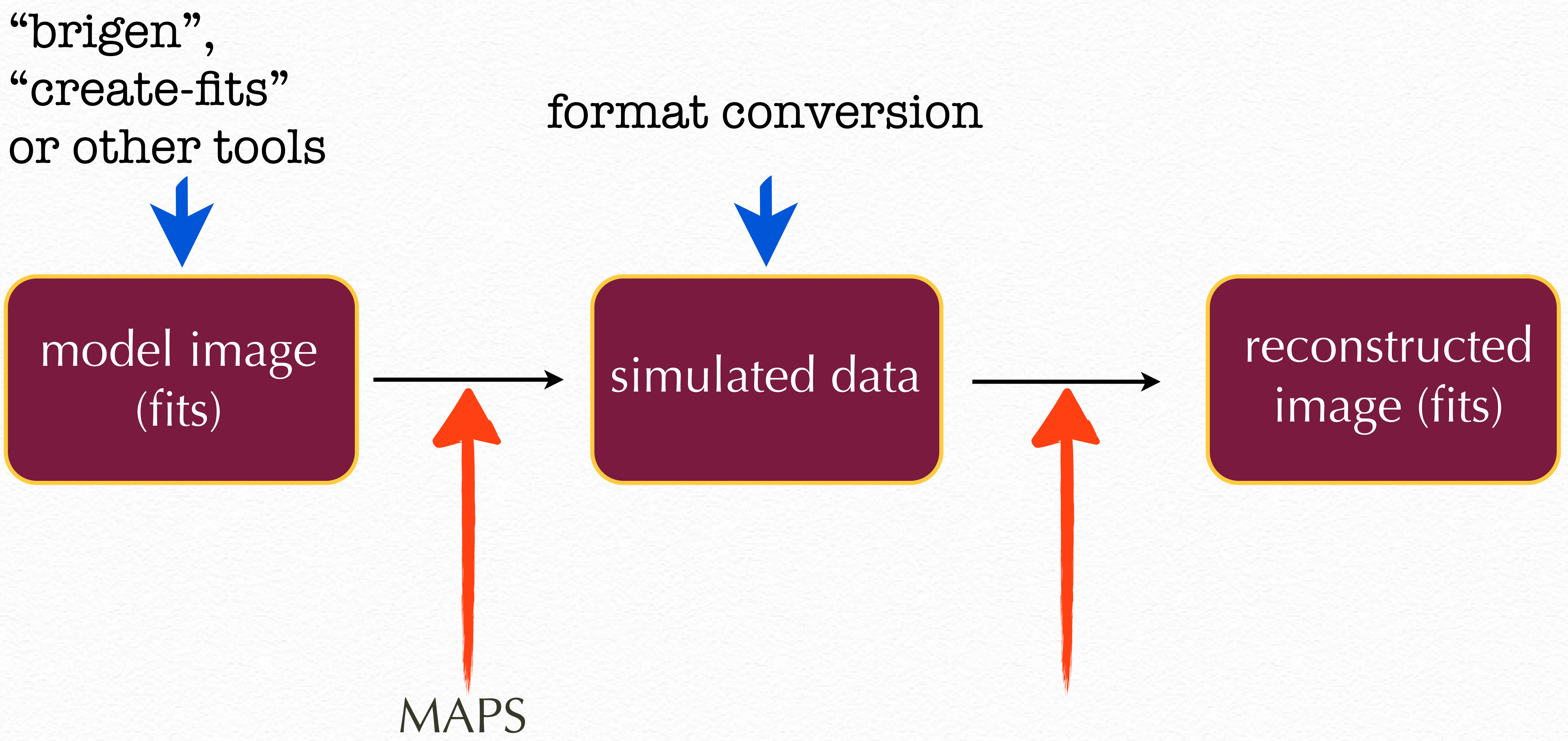
MAPS data simulation

Rusen Lu

Max Planck Institute for Radio Astronomy

A schematic



“brigen”,
“create-fits”
or other tools

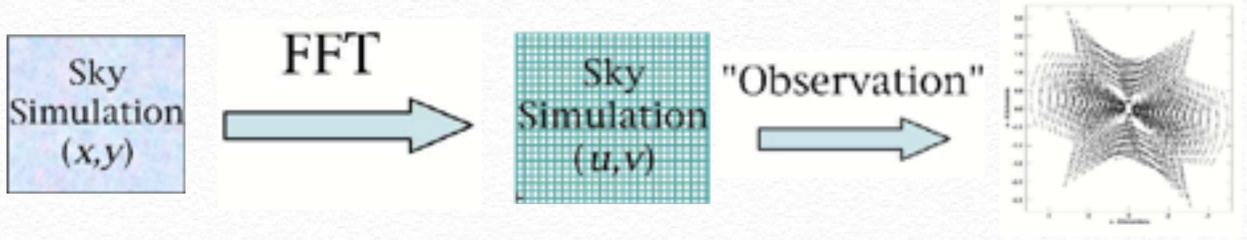
format conversion

model image (fits)

simulated data

reconstructed image (fits)

MAPS



imaging:
CLEAN/MS-CLEAN (UVFITS)
BSMEM/SQUEEZE (OIFITS)

……

MAPS: MIT Array Performance Simulator

(initially) designed to provide a flexible tool for the generation of low frequency array observations (e.g., LOFAR/SKA) and for testing new calibration and processing algorithms
（http://www.haystack.mit.edu/ast/arrays/maps/）

Some key features (relevant to VLBI)

. arbitrary input sky brightness distribution

. user-specified array geometry

. observing specifications (src, fov, time/freq. resolution, bw, integration time, etc.)

. option to include thermal noise

. export simulated data into FITS format

MAPS structure

`array:`

-contains ascii files (.txt) that describe station locations and properties

`stn_layout:`

-contains a set of station files (.layout) that describe the properties of antennas (or multi-element stations)

`text:`

-settings for the ionospheric model; “site” location

`test:`

-sample simulations (a “notes” file for each test, step-by-step guide)

