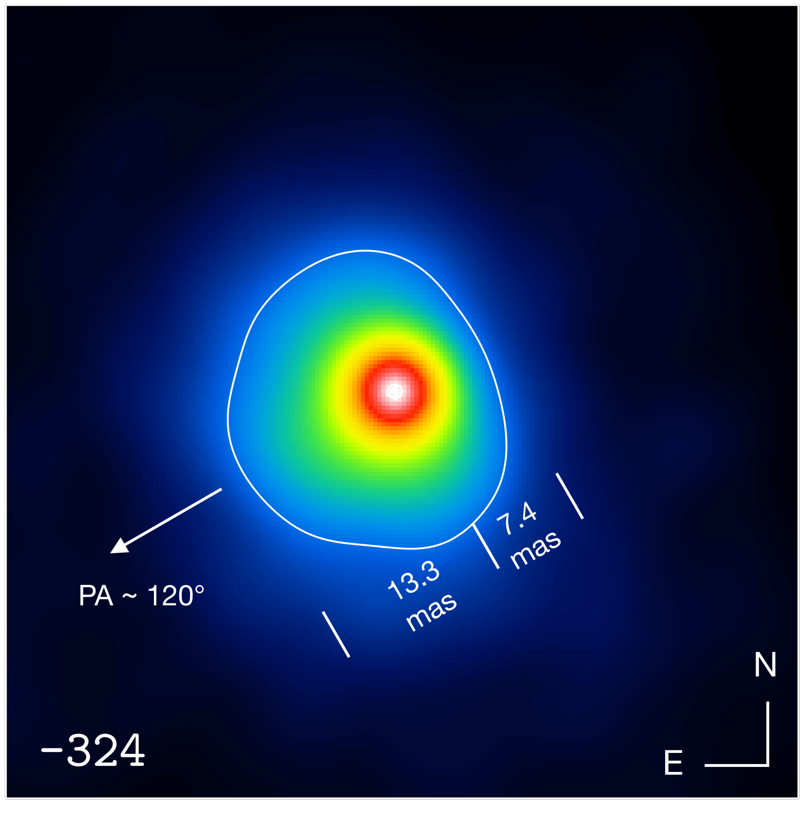
(Hofmann + 2021)

MATISSE - IRBis
Image Reconstruction

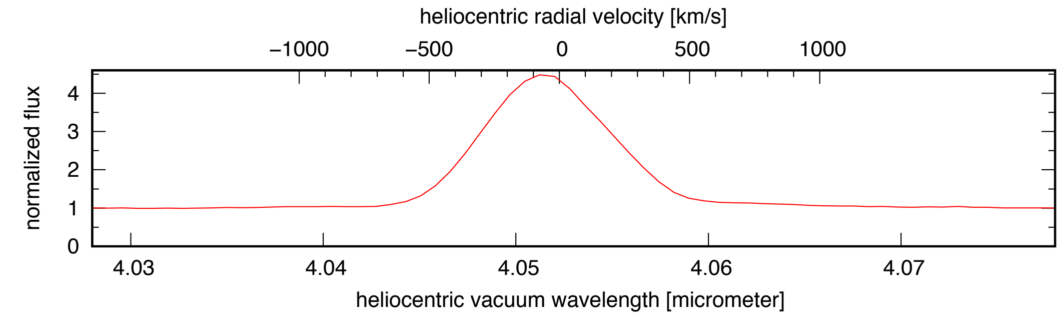
η Carinae

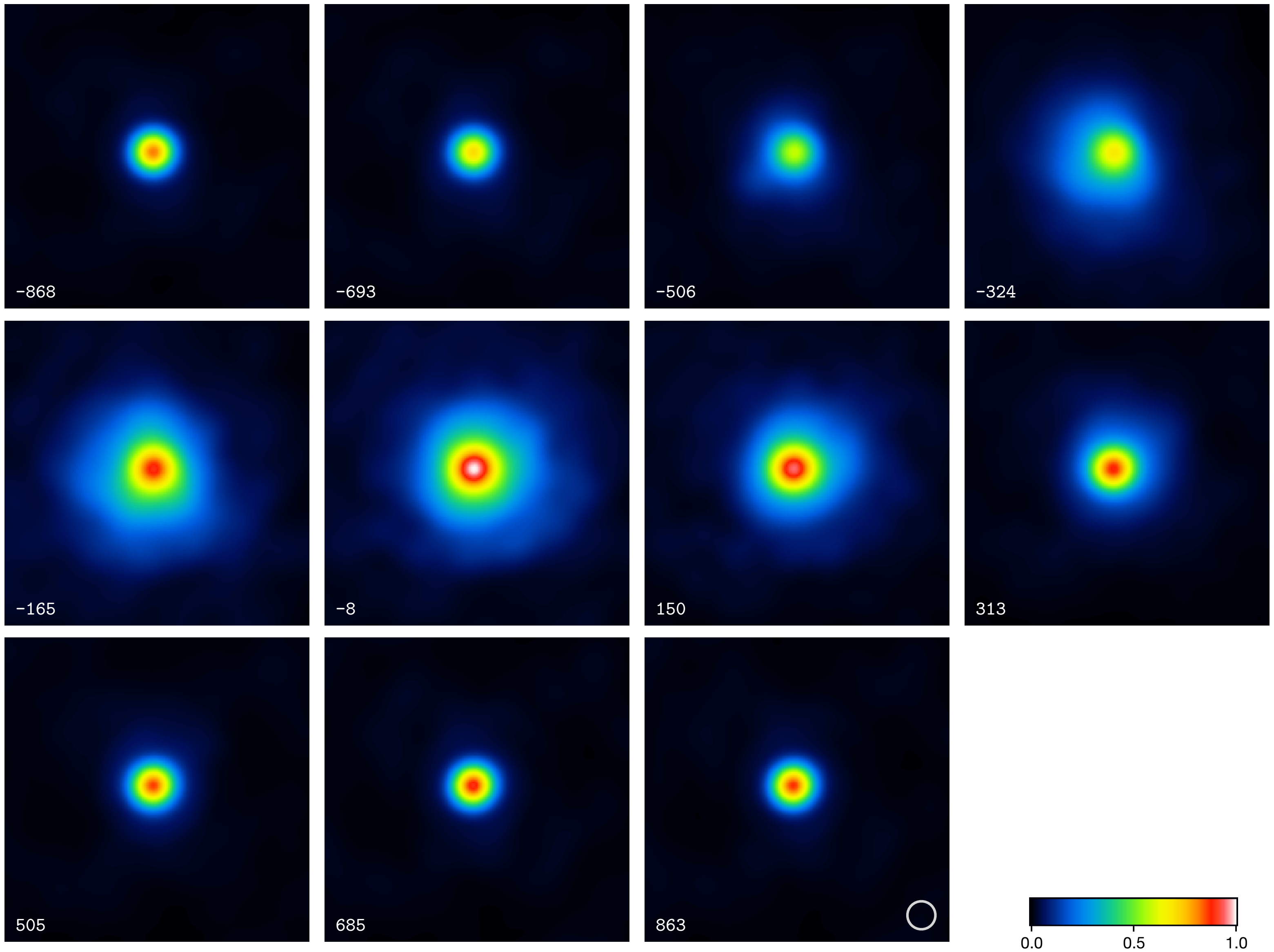
MATISSE imaging of η Car’s primary star stellar wind distorted by the secondary star wind (Brα line; 4.052 mu) at 11 different radial velocities.

- IRBis Fourier phase reconstructions (ATs; R ~ 950; 60 mas FOV)
- Fourier phases are derived from the differential phases



138 au (60 mas)





(Weigelt + 2021)

Basic principles of IRBis

(detailed info to IRBis in the paper: Hofmann, K.-H., Weigelt, G., & Schertl, D. 2014, A&A, 565, A48)

Basic principle: Minimization of a cost function J containing

a) a **data part Q** (built by measured closure phases and squared visibilities from the target) and

b) a **regularization part H** (enforcing positivity, smoothness, etc. in the reconstruction)

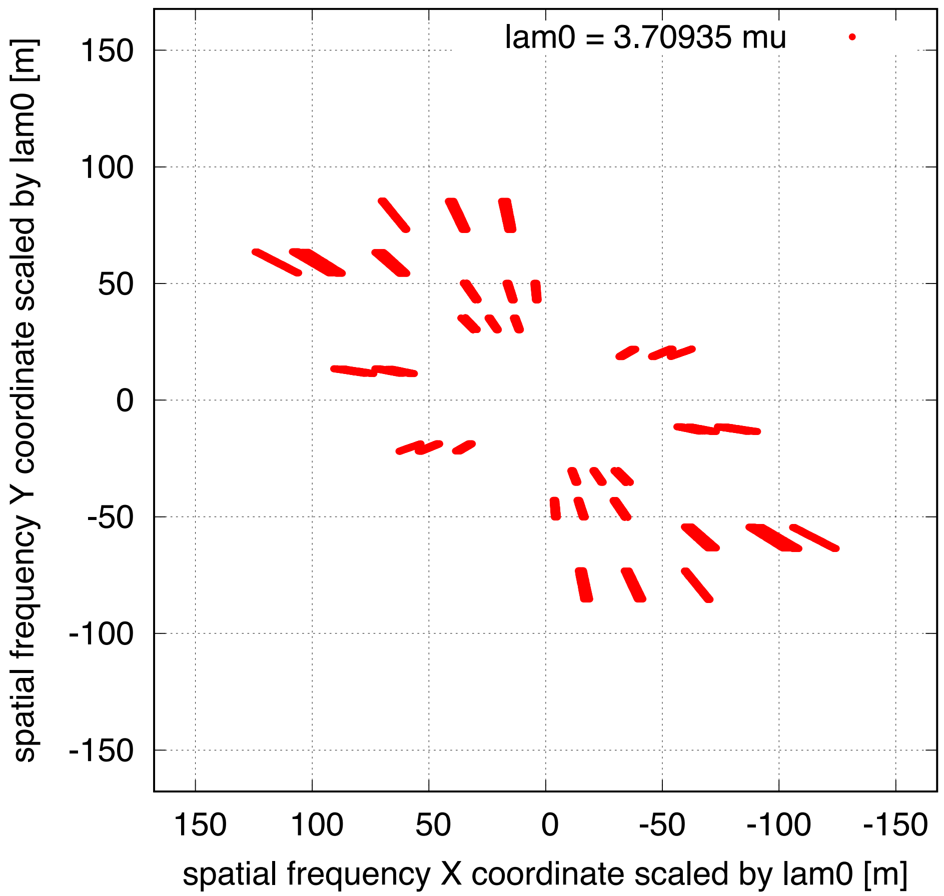
regularization is important in imaging with optical/infrared interferometers because of the sparse uv coverage

$$J[ok(x)] := Q[ok(x)] + \mu \cdot H[ok(x)]$$

- hyper parameter μ defines the strength of the regularization
- Q and H are functions of the iterated image ok(x)

sparse uv coverage
spatial frequency uv coverage obtained from NGC 1068 with the 4 UTs (VLT-8m telescopes on Paranal)

uv points are smeared out because of grey imaging in the spectral range of 3.4 - 4.0 mu



