Imaging
NAASC data analysis workshop

Jim Braatz

Based on material from David Wilner, Scott Schnee, Remy Indebetouw

Atacama Large Millimeter/submillimeter Array
Expanded Very Large Array
Robert C. Byrd Green Bank Telescope
Very Long Baseline Array
Overview

• Goals of this talk:
  – Gain some intuition for interferometric imaging
  – Introduce deconvolution in CASA (clean)
  – Introduce various imaging methods available in CASA

• More formal description of imaging available in NRAO Synthesis Imaging Workshop lectures
From Sky Brightness to Visibility

1. An interferometer measures the interference pattern observed by pairs of apertures.
2. The interference pattern is directly related to the source brightness. In particular, for small fields of view the complex visibility, \( V(u,v) \), is the 2D Fourier transform of the brightness on the sky, \( T(x,y) \)

\[
V(u, v) = \int \int T(x, y) e^{2\pi i (ux+vy)} dx dy
\]

\[
T(x, y) = \int \int V(u, v) e^{-2\pi i (ux+vy)} dudv
\]

(van Cittert-Zernike theorem)

Fourier space/domain

Image space/domain

uv plane

image plane
Some 2D Fourier Transform Pairs

T(x,y)

δ Function

Gaussian

Amp{V(u,v)}

Constant

Gaussian

narrow features transform to wide features (and vice-versa)
More 2D Fourier Transform Pairs

$T(x,y)$

elliptical Gaussian

Disk

$\text{Amp}\{V(u,v)\}$

elliptical Gaussian

Bessel

sharp edges result in many high spatial frequencies
ALMA observes planetary disk

Fourier transform of nearly symmetric planetary disk

Band 6

Band 7
Real Example: VLA observes Jupiter

- A 6cm VLA observation of Jupiter:

Fourier transform of nearly symmetric planetary disk

bad data
How to analyze (imperfect) interferometer data

- uv plane analysis
  - Ok for “simple” sources, e.g. point sources, disks
  - Recall our BP calibrator from yesterday
Amplitude and Phase

- complex numbers: (real, imaginary) or (amplitude, phase)
  - amplitude tells “how much” of a certain frequency component
  - phase tells “where” this component is located

\[ T(x,y) \quad \text{Amp}\{V(u,v)\} \quad \text{Pha}\{V(u,v)\} \]
Amplitude and Phase

- complex numbers: (real, imaginary) or (amplitude, phase)
  - amplitude tells “how much” of a certain frequency component
  - phase tells “where” this component is located

\[
T(x,y) \quad \text{Amp}\{V(u,v)\} \quad \text{Pha}\{V(u,v)\}
\]
Amplitude and Phase

- complex numbers: (real, imaginary) or (amplitude, phase)
  - amplitude tells “how much” of a certain frequency component
  - phase tells “where” this component is located

\[ T(x,y) \quad \text{Amp}\{V(u,v)\} \quad \text{Pha}\{V(u,v)\} \]
Dirty Images from a Dirty Beam

- We sample Fourier domain at discrete points
  \[ B(u, v) = \sum_k (u_k, v_k) \]

- the inverse Fourier transform is
  \[ T^D(x, y) = FT^{-1}\{ B(u, v) \times V(u, v) \} \]

- the convolution theorem tells us
  \[ T^D(x, y) = b(x, y) \otimes T(x, y) \]

where \( b(x, y) = FT^{-1}\{ B(u, v) \} \) (the point spread function)

Fourier transform of sampled visibilities yields the true sky brightness convolved with the point spread function (the “dirty image” is the true image convolved with the “dirty beam”)

Dirty Beam and Dirty Image

\[ b(x,y) \]  (dirty beam)

\[ T(x,y) \]  (dirty image)

\[ B(u,v) \]  

\[ T^D(x,y) \]  (dirty image)
8 Antennas
16 Antennas - Compact
16 Antennas - Extended
16 Antennas – Compact – 8 hours

- Sky response = PSF = dirty beam = Fourier transform of uv distribution
How to analyze (imperfect) interferometer data?

- image plane analysis
  - dirty image $T^D(x,y) = $ Fourier transform $\{V(u,v)\}$
  - deconvolve $b(x,y)$ from $T^D(x,y)$ to determine (model of) $T(x,y)$

visibilities  dirty image  sky brightness
Basic CLEAN Algorithm

1. Initialize a residual map to the dirty map

   1. Start loop
   2. Identify strongest feature in residual map as a point source
   3. Add this point source to the clean component list
   4. Convolve the point source with $b(x,y)$ and subtract a fraction $g$ (the loop gain) of that from residual map
   5. If stopping criteria not reached, do next iteration

2. Convolve Clean component (cc) list by an estimate of the main lobe of the dirty beam (the “Clean beam”) and add residual map to make the final “restored” image
Basic CLEAN Algorithm (cont)

• stopping criteria
  – residual map max < multiple of rms (when noise limited)
  – residual map max < fraction of dirty map max (dynamic range limited)
  – max number of clean components reached (no justification)

• loop gain
  – good results for $g \sim 0.1$ to 0.3
  – lower values can work better for smoother emission, $g \sim 0.05$

• easy to include a priori information about where to search for clean components (“clean boxes”)
  – very useful but potentially dangerous!
CLEAN