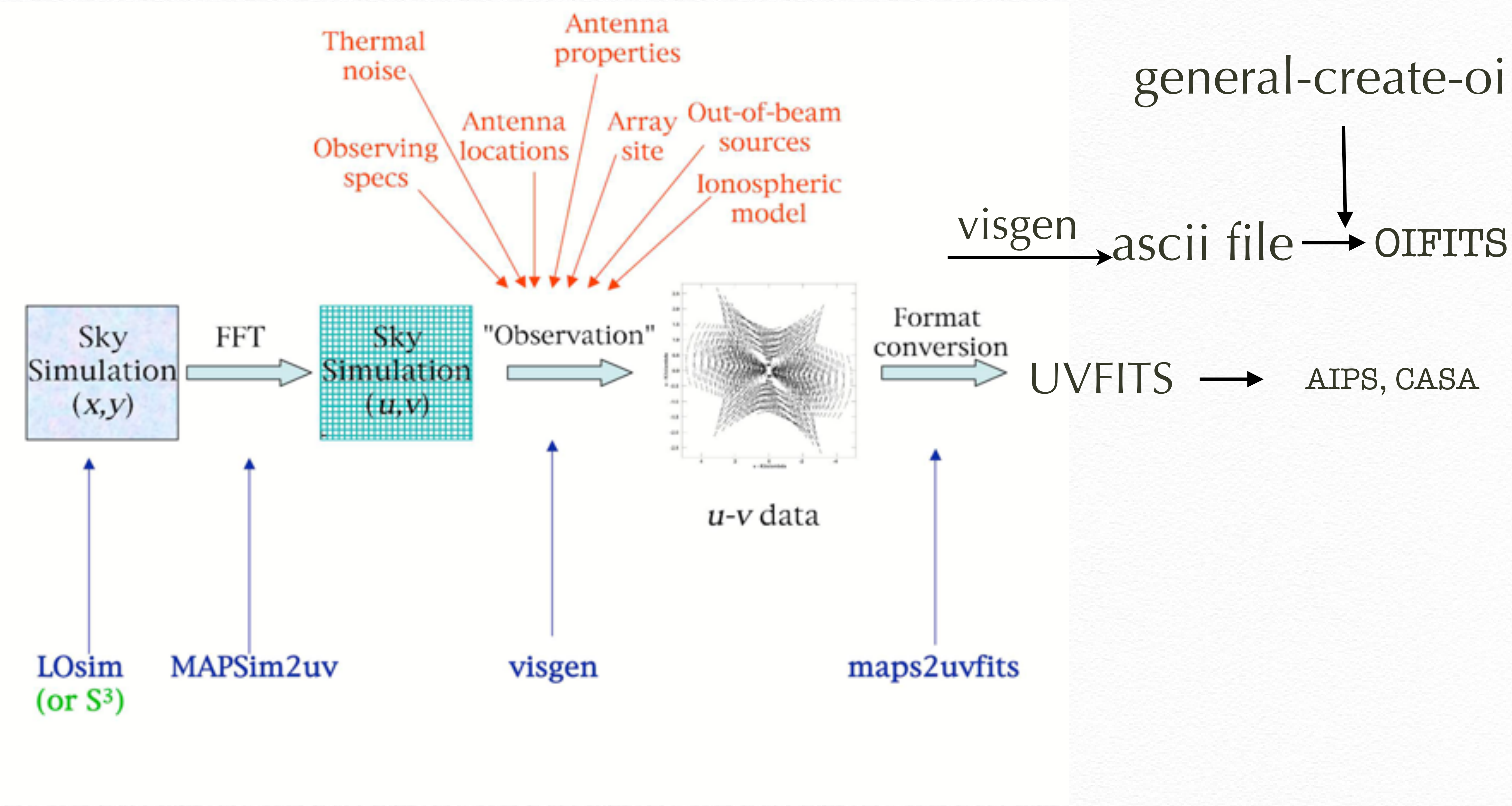
`doc:` the place to go for help

-manual.html, obs_spec.html, and a few memos.

`source code subdirectories:`

-source code

MAPS simulation:



general-create-oi

visgen → ascii file → OIFITS

UVFITS → AIPS, CASA

(courtesy of Lynn Matthews)

Some known limitations

. No comprehensive user manual (but several nice memos)

. Have to set “-n“ switch for MAPS_im2uv manually (requires Jy/steradian; can easily go wrong)

. MAPS_im2uv does not pass along any information from the header of an input FITS image (e.g., field center, or fov) -> specify in the “obs spec” file.

. Visgen fails if too many scans (>110?) in observing specs

. Currently not able to include realistic tropospheric effects (but can handle ionospheric effects for low frequency observations)

MAPS simulation demo:

Assuming MAPS properly installed, we need:

1) an array file (with suffix ".txt") that defines station locations and properties under $SIM/array

2) a set of files (with suffix ".layout") that describe the properties of antennas under $SIM/stn_layout