Basic principles of IRBis - data part Q

Data part Q[*ok*(*x*)] of the cost function is the χ^2 function built by the

input data:

- Squared visibilities (power spectra) and the closure phases (phase of the complex bispectrum)
- Some algorithms, e.g. IRBis, use the bispectrum $O^{(3)}(\mathbf{f_u}, \mathbf{f_v})$ built by the measured

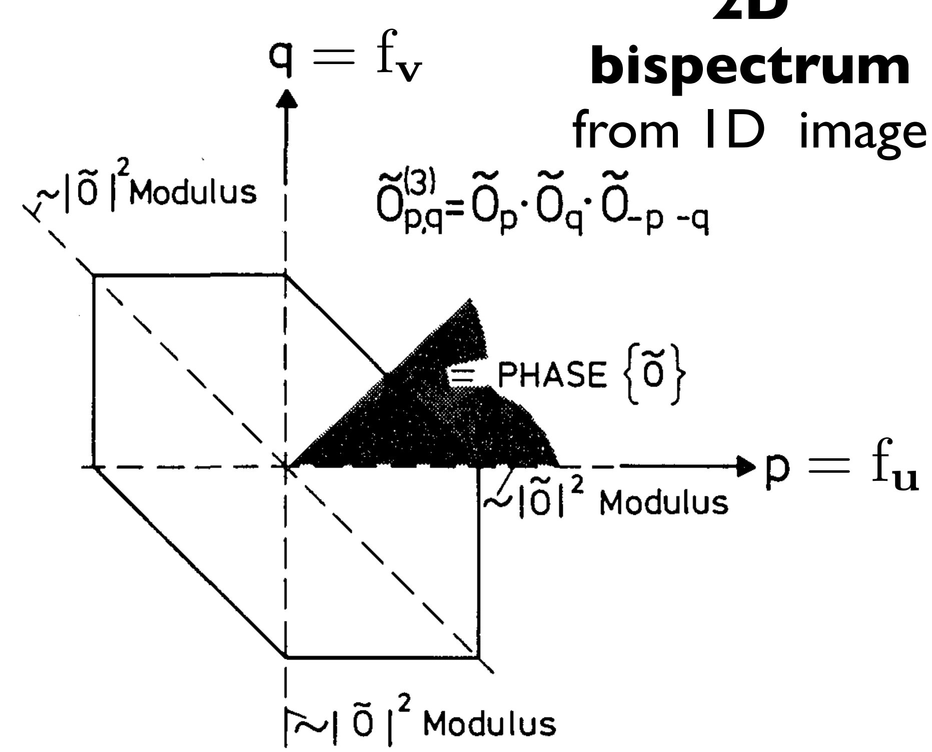
squared visibilities $V^2(\mathbf{f_u})$ and the measured closure phases $\beta(\mathbf{f_u}, \mathbf{f_v})$:

$$O^{(3)}(\mathbf{f_u}, \mathbf{f_v}) = O(\mathbf{f_u})\, O(\mathbf{f_v})\, O^*(\mathbf{f_u} + \mathbf{f_v})$$
$$= \sqrt{V^2(\mathbf{f_u})\, V^2(\mathbf{f_v})\, V^2(\mathbf{f_u} + \mathbf{f_v})}\ \exp\{\mathrm{i}\, \beta(\mathbf{f_u}, \mathbf{f_v})\}$$

- bispectrum also contains the squared visibilities $V^2(\mathbf{f_u})$ on the three axis in the bispectrum plane, e.g. if $\mathbf{f_v} = 0$

- bispectrum of 2-dimensional image is 4-dimensional

2D bispectrum
from 1D image



- the error of the bispectrum built is calculated from the errors of $V^2(\mathbf{f_u})$ and $\beta(\mathbf{f_u}, \mathbf{f_v})$
- this bispectrum is not complete because of the sparsity of the uv coverage.

Basic principles of IRBis - data part Q

Data part *Q[ok(x)]* of the cost function uses this χ^2 function:

$$Q[o_k(\mathbf{x})] := \int_{\mathbf{f_u},\mathbf{f_v} \in \mathrm{M}} \frac{w_d(\mathbf{f_u},\mathbf{f_v})}{\sigma^2(\mathbf{f_u},\mathbf{f_v})} \cdot |\gamma_0 \, O_k^{(3)}(\mathbf{f_u},\mathbf{f_v}) - O^{(3)}(\mathbf{f_u},\mathbf{f_v})|^2 \, d\mathbf{f_u} \, d\mathbf{f_v}$$

measured bispectrum

- $o_k(\mathbf{x})$: actual iterated image, and $\mathbf{x}$ is a 2D image space vector
- $O_k^{(3)}(\mathbf{f_u},\mathbf{f_v})$: bispectrum of the iterated image
- $w_d(\mathbf{f_u},\mathbf{f_v})$: weight to compensate for the unequal distribution of the uv points (it is proportional to the inverse of the uv point density)
- $\sigma(\mathbf{f_u},\mathbf{f_v})$: errors of the measured bispectrum
- γ_0 : scaling factor to minimize the value of Q during each iteration step, i.e. $o_k(\mathbf{x})$ will be not normalised to integral = 1;
- $\mathbf{f_u},\mathbf{f_v} \in \mathrm{M}$: spatial frequencies of the measured uv points: the amount of all measured bispectrum elements

Image reconstruction $\longrightarrow$ Minimization of $Q[o_k(\mathbf{x})]$

Basic principles of IRBis - data part Q

There are **three different data parts Q** in IRBis:

- the χ^2 function built by the bispectrum (see last slide; cost function 1)
- the χ^2 function built by bispectrum phasors and bispectrum moduli (cost function 2)

$$Q_2[o_k(\mathbf{x})] := \int\limits_{\mathbf{f_1},\mathbf{f_2}\in \mathrm{M}} \Big\{ \{ w_\beta(\mathbf{f_1},\mathbf{f_2}) \cdot |\gamma_0 \, \exp[i\,\beta_k(\mathbf{f_1},\mathbf{f_2})] - \exp[i\,\beta(\mathbf{f_1},\mathbf{f_2})]|^2 + $$
$$+ w_m(\mathbf{f_1},\mathbf{f_2}) \cdot f_0 \cdot |\gamma_0\, |O_k^{(3)}(\mathbf{f_1},\mathbf{f_2})| - |O^{(3)}(\mathbf{f_1},\mathbf{f_2})||^2 \} \Big\} \, d\mathbf{f_1}\, d\mathbf{f_2}.$$

- the χ^2 function built by bispectrum phasors and squared visibilities (cost function 3)

$$Q_3[o_k(\mathbf{x})] := \int\limits_{\mathbf{f_1},\mathbf{f_2}\in \mathrm{M}} \frac{w_d(\mathbf{f_1},\mathbf{f_2})}{\sigma_\beta^2(\mathbf{f_1},\mathbf{f_2})} \cdot |\gamma_0 \, \exp[i\,\beta_k(\mathbf{f_1},\mathbf{f_2})] - \exp[i\,\beta(\mathbf{f_1},\mathbf{f_2})]|^2 \, d\mathbf{f_1}\, d\mathbf{f_2} +$$
$$+ f_0 \cdot \int\limits_{\mathbf{f}\in \mathrm{M_p}} \frac{w_d^p(\mathbf{f})}{\sigma_p^2(\mathbf{f})} \cdot |\gamma_0\, |O_k(\mathbf{f})|^2 - |O(\mathbf{f})|^2|^2 \, d\mathbf{f}.$$

because of the use of bispectrum phasors, Q2 and Q3 give higher weight to the closure phases !

Basic principles of IRBis - regularization part

Because of the sparse uv coverage, the noise in the data and the non-linearity of the data

→ many local minima of the χ^2 function exist, and therefore many solutions does exist which could fit the data within the error bars

→ the algorithm has to be helped to find the right solution (= absolut minimum of χ^2 fct.) by introducing prior information about the target:

- prior info is to force positivity of the reconstruction — all observed astronomical targets are intensity distributions → have positive values; this is introduced by the minimization routines.

- prior info is the knowledge about the extent of the target: the restriction of the reconstruction area avoids spikes in the Fourier plane between the observed uv points: small structures → smooth Fourier transform. the reconstruction region can be restricted by a) restricting the FOV to be reconstructed and b) by a binary mask in image space

- prior info is also, for example, smoothness of the target: most targets don’t have a „noisy structure“ but mostly a smooth one

Basic principles of IRBis - regularization part

- prior info is introduced by adding a weighted regularization term $H[o_k(\mathbf{x})]$ to the data constraint term $Q[o_k(\mathbf{x})]$

- the algorithm tries to minimize the cost function

$$J[o_k(\mathbf{x})] := Q[o_k(\mathbf{x})] + \mu \cdot H[o_k(\mathbf{x})] \quad \text{(cost function)}$$

μ : regularization parameter (or hyper parameter) defining the strength of the influence of $H[o_k(\mathbf{x})]$

- the goal of the regularization functions is to select that image $o_k(\mathbf{x})$

out of the pool of all images with $\chi^2 \approx 1$ (many could exist because of sparse uv coverage) which has value of $H[o_k(\mathbf{x})]$ that is closest to the minimum of H and Q

for example: smoother images are preferred against spiky images

Basic principles of IRBis - regularization part

Example of an regularization function in IRBis:

3. „Pixel difference“ quadratic regularization function

$$H[o_k(\mathbf{x})] := \int \frac{[|o_k(\mathbf{x}) - o_k(\mathbf{x}+\Delta\mathbf{x})|^2 + |o_k(\mathbf{x}) - o_k(\mathbf{x}+\Delta\mathbf{y})|^2]}{prior(\mathbf{x})} \, d\mathbf{x}$$

- this function enforces smoothness, since it has a minimum value for very small pixel intensity differences

$prior(\mathbf{x})$ can be: a) an estimate of the target, for example, a Gaussian fitted to the visibilities; it gives the algorithm the info about the extent of the target

b) a constant, if the target extension is not well known

$o_k(\mathbf{x})$: actual reconstruction normalised to $\int o_k(\mathbf{x}) d\mathbf{x} = 1$

a) prior(x) = constant:
minimum of H if (1) the reconstructed object located anywhere is smooth and if (2) the background around the object is smooth, too

b) prior(x) = space-limited function like a small Gaussian:
minimum of H if the reconstructed image is at the same position, has the same size as the prior and is smooth, too, and if (2) the background is smooth, but this is not as important as in the case of a constant prior
⟶ the non-constant prior weights down all parts outside the prior!!

Basic principles of IRBis - regularization part

Other regularization functions in IRBis:

1. „Pixel intensity“ quadratic regularization enforcing flatness & smoothness

$$H[o_k(\mathbf{x})] := \int \frac{o_k(\mathbf{x})^2}{prior(\mathbf{x})}\, d\mathbf{x}$$

2. „Maximum entropy“ enforcing smoothness

$$H[o_k(\mathbf{x})] := \int \left\{ o_k(\mathbf{x}) \cdot \log\{\frac{o_k(\mathbf{x})}{prior(\mathbf{x})}\} - o_k(\mathbf{x}) + prior(\mathbf{x}) \right\} d\mathbf{x}$$

Minimization algorithms in IRBis

- in IRBis and many image reconstruction algorithms, the minimisation of the cost function $J[o_k(\mathbf{x})]$ is performed by large-scale, bound-constrained nonlinear optimization algorithms using the gradient of the cost function to find the solution

  - large-scale : because of the huge number of image pixels in the reconstruction; $o_k(\mathbf{x})$ is considered as a NxN dimensional vector with the pixel intensities as coordinate values (NxN number of pixel of the reconstruction)

  - bound-constrained: because of the positivity of the pixel intensities = coordinate values $\longrightarrow$ coordinate values >0

  - nonlinear: because of the non-linearity of the cost function (bispectrum & power spectrum)

- nearly all these optimization algorithms search for the position of the local minimum close by, and not the desired absolute minimum!

  this means that the start image should already be close/similar to the true image.

Minimization algorithms in IRBis

- IRBis uses the nonlinear optimization algorithms ASA_CG (Hager & Zhang 2006) and L-BFGS-B (Byrd, Lu & Nocedal 1995):

ASA_CG is a conjugate gradient based large-scale, bound-constrained, nonlinear optimization algorithm

L-BFGS-B is limited-memory, large-scale, bound-constrained, and very effective optimization algorithm

- Inputs to ASA_CG and L-BFGS-B are

1. the actual position vector $(o_k(\mathbf{x_1}), o_k(\mathbf{x_2}), ..., o_k(\mathbf{x_j}), ..., o_k(\mathbf{x_M}))$

2. the value of the cost function $J[o_k(\mathbf{x_1}), o_k(\mathbf{x_2}), ..., o_k(\mathbf{x_j}), ..., o_k(\mathbf{x_M})]$

3. the gradient of the cost function at the actual position vector

$$\mathrm{grad}[J] = \left(\frac{\partial J[o_k(\mathbf{x})]}{\partial o_k(\mathbf{x_1})}, .., \frac{\partial J[o_k(\mathbf{x})]}{\partial o_k(\mathbf{x_j})}, ..., \frac{\partial J[o_k(\mathbf{x})]}{\partial o_k(\mathbf{x_M})} \right)$$

With these inputs ASA_CG and L-BFGS-B work roughly similar as the „steepest descent method“ for searching the position of the local minimum

Minimization algorithms in IRBis

- the data part Q[ok(x)] of the cost function in the case of the bispectrum (Q1)

$$Q[o_k(\mathbf{x})] := \int\limits_{\mathbf{f_u},\mathbf{f_v}\in \mathrm{M}} \frac{w_d(\mathbf{f_u},\mathbf{f_v})}{\sigma^2(\mathbf{f_u},\mathbf{f_v})} \cdot |\gamma_0\, O_k^{(3)}(\mathbf{f_u},\mathbf{f_v}) - O^{(3)}(\mathbf{f_u},\mathbf{f_v})|^2\, d\mathbf{f_u}\, d\mathbf{f_v}$$

- its gradient:

$$\frac{\partial Q[o_k(\mathbf{x})]}{\partial o_k(\mathbf{x_j})} = \int\limits_{\mathbf{f_u},\mathbf{f_v}\in \mathrm{M}} 6 \cdot \frac{w_d(\mathbf{f_u},\mathbf{f_v})}{\sigma^2(\mathbf{f_u},\mathbf{f_v})} \cdot [\gamma_0\, O_k^{(3)}(\mathbf{f_u},\mathbf{f_v}) - O^{(3)}(\mathbf{f_u},\mathbf{f_v})]^* \times$$

$$\times O_k(\mathbf{f_u})\, O_k(\mathbf{f_v}) \cdot \exp\left[+2\pi\, \mathrm{i}\, (\mathbf{f_u}+\mathbf{f_v}) \cdot \mathbf{x_j}\right] d\mathbf{f_u}\, d\mathbf{f_v}$$

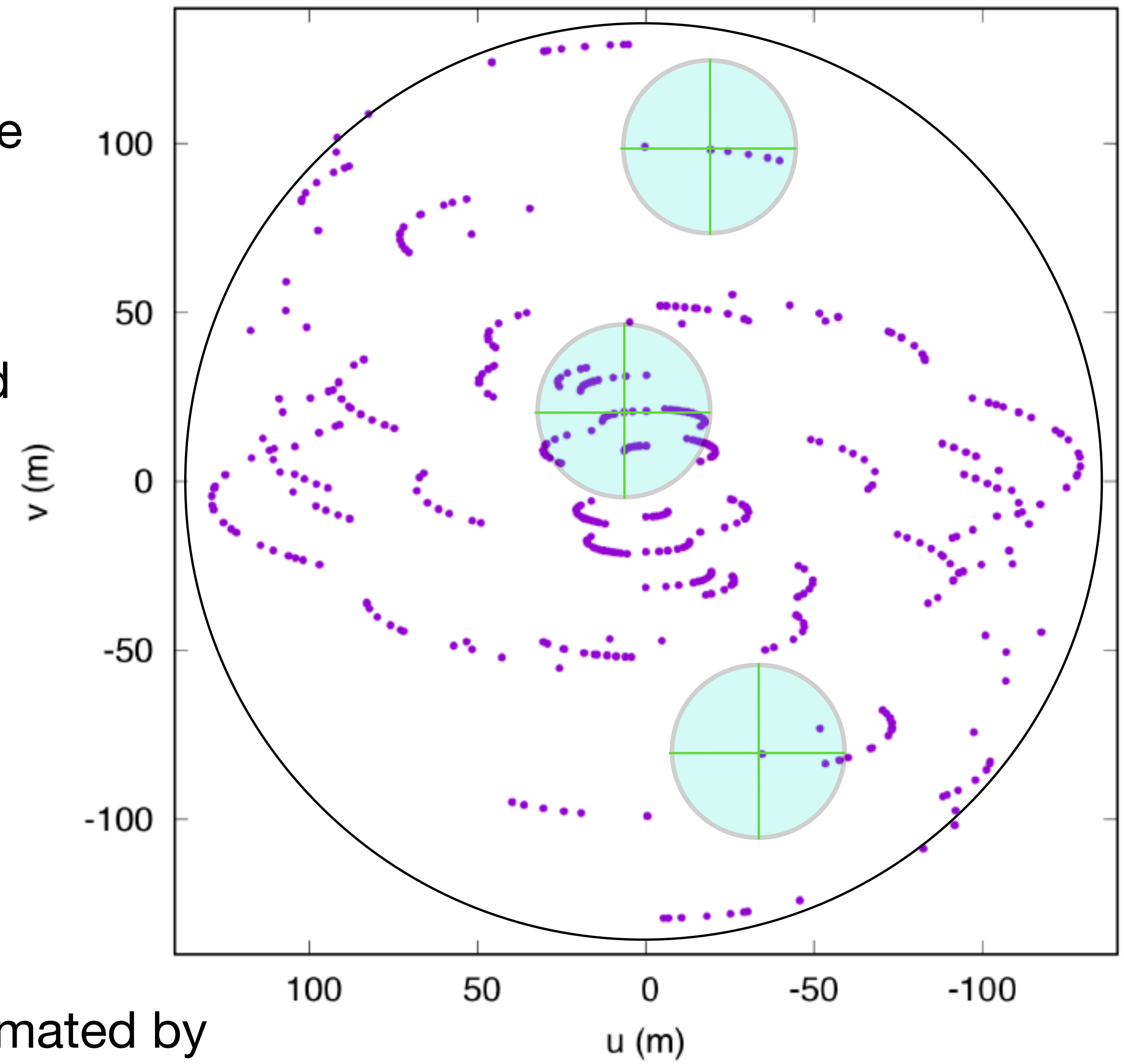
- the other data parts of IRBis consisting of bispectrum phasors have much more complicated gradients

Additional features in IRBis - uv density weight

MOTIVATION:
data in different parts of the uv coverage should have the same influence to the reconstructed image

—> best is an equal distribution of the uv points
BUT
—> in reality are the uv points unequally distributed

A SOLUTION:
give measures (CP, V) at uv points a) in high density clusters a smaller weight and b) in low density regions a higher weight



REALIZATION in IRBis:

- around each uv point the uv point density is estimated by
  - counting all neighbouring uv points within a circle around the „master“ uv point
  - the radius of this circle is the smallest distance between two neighbouring uv points, if all uv points are equally distributed within a circle with the maximum baseline length as radius.
- the weight at the visibility at one uv point is ~ the inverse uv point density at this uv point
  the weight in IRBis is the inverse uv point density with a certain power (usually 0.5)
- the weight of the bispectrum elements (CPs) is playing the same game in 4D bispectrum space

Additional features in IRBis - image quality parameter qrec

- the quality of the reconstruction can be evaluated by its χ^2 values:

a) reduced χ^2 of the squared visibilities : $$\chi^2_{V^2} = \frac{1}{N_{V^2}} \int_{\mathbf{f} \in \mathrm{M}} \left| \frac{V_k^2(\mathbf{f}) - V^2(\mathbf{f})}{\sigma_{V^2}(\mathbf{f})} \right|^2 d\mathbf{f}$$

b) reduced χ^2 of the closure phases : $$\chi^2_{CP} = \frac{1}{N_{CP}} \int_{\mathbf{f_u}, \mathbf{f_v} \in \mathrm{M}} \left| \frac{\beta_k(\mathbf{f_u}, \mathbf{f_v}) - \beta(\mathbf{f_u}, \mathbf{f_v})}{\sigma_{\beta(\mathbf{f_u}, \mathbf{f_v})}} \right|^2 d\mathbf{f_u} d\mathbf{f_v}$$

N_{V^2}, N_{CP} : number of measured elements

Good fit to the data : reduced $\chi^2 \approx 1$, because the deviations between measured and fitted data lies within the error bars.