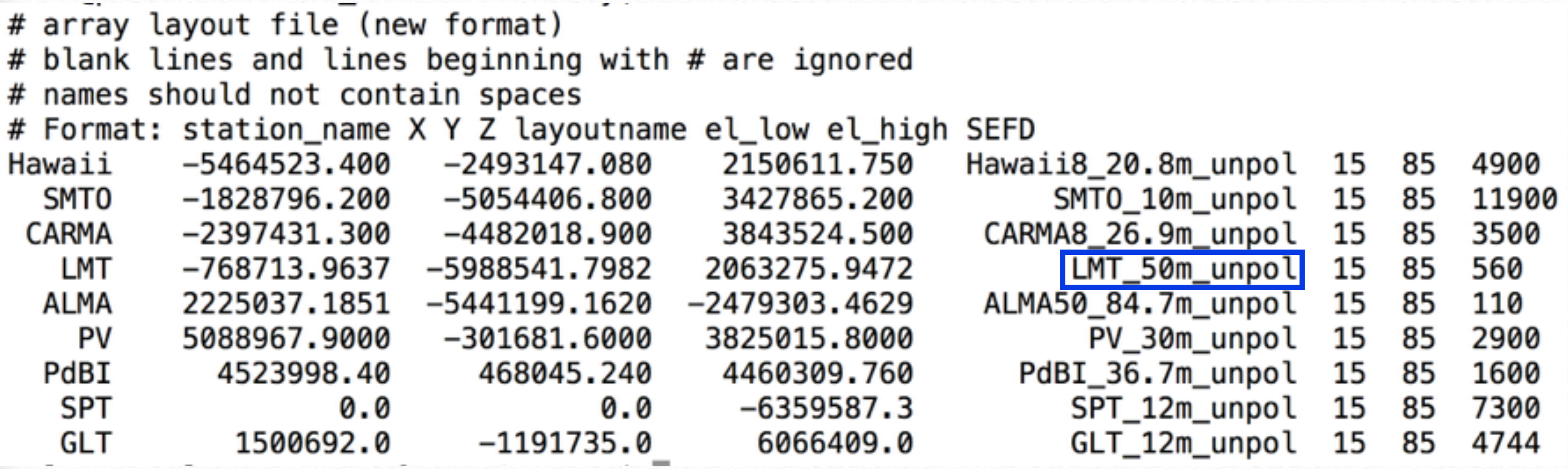
3) a file that defines the "site" name and site coordinates ($SIM/text/site.txt)

In addition, we need to set up a meta-file (obs spec file) to specify observational information(e.g, frequency, center and size of fov, scans, integration time etc.). This file has to be in the directory where you run "visgen".

Demonstration: simulation for a static source

(Scripts that can handle movie frames are available)

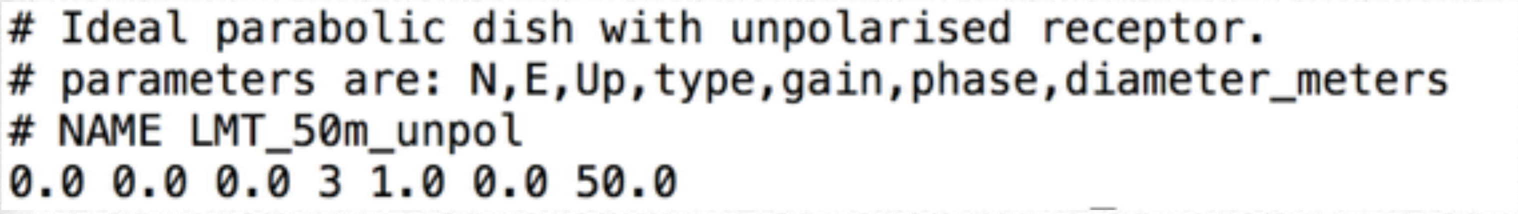
array file:



```
# array layout file (new format)
# blank lines and lines beginning with # are ignored
# names should not contain spaces
# Format: station_name X Y Z layoutname el_low el_high SEFD
Hawaii     -5464523.400   -2493147.080    2150611.750   Hawaii8_20.8m_unpol  15  85  4900
  SMTO     -1828796.200   -5054406.800    3427865.200        SMTO_10m_unpol  15  85  11900
 CARMA     -2397431.300   -4482018.900    3843524.500    CARMA8_26.9m_unpol  15  85  3500
   LMT     -768713.9637  -5988541.7982   2063275.9472          LMT_50m_unpol  15  85  560
  ALMA     2225037.1851  -5441199.1620  -2479303.4629    ALMA50_84.7m_unpol  15  85  110
    PV     5088967.9000   -301681.6000   3825015.8000          PV_30m_unpol  15  85  2900
  PdBI       4523998.40     468045.240    4460309.760      PdBI_36.7m_unpol  15  85  1600
   SPT              0.0            0.0     -6359587.3          SPT_12m_unpol  15  85  7300
   GLT        1500692.0     -1191735.0      6066409.0          GLT_12m_unpol  15  85  4744
```