Additional features in IRBis - image quality parameter qrec

- Additional measure of the reconstruction quality is the so called residual ratio:

a) residual ratio of the squared visibilities : $\rho\rho_{V^2} := \dfrac{\int\limits_{\mathbf{f}\in M_+} [V_k^2(\mathbf{f}) - V^2(\mathbf{f})]/\sigma_{V^2(\mathbf{f})}\, d\mathbf{f}}{\int\limits_{\mathbf{f}\in M_-} [V_k^2(\mathbf{f}) - V^2(\mathbf{f})]/\sigma_{V^2(\mathbf{f})}\, d\mathbf{f}}$

b) residual ratio of the closure phases : $\rho\rho_{CP} := \dfrac{\int\limits_{\mathbf{f_u},\mathbf{f_v}\in M_+} [\beta_k(\mathbf{f_u},\mathbf{f_v}) - \beta(\mathbf{f_u},\mathbf{f_v})]/\sigma_{\beta(\mathbf{f_u},\mathbf{f_v})}\, d\mathbf{f_u} d\mathbf{f_v}}{\int\limits_{\mathbf{f_u},\mathbf{f_v}\in M_-} [\beta_k(\mathbf{f_u},\mathbf{f_v}) - \beta(\mathbf{f_u},\mathbf{f_v})]/\sigma_{\beta(\mathbf{f_u},\mathbf{f_v})}\, d\mathbf{f_u} d\mathbf{f_v}}$

M_+, M_- define the elements with positive and negative residuals, respectively

Good fit to the data : $\rho\rho \approx 1$ because in this case the fit is balanced between negative and positive residuals

- A global measure of the reconstruction quality

$\longrightarrow$ $\mathrm{q_{rec}} := 1/4 \cdot [|\chi^2_{V^2} - 1| + |\rho\rho_{V^2} - 1| + |\chi^2_{CP} - 1| + |\rho\rho_{CP} - 1|]$

Good reconstruction with q_{rec} close to zero.
Note: q_{rec} is defined to recognize very well if χ^2 and $\rho\rho$ >>0, but NOT if χ^2, $\rho\rho$ <1!!

Solution: if χ^2, $\rho\rho$ <1, then $1/\chi^2$ instead of χ^2 is used, same for $\rho\rho$
$\longrightarrow$ large q_{rec} values for χ^2 and $\rho\rho$ <<1

Scan of image reconstruction parameters in IRBis

- the varying image reconstruction parameters are:

1. size of the binary circular mask in image space:
   only within the mask, reconstructed intensities >0 are allowed;
   the mask enforces the algorithm to reconstruct images with smoother Fourier spectra
   (reason: small structures in image plane produce large, smooth structures in Fourier plane)

2. strength of the regularization parameter μ :
   balancing the influence of the measured data (χ^2 term) and the prior data (regularization term)
   to the reconstruction: high μ value = strong influence of the prior term

- to find a good reconstruction these 2 parameters are varied within two loops:

outer loop: ***n*** (~6) different mask radii (usually mask radii increase within this loop*)
inner loop: ***m*** (~12) different values of the regularisation parameter;
for each image mask, m reconstruction runs with m μ values are performed
(usually, the μ values decrease from one to the next run**)

* increasing mask radii: stronger regularization at the beginning helps to come closer to the absolute minimum of the cost function $J[o_k(\mathbf{x})]$ and helps to find the correct image

** decreasing μ values: stronger ....

Scan of image reconstruction parameters in IRBis

- the ***n*** x ***m*** reconstructions obtained will be evaluated roughly according to their quality and sorted according to decreasing quality (the best reconstruction and a few of the next best reconstructions will be stored)

- the image quality is evaluated with the quality measure „ $\mathrm{q_{rec}}$" discussed a few slides ago

- start & prior for the first run are the images defined at the beginning of the reconstruction session: a model image, e.g. a Gaussian, Uniform disk or a more physical model.

  start & prior image for the current reconstruction run are the reconstruction of the run before: this was found out by many experiments to be a good choice and this is the default setting in IRBis

- the next image mask radius is obtained by adding a radius step to the actual radius:

$$R(j+1) = R(j) + \Delta R$$

- the next regularization parameter μ is obtained by multiplying the actual μ by a factor:

$$\mu(j+1) = \mu(j) \cdot \mathrm{factor}$$

- it is also useful to run through the outer and inner loop without changing the size of the image mask, these optimization routines yield better results after several restarts

Enough data? uv coverage and target size

- The number of independent uv points ≥ the number of resolution elements within the image size
  size of a resolution element ~λ/B_{max} ;usually, more than ~20 uv points are required!

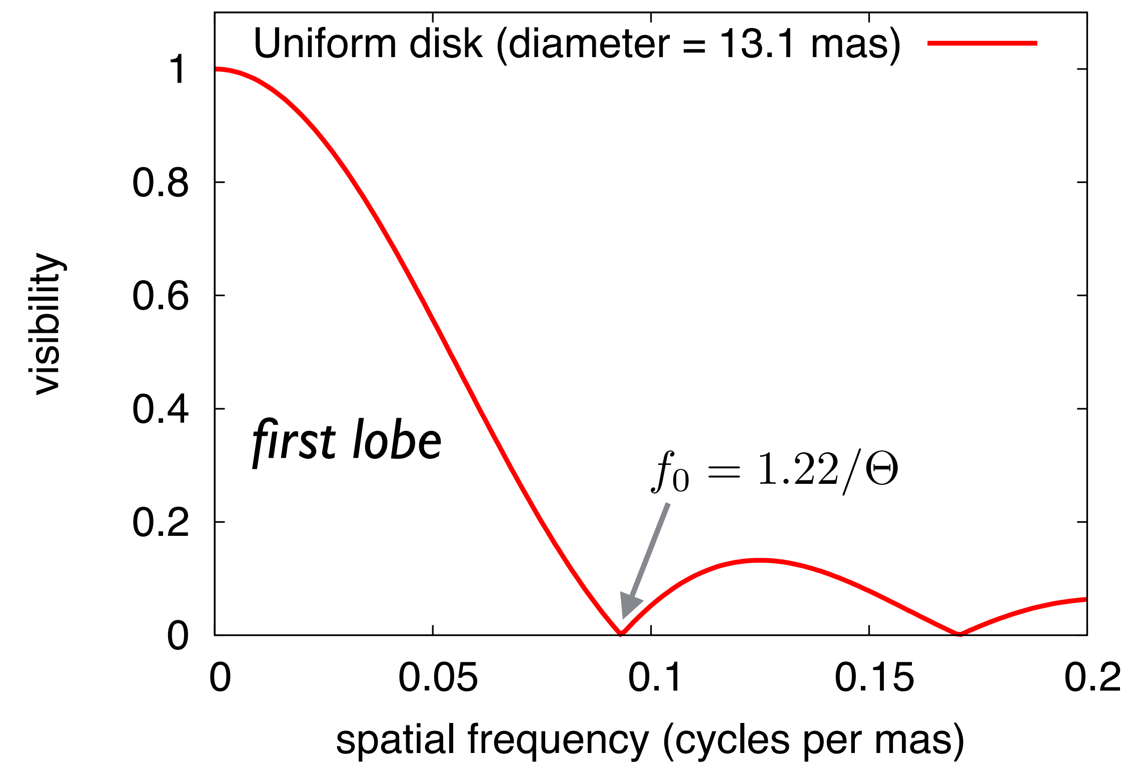
- Holes in the uv coverage will give artefacts in the reconstructed image

- Shortest baseline $B_{\min}$ should be well inside the first lobe of the target visibility (visibility ~0.3-0.5):
  the first lobe contains information about the extension of the target <== important for imaging

Example:
- a stellar disk is roughly a uniform disk (UD)
- the first zero of the visibility of a UD with angular diameter Θ lies at spatial frequency $f_0 = 1.22/\Theta$
- to get visibilities within the first lobe, i.e.

$$f < f_0 \quad \rightarrow \quad \frac{B_{\min}}{\lambda} < \frac{1.22}{\Theta}$$

$$\rightarrow \quad \Theta < 1.22 \cdot \frac{\lambda}{B_{\min}} \approx \boxed{\frac{1.22}{2} \cdot \frac{\lambda}{B_{\min}} =: \Theta_{\max}}$$



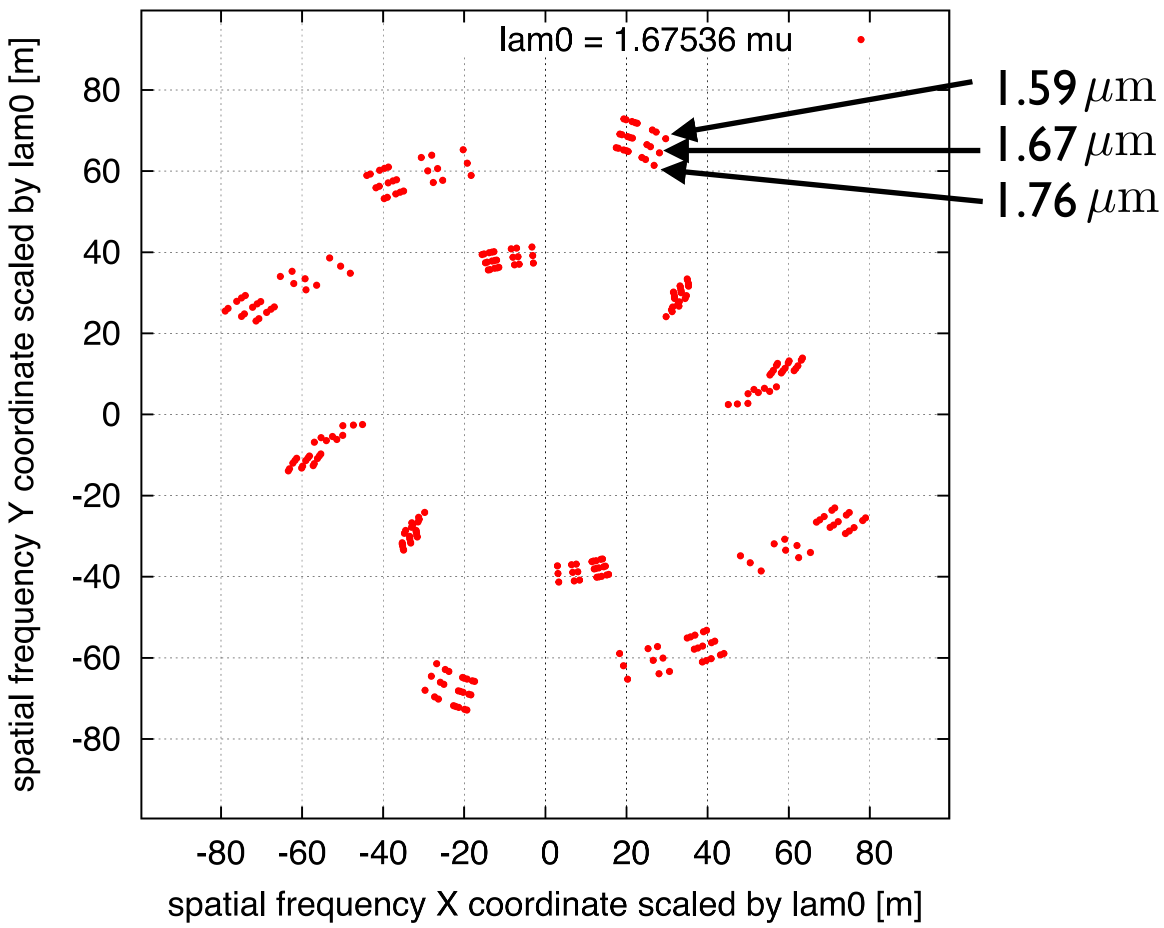
due to the smallest baseline $B_{\min}$ of the array: the diameter of the target should be not larger than $\Theta_{\max}$.