station layout for each station:

e.g., “LMT_50m_unpol.layout”



```
# Ideal parabolic dish with unpolarised receptor.
# parameters are: N,E,Up,type,gain,phase,diameter_meters
# NAME LMT_50m_unpol
0.0 0.0 0.0 3 1.0 0.0 50.0
```

site file: coordinates of the “observatory”

```
LMT    -97:18:53      18:59:06
SMTO -109:52:19      32:42:06
…
```

Why BSMEM or MACIM/SQUEEZE?

- Primary observables:

  OI:
  power spectrum (V^2) and bispectrum (T3)

  VLBI:
  (at cm: complex visibility: amp + phase)
  mm/sub-mm: amp+ bispectrum (phase strongly corrupted on short time scales)

- Limited uv-coverage:

  OI: at most 6 (?) antennas

  mm-VLBI (EHT): 8(?)

Why BSMEM or MACIM/SQUEEZE?

- A large class of images can be consistent with a particular interferometric data set. This is more true for optical interferometry/mm-VLBI due to the unavailability of absolute visibility phase.
- Imaging algorithms such as CLEAN or MEM combined with self-cal attempts to find the ‘best’ possible image, but both finding this ‘best’ image and interpretation of features within the image can be difficult.
- This in general requires some kind of regularization (prior). Regularization punishes images that look ’bad’ to find a compromise between lowering the $\chi 2$ statistic and achieving an optimal regularization statistic.

“best-fitting” with regularizers !

Find “smoothest” image consistent with data

MEM uses “entropy” to parametrize “smoothness”

Fraction of flux in pixel i

Entropy

$$S = -\sum_i f_i \ln \frac{f_i}{I_i}$$

Image prior

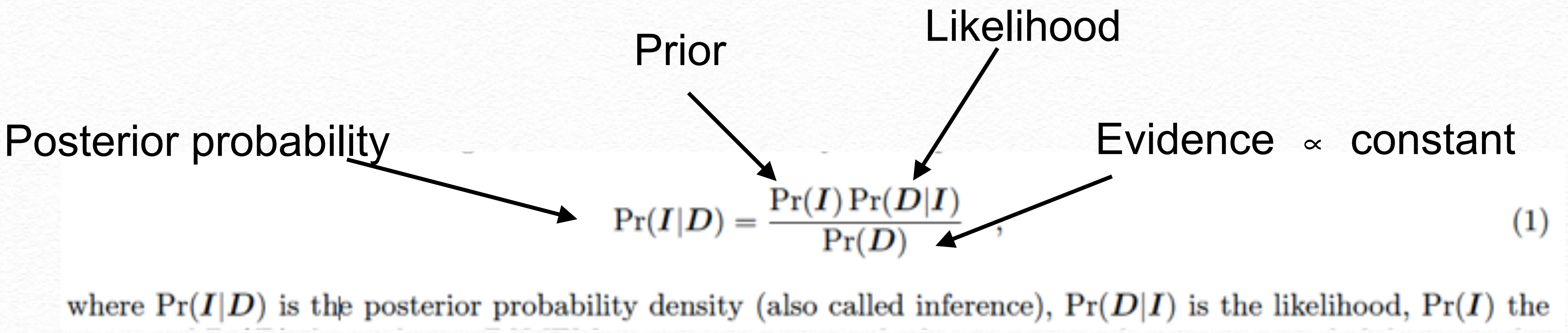
Sum over all pixels

Skilling & Bryan (1984)

(MEM is not unique, there are many other regularizers)