→ usable interferometric FOV ~ λ/B_{min}

Enough data? different spectral channels

- number of uv points can be increased by using different spectral channels, if the target is not wavelength-dependent

spatial frequencies $\mathbf{B}/\lambda$



Optimal FOV and pixel grid for image reconstruction

- to avoid cutting artefacts, the field-of-view (FOV) should be ~2-4x the target size
- the highest spatial frequency in the Fourier plane of a NxN pixel grid is N/2

  with the relation $\boxed{f_{pixel} = f[1/mas] \cdot FOV[mas]}$ (valid for discrete Fourier transf.)

  we get the largest possible FOV for the NxN pixel grid

$$\frac{N}{2} = \frac{B_{max}}{\lambda} \cdot FOV_{max}$$

$$\rightarrow FOV_{max} = \frac{N}{2} \cdot \frac{\lambda}{B_{max}}$$

- the best reconstructions are obtained when the highest frequency lies at ~N/4

$$FOV = \frac{N}{4} \cdot \frac{\lambda}{B_{max}}$$

→ **optimal FOV and pixel grid**

Fourier plane NxN pixel grid

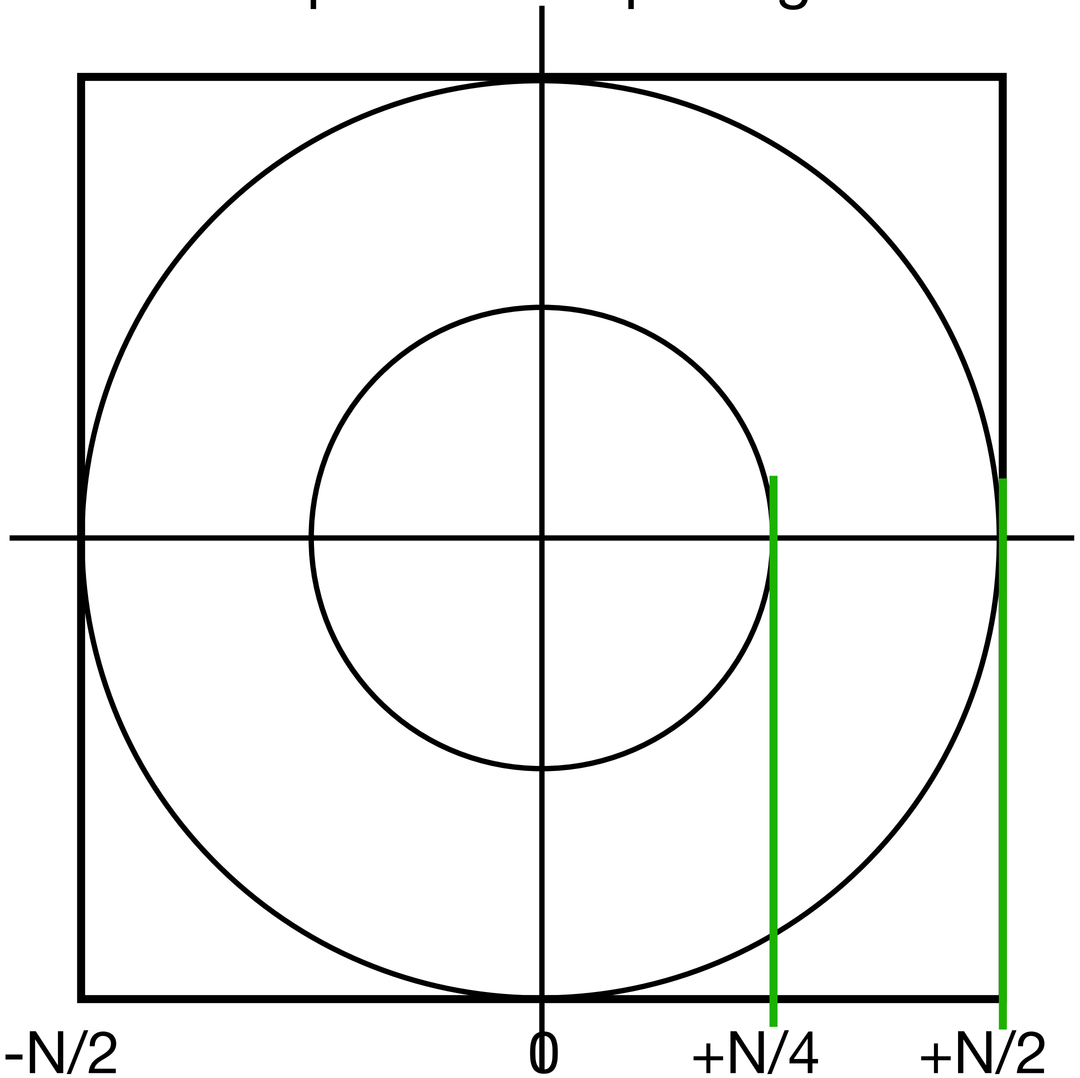


Image reconstruction parameter in IRBis

- Interferometric data are stored in files with **OIFITS** format (standard in optical/infrared long-baseline interferometry)

- Data selection parameter: observing target and wavelength range

large wavelength range → better uv coverage → better reconstruction!?
large wavelength range → wavelength dependence of the target

small wavelength range → worse uv coverage → worse reconstruction!?
small wavelength range → less wavelength dependence of target

- Size of the **FOV** and size **NxN** of the pixel grid to be reconstructed

- Size of the **binary image mask**

- **Regularization function** and regularization parameter μ

- **Start image** & prior image

Handling of IRBis - the image reconstruction script

- IRBis (= **I**mage **R**econstruction using the **Bis**pectrum) is part of the ESO/VLTI/MATISSE data reduction pipeline

- IRBis is coded in C and its technical name is „mat_cal_imarec“

- the man page of IRBis can be called by: „esorex —man-page mat_cal_imarec“

- the basic reconstruction run can be called by:

  „esorex —log-dir=[directory of the logfile] —output-dir=[directory of the reconstruction results] \
  mat_cal_imarec [mat_cal_imarec options] [sof = ASCII file containing the input data]“

- for easier handling of an image reconstruction session with IRBis, the shell scripts „mat_cal_imarec_all.csh“ and „mat_cal_imarec_all.2.csh“ are provided

  the scripts, a short tutorial and test data can be downloaded from

  https://gitlab.oca.eu/MATISSE/tools/-/tree/master/imarec

Handling of IRBis - the image reconstruction script

the scripts perform two actions:

1) **Estimating** the size of the target by fitting a simple geometrical model (Gaussian, Uniform disk, etc.) to the measured visibilities and **proposing** the optimal size of of the FOV and pixel grid for image reconstruction from these data.

2) The **reconstruction run** and the presentation of the **results**.

the scripts are called, for example, by

```
$SCRIPTS/mat_cal_imarec_all.2.csh   mat_cal_imarec_all.2.par
```

- $SCRIPTS is the path to the directory containing the scripts
- a template parameter file „mat_cal_imarec_all.2.par“ can be copied from $SCRIPTS/

Handling of IRBis - the image reconstruction script

1. action of the scripts:

1) editing the parameter file:

- the oifits data: set data = (path/data1 path/data2)
- wavelength range (in μm = mu): set lambda list = (0.1 20.0) (to see all spectral channel in the first run)
- set the switch to parameter estimation: set guess = 1

2) the results after the run are in the directory Parameter.Estimation/

- plots of a) uv coverage (uv.ps), b) wavelength table (wavelength.ps), c) fit of the geometrical models to visibilities (gaussudfdd.ps)

- data are in ASCII file data.parameter:

```
A) Information about the data:
   - minimum and maximum wavelength (in m)      : 1.71554e-06 1.7373e-06
   - minimum and maximum baseline length (in m): 8.07256 133.912
   - resolution lambda / B_max (in mas)         :  2.65921
   - number of measured squared visibilities    : 901
   - number of measured closure phases          : 516
   - number of measured Fourier phases          : 0
   - average SNR of measured squared visibilities: 25.9281
   - average error of the closure phase (in deg) : 2.54835
   - average error of the Fourier phase (in deg) : 0
 --> closure phase imaging is possible only!
```

Handling of IRBis - the image reconstruction script

1. action of the scripts:

2) the results after the run are in the directory Parameter.Estimation/
   - data are in ASCII file data.parameter.txt:

```
B) Estimated target size by fitting the V^2 data:
   * (2) Gaussian              --> FWHM = 7.518 mas (red. Chi^2 = 26.879,)
   * (3) Uniform disk          --> diameter = 10.119 mas (red. Chi^2 = 7013.583,)
   * (4) Fully darkened disk --> diameter = 11.539 mas  (red. Chi^2 = 481.906,)
   * (5) Lorentzian function --> FWHM = 8.209 mas (red. Chi^2 = 720.159,)

C) Recommended size of the angular FOV and the size of the NxN pixel grid for the image reconstruction run:
    - the optimal FOV[mas] should be about ~ 2 to 4 times the size of the target (see estimations in B))
    - the optimal FOV[mas] is covered by a NxN pixel grid where in the Fourier plane all uv points
      are lying within the cut-off frequency f_pixel=N/4, i.e. where all uv points have distances to
      the Fourier center which are smaller or equal to N/4 (to avoid aliasing)
    - the maximum target size derived from the present uv coverage for reliable image reconstruction
      should be not larger than Theta_max = (1.22/2)*lambda_max/B_min = 27.0781 mas
      (in order to have a few data points within the first loop of the Fourier transform of the target, e.g. a Uniform Disk.)
      --> therefore the max. FOV[mas] for image reconstruction should be ~ 2 to 4 x Theta_max, i.e. between 54.1565 and 108.313 ma

==> a collection of NxN pixel grids and their FOVs[mas] providing that the uv point with the largest distance to the Fourier cente
    has a distance of N/4 pixels to it:
      * 16x16 pixels --> FOV = 10.5698 mas
      * 32x32 pixels --> FOV = 21.1396 mas
      * 64x64 pixels --> FOV = 42.2792 mas
      * 128x128 pixels --> FOV = 84.5584 mas
      * 256x256 pixels --> FOV = 169.117 mas
      * 512x512 pixels --> FOV = 338.234 mas
```

these **pixel grid / FOV** sizes are for the case, that the length of the largest spatial frequency is N/4 in the Fourier transform of the NxN pixel grid !