Solving the problem in a bayesian framework

- maximize the posterior probability of the image



Prior

Likelihood

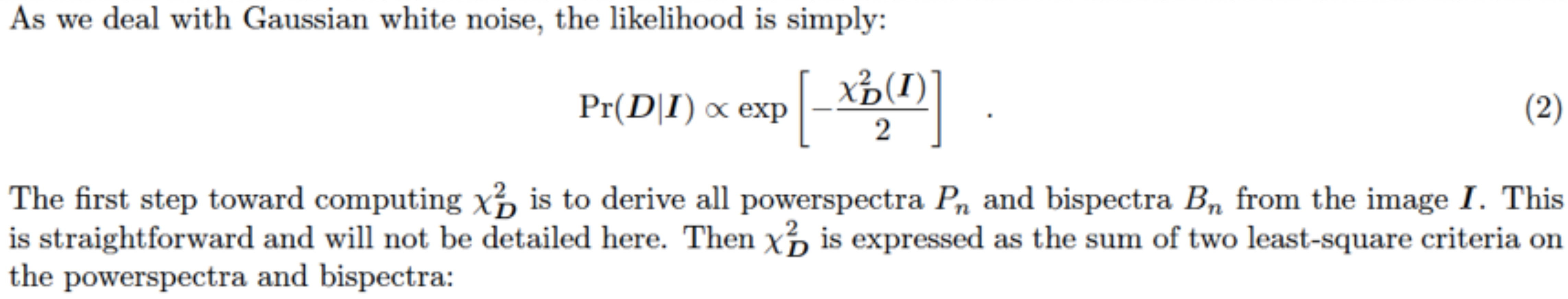
Posterior probability

Evidence ∝ constant

$$\Pr(\boldsymbol{I}|\boldsymbol{D}) = \frac{\Pr(\boldsymbol{I})\,\Pr(\boldsymbol{D}|\boldsymbol{I})}{\Pr(\boldsymbol{D})}, \qquad (1)$$

where $\Pr(\boldsymbol{I}|\boldsymbol{D})$ is the posterior probability density (also called inference), $\Pr(\boldsymbol{D}|\boldsymbol{I})$ is the likelihood, $\Pr(\boldsymbol{I})$ the

likelihood function Pr(D|I):

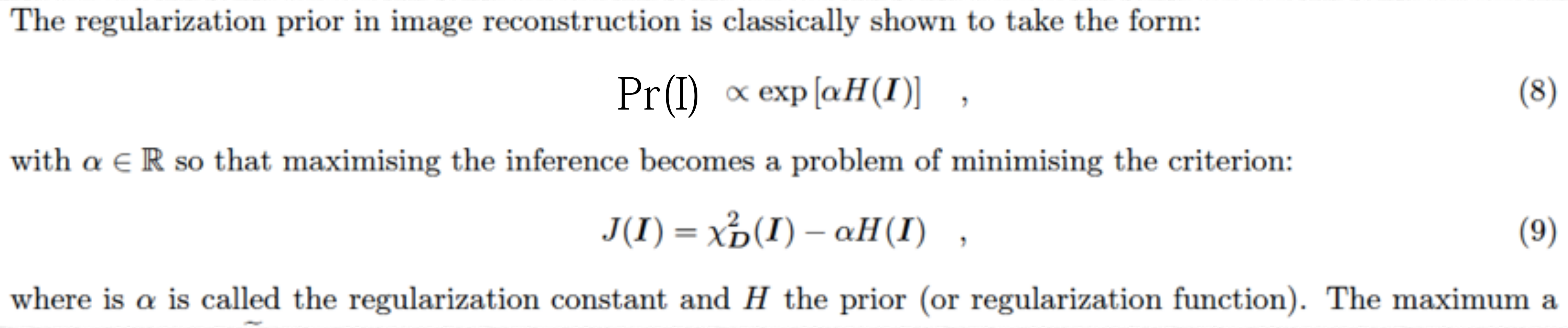


As we deal with Gaussian white noise, the likelihood is simply:

$$\Pr(\boldsymbol{D}|\boldsymbol{I}) \propto \exp\left[-\frac{\chi^2_{\boldsymbol{D}}(\boldsymbol{I})}{2}\right] \quad . \tag{2}$$