Handling of IRBis - the image reconstruction script

2. action of the scripts:

1) flow chart of the 2. action:

```
FOR EACH "regularization function" $regFuncs DO
  FOR EACH "start value of the hyperparameter" $muStarts DO
    # begin of IRBis call
      call IRBis($regFuncs[i], $muStarts[i], ...)
    # end of IRBis call --> reconstruction with the best (=smallest) qrec value
    # Maintain an external list of all best reconstructions returned from IRBis,
    # and continue to loop over $regFuncs and $muStarts
  ENDFOR ($muStarts)
ENDFOR ($regFuncs)
```

2) difference between the two scripts:

- mat_cal_imarec_all.csh : loop overall $regFuncs, and for each regularisation function using a list of regularization parameter start values mu in $muStarts
  —> IRBis call with $regFunc[i] and $muStarts[i]

- mat_cal_imarec_all.2.csh : loop overall $regFuncs, but using start values calculated as $muStart(next) = $muStart(current) * $muFactor0, starting with $muStart0
  —> IRBis call with $regFunc[i] and $muStarts[i]

This loop stops, if the returned qrec value starts to increase

Handling of IRBis - the image reconstruction script

2. action of the scripts:

3) editing the parameter file:

- specify the optimisation engine: set engine = 2
  1 = ASA_CG; 2 = L-BFGS-B (this is faster & often used in the case of many pixels)
- specify if bispectrum or complex visibilities should be used : set algoMode = 1
  1 = use bispectrum only; 2 = use complex visibility only; 3 = use both

- specify the target part of the cost function : set costFunc = 1
  1 = use the bispectrum; 2 = use bispectrum phasors plus bispectrum moduli;
  3 = use bispectrum phasors plus squared visibilities

- specify the FOV of the reconstruction (in milliarcsec = mas) : set fov =

- specify the pixel grid npix x npix of the reconstruction FOV : set npix =

- specify parameter of the binary image mask loop **within IRBis** (**outer loop**):
  * start radius of the mask (in mas) : set oradiusStart =
  * step size for increasing the mask radius (in mas) : set setpSize =
  * number of mask radii to tested : set oradiusNumber = 6

Handling of IRBis - the image reconstruction script

2. action of the scripts:

3) editing the parameter file:

- specify the regularization parameter loop **within IRBis** (**inner loop**):
  * start value of the regularization parameter mu : set muStart =
  * factor to calculate the next regularization parameter : set muFactor = 0.5
  * number of regularization parameter to be tested : set muNumber = 12

- specify regularzation parameter start values **outside IRBis** in the scripts:
  * in script_mat_cal_imarec_all.csh —> list of start values : set muStarts =

  * in script mat_cal_imarec_all.2.csh —> start values are calculated by
    muStart(next) = muStart(current) * muFactor0; starting with muStart = muStart0
    (no more muStart will be tested, if qrec (=image quality measure) starts to increase!)
    ** start value of the regularization parameter start value : set muStart0 =
    ** factor to calculate the next regularization parameter start value : set muFactor0 = 0.5

- specify which **qrec criterium** should be used in the script **mat_cal_imarec_all.2.csh** to terminate the regularization parameter start value test : set qrecmode = 2
  1 = qrec calculated using Chi^2/residual ratios of both, closure phases and visibilities
  2 = ,, ,, ,, ,, ,, of the closure phases only

Handling of IRBis - the image reconstruction script

2. action of the scripts:

3) editing the parameter file:

- specify the regularization functions to be tested : set regFuncs =
- choose the power of the uv density weight : set weightPower = 0.5
- define the type of start object : set startmode = 2
  - 0 = read from file
  - 1 = point source
  - 2 = Gaussian disk
  - 3 = Uniform disk
  - 4 = Fully darkened disk
  - 5 = modified Lorentz function
- size of the start object (diameter / full width half maximum in mas): set startparam =

NOTE: in the first run, the only parameters which have to be changed in the parameter file are:

- the FOV of the reconstruction : $fov
- the pixel grid for the reconstruction FOV : $npix
- the binary mask parameter : $oradiusStart & $stepSize
- start object : $startmode & startparam

These values depend on the target, uv coverage and wavelength!

Handling of IRBis - the image reconstruction script

2. action of the scripts:

4) results after an image reconstruction run

```
 1) The results are stored in result folders named
       BIS.*.Script*.E.1/    ($algoMode=1 : only the bispectrum was used),
       FT.*.Script*.E.1/     ($algoMode=2 : only the complex visibilities were used),
       FTBIS.*.Script*.E.1/ ($algoMode=3 : both bispectrum and complex visibilities were used).
    If later the script is started again, the extention of the folder will automatically increase,
       i.e. -> .2 .3 .. (if the folder name could be identical to another one)
     Script* == Script1 means produced by script "mat_cal_imarec_all.csh",
     Script* == Script2 means produced by script "mat_cal_imarec_all.2.csh".

 2) In the result folder, each subfolder E.1, E.2, .. contains the best reconstruction of one run*
    Contents of the subfolders E.1, E.2 ...:
          - rec_*.fits : contains the "best" reconstruction of each run. This is the direct outcome
                         of the recipe mat_cal_imarec (IRBis).
                         It not only contains the unconvolved and convolved reconstruction, but also
                         other data. See the manual of mat_cal_imarec for more.
            Man page call: % esorex --man-page mat_cal_imarec
          - bestrec.fits, bestrecconv.fits : the "best" reconstruction, unconvolved and convolved.
          - model.fits, modelconv.fits : model images (unconvolved/convolved), in case of simulation.
```

* each run in E.?/ contains the best reconstruction of an IRBis start with the script parameter
$regFuncs[i] (= current chosen regularization function) and
$muStarts[j] (= current value of the hyper parameter start value)

Handling of IRBis - the image reconstruction script

2. action of the scripts:

4) results after an image reconstruction run

```
3) In the result folder, the textfile E.liste contains the Chi^2/ResidualRatio values and the
   image reconstruction parameters (regularisation function (Reg-Fct.), hyperparameter mu, etc...) for
   each IRBis run, i.e. in E.1, E.2, ...
    Each run is listed in one line; the runs are sorted with
     1. increasing image quality parameter qrec (qrecmode=1: phase-visibiliy-qrec, derived from chi^2
        and residual ratios of the squared visibilities and CP's or Fourier phases),
     2. increasing image quality parameter qrec (qrecmode=2: phase-qrec, derived from chi^2 and
        residual ratios of the phases (CPs or Fourier phases) only

4) In the result folder the best reconstruction out of all ones, stored in E.1, E.2, .., is named:
        - bestrec.qrec1.fits (best according to the measure phase+visibility-qrec, i.e. qrecmode=1)
        - bestrec.qrec2.fits (best according to the measure phase-qrec, i.e. qrecmode=2)
   (both are the direct outcome of IRBis (= the fits file rec_*.fits in the subdirectories E.*/).


5) In the result folder there are postscript files which contain all reconstructions (E.1, E.2, ...)
   sorted according to the increasing qrec measure as listed in E.liste:
        a) in folder qrecmode=1/ are recs sorted according to increasing phase+visibility-qrec values.
        b) in folder qrecmode=2/ are reconstructions sorted according to increasing phase-qrec values.
         The postscript files in these two folders are
                 - *.lin.ps (linear display)
                 - *.sqrt.ps (sqrt display)
                 - *.log.ps (log display)
```

Handling of IRBis - summary

```
- the script is called with the parameter file mat_cal_imarec_all.2.par, which has to be copied from $SCRIPTS
  and edited:
        $SCRIPTS/mat_cal_imarec_all.2.csh mat_cal_imarec_all.2.par

- 1. action: estimation of the target size and image reconstrcution parameter
     edit in the parameter file:
        set data        = ("insert the oifits data of the target")
        set lambdaList = ("insert the wave length intervall to be used for image reconstruction")
        set guess       = 1

- main results of 1. action in
  * Parameter.Estimation/data.parameter.txt:
  B) Estimated target size by fitting the V^2 data:
     * (2) Gaussian             --> FWHM = 7.518 mas (red. Chi^2 = 26.879,)
     * (3) Uniform disk         --> diameter = 10.119 mas (red. Chi^2 = 7013.583,)
     * (4) Fully darkened disk --> diameter = 11.539 mas  (red. Chi^2 = 481.906,)
     * (5) Lorentzian function --> FWHM = 8.209 mas (red. Chi^2 = 720.159,)

  C) Recommended size of the angular FOV and the size of the NxN pixel grid for the image reconstruction run:
    ==> a collection of NxN pixel grids and corresponding FOVs[mas]
       with the largest distance to the Fourier center = N/4 pixels (N/2 = highest frequency for a NxN pixel grid)
      * 16x16 pixels --> FOV = 10.5698 mas
      * 32x32 pixels --> FOV = 21.1396 mas
      * 64x64 pixels --> FOV = 42.2792 mas
      * 128x128 pixels --> FOV = 84.5584 mas
      * 256x256 pixels --> FOV = 169.117 mas
```