The first step toward computing $\chi^2_{\boldsymbol{D}}$ is to derive all powerspectra P_n and bispectra B_n from the image $\boldsymbol{I}$. This is straightforward and will not be detailed here. Then $\chi^2_{\boldsymbol{D}}$ is expressed as the sum of two least-square criteria on the powerspectra and bispectra:

Prior:



The regularization prior in image reconstruction is classically shown to take the form:

$$\Pr(I) \propto \exp\left[\alpha H(\boldsymbol{I})\right] \quad , \tag{8}$$

with $\alpha \in \mathbb{R}$ so that maximising the inference becomes a problem of minimising the criterion:

$$J(\boldsymbol{I}) = \chi^2_{\boldsymbol{D}}(\boldsymbol{I}) - \alpha H(\boldsymbol{I}) \quad , \tag{9}$$

where is α is called the regularization constant and H the prior (or regularization function). The maximum a

Finding the minimum

$$\mathrm{I}_{\mathrm{MAP}} = \underset{I}{\operatorname{argmin}} \left[\chi^2_{\boldsymbol{D}}(\boldsymbol{I}) - \alpha H(\boldsymbol{I}) \right]$$

- Two basic categories of algorithms:

Method 1. Gradient descent (semi-Newton, trust region)

- Requires gradient of the criterion with respect to the image pixels
- Explores the parameter space much faster, but will fall into local minimum

BSMEM

Method 2. Markov Chain Monte Carlo

- Flux elements randomly move on a pixel grid, stopping based on criterion values
- Simulated annealing, parallel tempering, nested sampling
- Flexible priors, use of custom regularizers (e.g., pt sources, dark energy, TotVar)

(courtesy of John Monnier)

MACIM/SQUEEZE

Classic regularizers

- pixel-based

$$R(\boldsymbol{i}) = -\sum_k i_k \log \frac{i_k}{i_k^0}$$

Maximum Entropy

$$R(\boldsymbol{i}) = \sum_k \log i_k$$

Burg Entropy

$$R(\boldsymbol{i}) = \sum_k i_k^2$$

l2 norm (smoothness),Tikhonov regularizer

- Total Variation (TV) is based on norm of the gradient

$$R(\boldsymbol{i}) = \ell_1(\nabla \boldsymbol{i}) = \sum_k |\nabla(\boldsymbol{i})|_k$$

  - Favors uniform zones with sharp edges
  - Used in medical imaging

(courtesy of John Monnier)

Run BSMEM or SQUEEZE

(Before imaging, we use "general-create-oi" to convert MAPS simulated data to oifits format)

bsmem can be run like this (bsmem -h for more help):

```
bsmem -d input_file -mt 0 -mf model_flux -p x -w x -wavmin x  -wavmax xxx
```

here -mt 0 means a flat prior, -mf is for the model total flux, -p followed by pixel size in mas, -w for width of the image (in pixels) and wavmin and wavmax are wavelength limits for data selection purposes.

squeeze can be run like this (squeeze -h for more help):

```
squeeze input_file -s x -w x -fs x -en 1 -f_any 0.001 -f_copy 0.5 -e x
```

here -s is for pixel size in mas, -w for width of the image (in pixels), -fs the total flux, and -en for entropy regularization. f_any and f_copy are convergence settings and set fraction of steps that look anywhere and do copycat. -e is for the number of elements per realization (do not use a number more than 10000, would be very slow otherwise).