Handling of IRBis - summary

```
- 2. action: first image reconstruction run
  edit in the parameter file:
  * Switch image reconstruction ON
       set guess = 0
  * Select that NxN pixel grid where the corresponding FOV is ~4x target size, for example, in the case
    of an uniform disk size of ~10 mas, select:
     64x64 pixels --> FOV = 42.2792 mas
      set fov    = 42
      set grid   = 64
       set oradiusStart   = 20  (without limitations due to a binary mask, in order to not overlook details outsi
      set stepSize       = 1.0
  * Select one regularization function
      set regFuncs       = (-3) pixel difference quadratic enforcing smoothness (is a good beginning)
  * Select the fit start object and its size
       set startmode      =  # 2 = gaussian disc, 3 = uniform disc, 4 = fully darkened disk
       startparam         =  #  mode=2 -> FWHM [mas], mode=3 -> diameter [mas], mode=4 -> diameter [mas]
                              from Parameter.Estimation/data.parameter.txt (fit sizes) in B)

- main results of the image reconstruction run:
  * the results are in the directory BIS.*.Script2.E.1/
  * ASCII file E.liste contains Chi^2 values and image reconstruction parameters for each IRBis run E.? in one l
    (1. part: the runs are sorted according to increasing phase+visibility-qrec value (qrecmode = 1);
     2. part: the runs are sorted according to increasing phase-qrec value (qrecmode = 2))
  * in the postscript file qrecmode\=1/*.lin.ps are the result plots of each run sorted according to
    increasing phase+visibility-qrec value (qrecmode = 1 is in most cases the best choice)
  * the fits files of the reconstructions are in E.?/bestrec.fits (unconvolved)
```

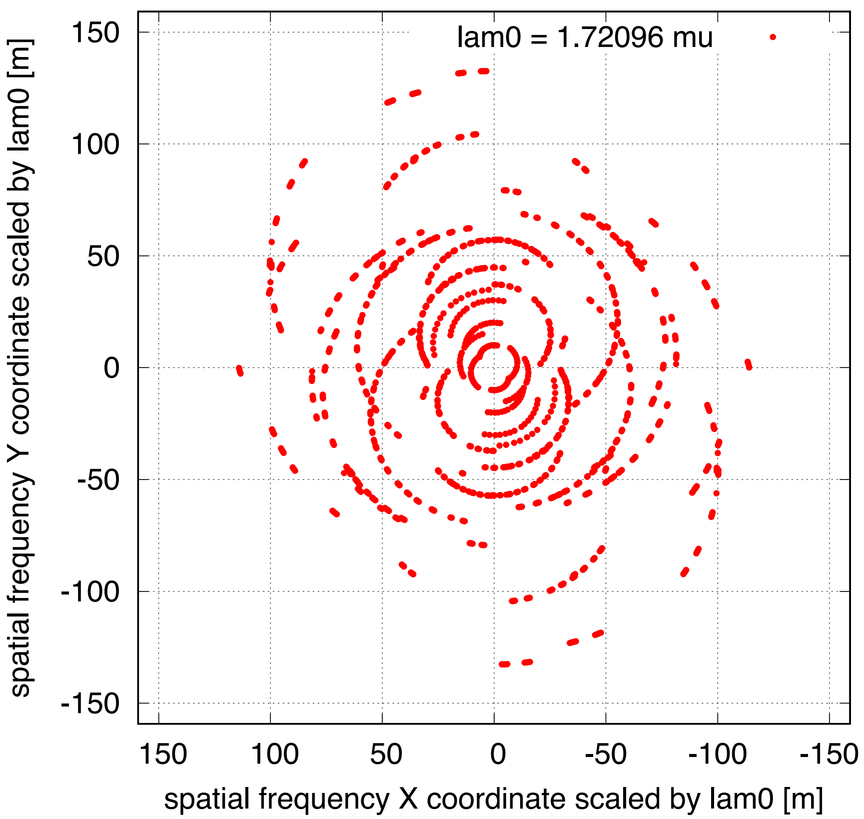
from data.parameter.txt (1. action)

Handling of IRBis - image reconstruction example

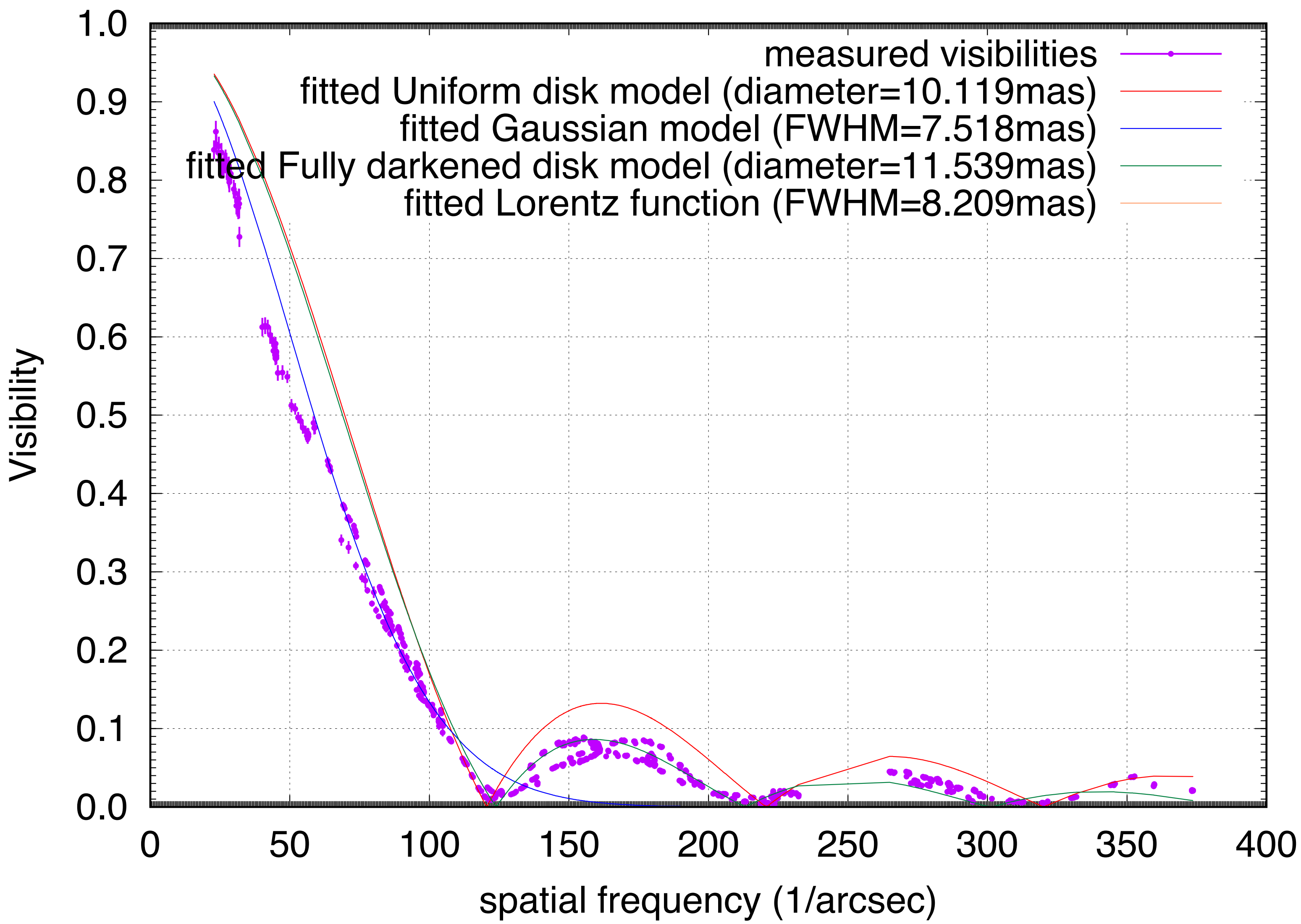
<u>Interferometric observations of R Car with VLTI/PIONIER using the ATs (4 Auxillary Telscopes)</u>

1. action: estimation target size and data parameter

Paramater.Estimation/uv.ps



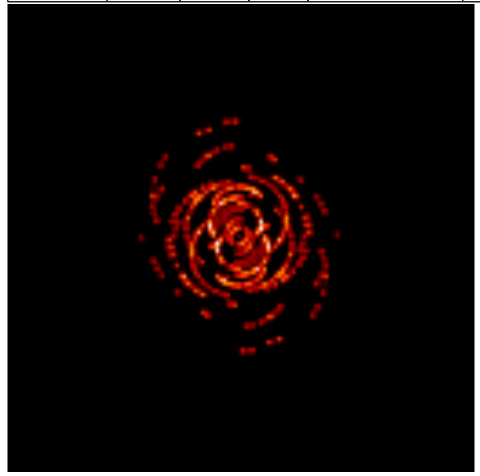
Paramater.Estimation/gaussudfdda.ps

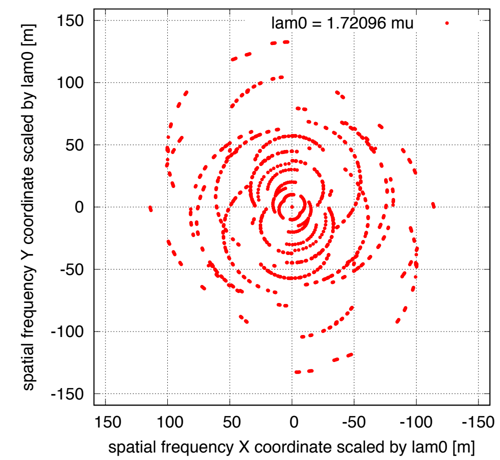


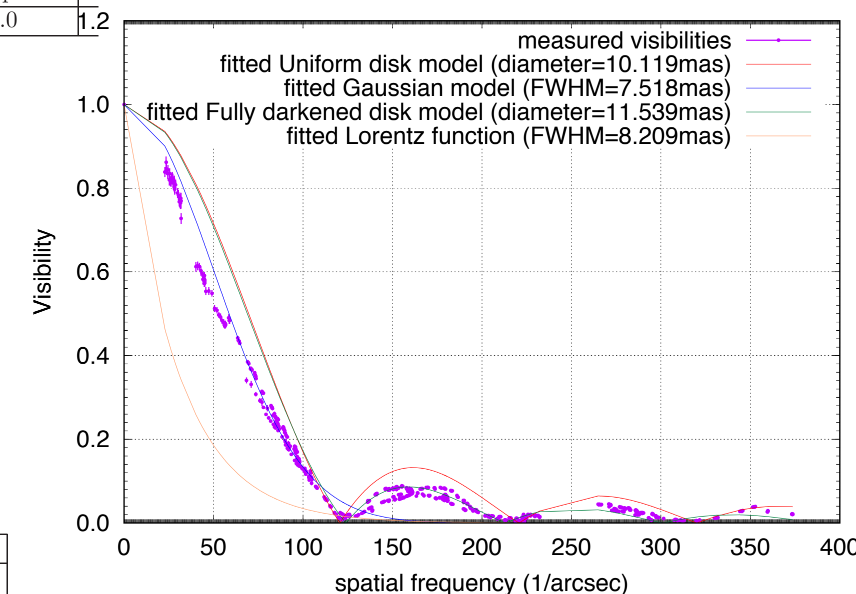
Handling of IRBis - image reconstruction example