Image fidelity measurement

MSE (Mean Square Error): Pixel-to-pixel comparison

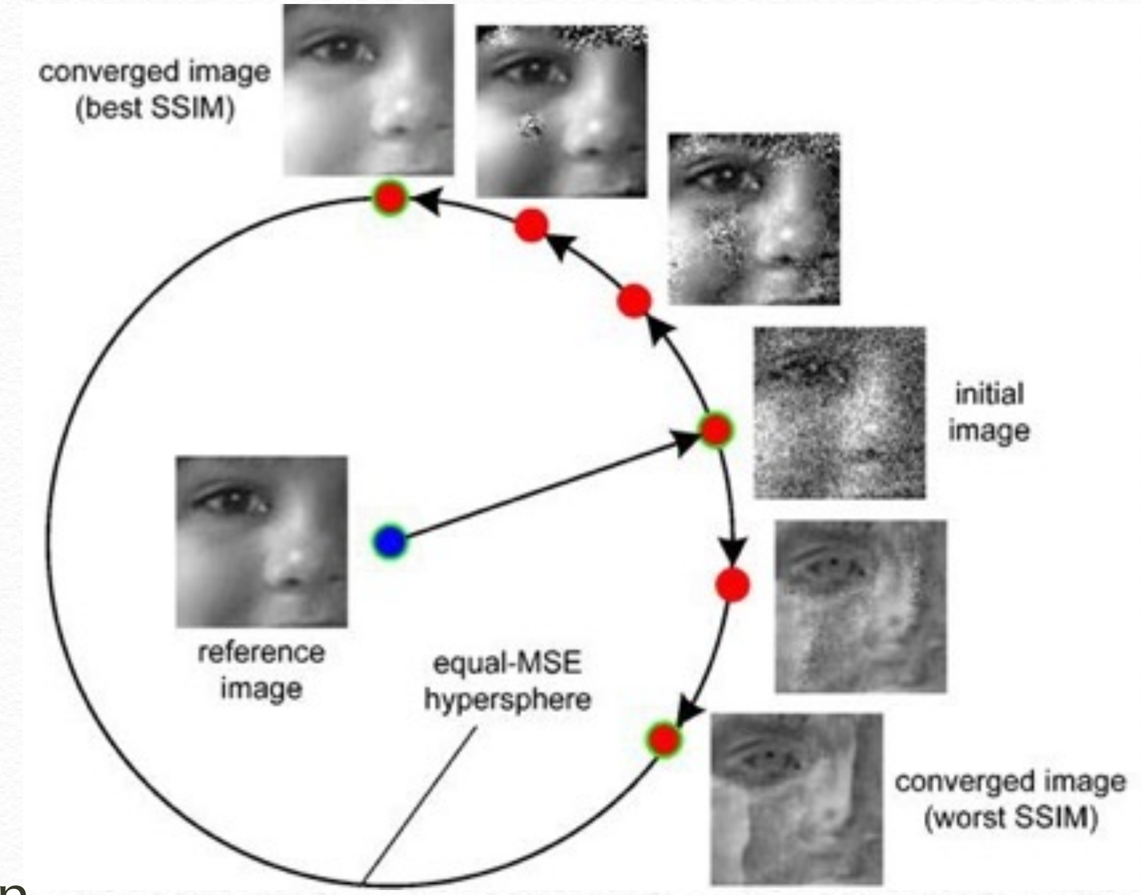
$$\mathrm{MSE} = \frac{\sum_{i=1}^{L} |I_i - K_i|^2}{\sum_{i=1}^{L} |I_i|^2},$$

<u>Structural Similarity</u> (SSIM, Wang 2004; DSSIM, Loza et al., 2009)

More natural/human-like metric

$$\begin{aligned}\mathrm{SSIM(I, K)} &= \left(\frac{2\mu_I\mu_K}{\mu_I^2+\mu_K^2}\right)\left(\frac{2\sigma_I\sigma_K}{\sigma_I^2+\sigma_K^2}\right)\left(\frac{\sigma_{IK}}{\sigma_I\sigma_K}\right)\\ &= l(I, K)c(I, K)s(I, K),\end{aligned}$$

three comparisons:
luminance comparison, contrast comparison, and structure comparison (s(I, K))



Wang 2004