2. action:
result of the imaging run

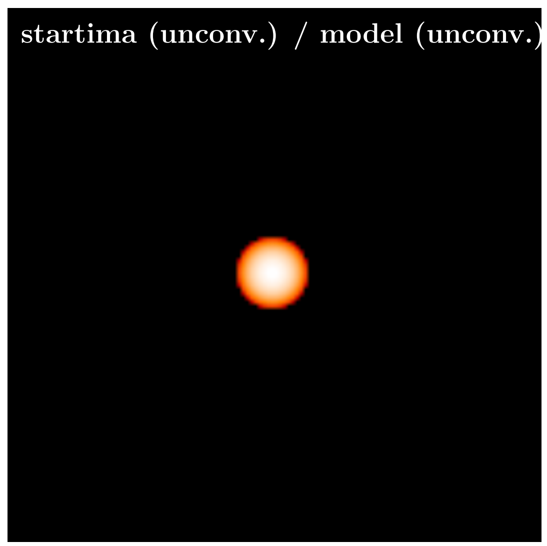
FOV	npix	cost	reg	weightpower	orad	mu	startmode	startparam	superres
84 mas	128	1	-3	0.5	36.0 mas / 1.0 mas / 6	0.0625/0.5/12	4	11.539 mas	-1.0

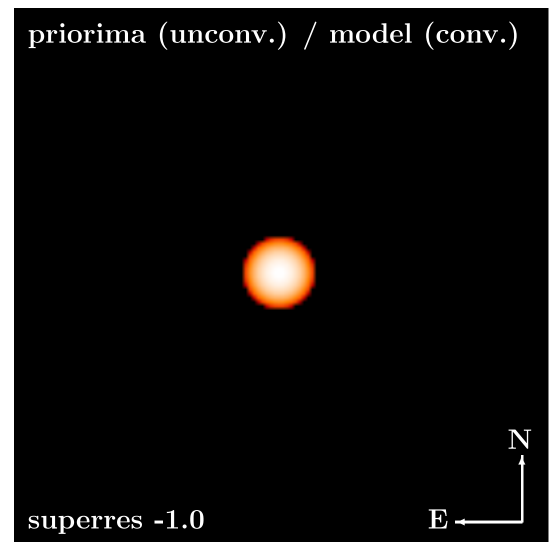


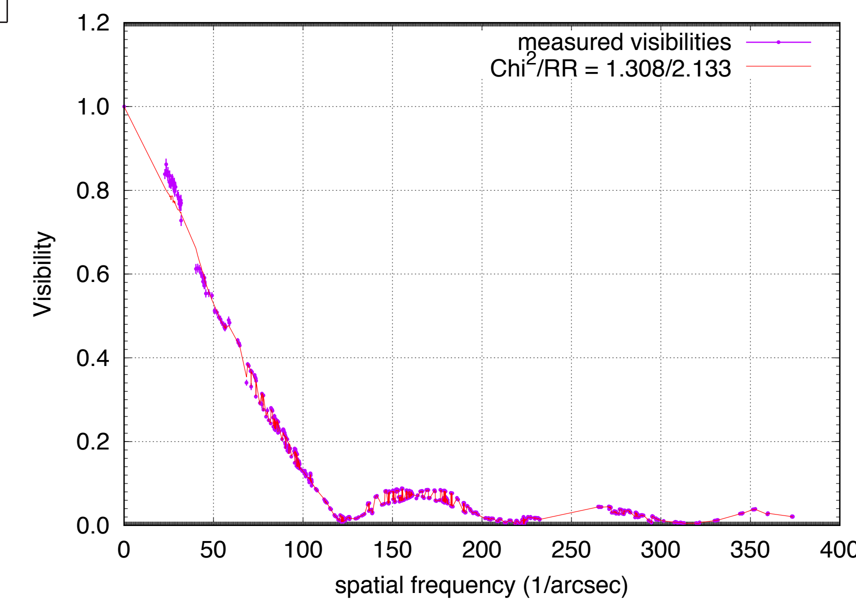


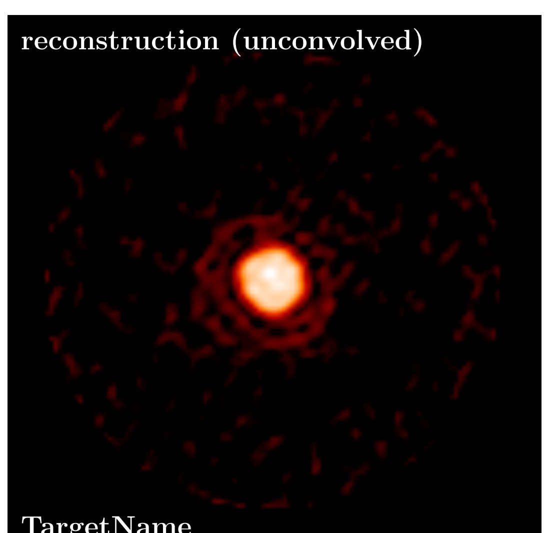


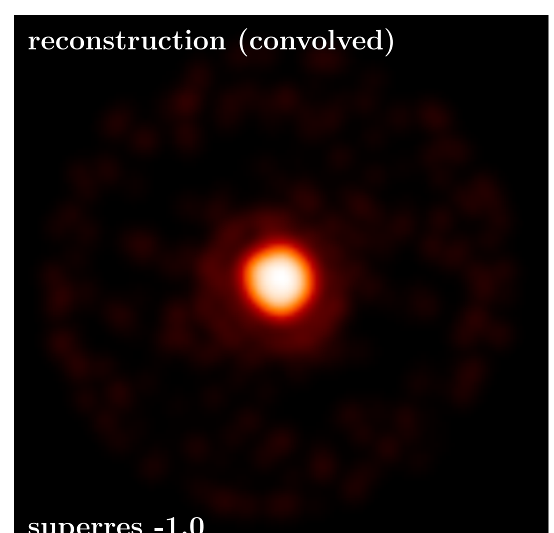
χ^2 BIS+POW	ResRatio	χ^2 V2	ResRatio	χ^2 CP	ResRatio	dist	qrec	cpqrec	
1.025	1.168	0.993	2.153	1.219	1.029	0.000	0.352	0.124	with uv weight
		1.308	2.133	1.563	1.112	-1.000	0.529	0.338	without uv weight

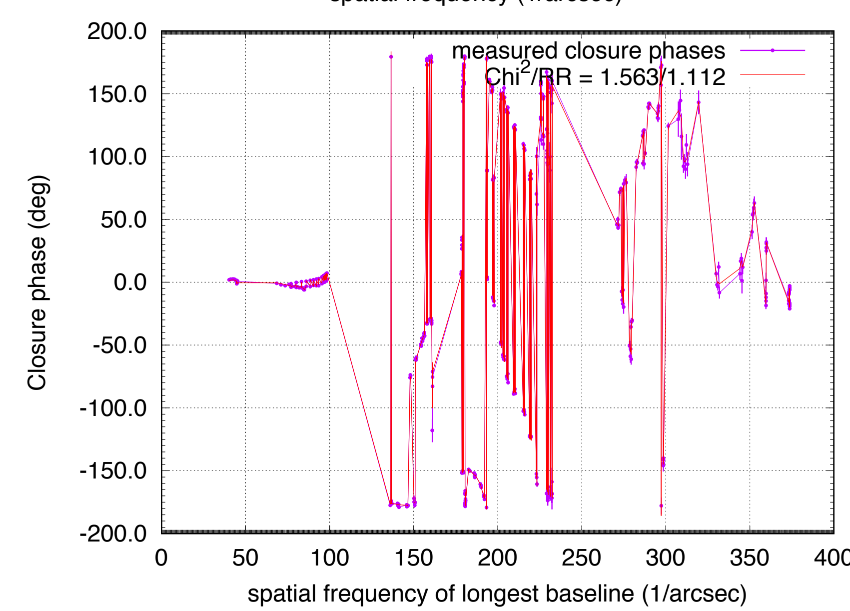












the best reconstruction
displayed in the result
postscript file

```
BIS.*.Script2.E.1/qrecmode\=1/*.sqrt.ps
```

END of the talk

Practice session

```
IRBis image reconstruction tasks:
--------------------------------
1) perform reconstruction from simulated data Simulation.LkHa101.tar (BeautyContest2004
2) perform reconstruction from real MATISSE data of FS CMa HD45677.Lband.data.tar
```

(if you want, further data are available: Binary.delSco.tar and StellarDisk.RCar.tar)

```
Steps to be perforned:
---------------------
1) create IRBisImarec/
mkdir IRBisImarec
cd IRBisImarec

to 1. task:
===========
1) create directory Simulation.LkHa101/ by unpcking the data tar file
   tar -xzvf /data/imarec/Simulation.LkHa101.tar
   cd Simulation.LkHa101/

2) copy the IRBis parameter file mat_cal_imarec_all.2.par from $SCRIPTS/
   cp $SCRIPTS/mat_cal_imarec_all.2.par .

   edit the parameter file mat_cal_imarec_all.2.par
   a) set data = (data/{*.oifits})
      (the data are in data/{*.oifits})
      set lambdaList = (0.1 20.0)
      set fitfwhm2 = 2.0
   b) the original object is in Simulation.LkHa101/data/orig.fits,  1 pixel == 0.05 mas
      set model  = data/orig.fits
      set modelPixelScale = 0.05
   c) for the 1. action:
      set guess = 1
   d) start the 1. action run:
      $SCRIPTS/mat_cal_imarec_all.2.csh mat_cal_imarec_all.2.par

3) next steps and more details are described in IRBisScriptGuidance.pdf
```

! these notes are in **StartIRBisSession.pdf** and the script summary in **IRBisScriptGuidance.pdf** in /data/imarec/ !