$T^D(x,y)$

restored image

residual map

CLEAN model
Deconvolution algorithms: Hogbom

Data visibilities → grid & FFT → Dirty image

Iterative removal of dirty beam

Model image

Subtracts full PSF in image domain

For complex images, errors can build
Deconvolution algorithms: Clark

Data visibilities → Gridded data → Dirty image

Gridded model → FFT → Model image

Iterative removal of dirty beam

Subtracts truncated PSF in image domain

Periodically subtracts from gridded data in uv domain
Deconvolution algorithms: Cotton-Schwab

Cotton-Schwab (csclean):
- substracts truncated PSF in image domain
- major cycle substracts from full visibilities
- significant I/O per major cycle
Dirty Beam Shape and Weighting

• Each visibility point is given a weight in the imaging step

• Natural
  – Weights inversely proportional to noise variance
  – Best point-source sensitivity; poor beam characteristics

• Uniform
  – Weights inversely proportional to sampling density (longer baseline are given higher weight than in natural)
  – Best resolution; poorer noise characteristics

• Briggs (Robust)
  – A graduated scheme using the parameter \textit{robust}
  – In CASA, set \textit{robust} from -2 (~ uniform) to +2 (~ natural)
  – \textit{robust} = 0 often a good choice
Imaging Results

Natural Weight Beam

CLEAN image
Imaging Results

Uniform Weight Beam

CLEAN image
Imaging Results

Robust=0 Beam

CLEAN image
Clean in CASA:

```
CASA <13>: inp clean
--------> inp(clean)
# clean :: Invert and deconvolve images with selected algorithm
vis    =  
imagename =  
outlierfile =  
field =  
spw =  
selectdata = False  
mode = 'mfs'  
   nterms = 1  
   reffreq =  
gridmode =  
niter = 500  
gain = 0.1  
threshold = '0.0Jy'  
psfmode = 'clark'  
   imagemode = 'csclean'  
   cyclefactor = 1.5  
   cyclespeedup = -1  
multiscale =  
interactive = False  
mask =  
   imsize = [256, 256]  
   cell = ['1.0arcsec']  
   phasecenter =  
   restfreq =  
   stokes = 'I'  
weighting = 'natural'  
   uvtaper = False  
modelimage =  
   restoringbeam = ['']  
   pbcor = False  
   minpb = 0.2  
   usescratch = False  
allowchunk = False  
async = False  
```

# Name of input visibility file
# Pre-name of output images
# Text file with image names, sizes, centers for outliers
# Field Name or id
# Spectral windows e.g. '0"3', '' is all
# Other data selection parameters
# Spectral gridding type (mfs, channel, velocity, frequency)
# Number of Taylor coefficients to model the sky frequency dependence
# Reference frequency (nterms > f),'' uses central data-frequency
# Gridding kernel for FFT-based transforms, default='' None
# Maximum number of iterations
# Loop gain for cleaning
# Flux level to stop cleaning, must include units: '1.0Jy'
# Method of PSF calculation to use during minor cycles
# Options: 'csclean' or 'mosaic', '' uses psfmode
# Controls how often major cycles are done, (e.g. 5 for frequently)
# Cycle threshold doubles in this number of iterations
# Deconvolution scales (pixels): [] = standard clean
# Use interactive clean (with GUI viewer)
# Cleanbox(es), mask image(s), region(s), or a level
# x and y image size in pixels. Single value; same for both
# x and y cell size(s). Default unit arcsec,
# Image center: direction or field index
# Rest frequency to assign to image (see help)
# Stokes params to image (eg I,IV,IQ,IOUV)
# Weighting of uv (natural, uniform, Briggs, ...)
# Apply additional uv tapering of visibilities
# Name of model image(s) to initialize cleaning
# Output Gaussian restoring beam for CLEAN image
# Output primary beam-corrected image
# Minimum PB level to use
# True if to save model visibilities in MODEL_DATA column
# Divide large image cubes into channel chunks for deconvolution
# If true the taskname must be started using clean(...)

(29)
Basic Image Parameters: Pixel Size and Image Size

• pixel size
  – should satisfy $\Delta x < 1/(2 \ u_{\max})$ $\Delta y < 1/(2 \ v_{\max})$
  – in practice, 3 to 5 pixels across the main lobe of the dirty beam

• image size
  – Consider FWHM of primary beam (e.g. ~ 20” at Band 7)
  – Be aware that sensitivity is not uniform across the primary beam
  – Use mosaicing to image larger targets
  – Not restricted to powers of 2

* if there are bright sources in the sidelobes, they will be aliased into the image (need to make a larger image)
# Maximum Angular Scale

<table>
<thead>
<tr>
<th>Band</th>
<th>Frequency (GHz)</th>
<th>Primary beam (&quot;)</th>
<th>Range of Scales (&quot;)</th>
<th>C32-1</th>
<th>C32-9</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>84-116</td>
<td>72 - 52</td>
<td>4.2 - 24.6</td>
<td>0.7 - 15.1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>211-275</td>
<td>29 - 22</td>
<td>1.8 - 10.7</td>
<td>0.3 - 6.6</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>275-373</td>
<td>22 - 16</td>
<td>1.2 - 7.1</td>
<td>0.2 - 4.4</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>602-720</td>
<td>10 - 8.5</td>
<td>0.6 - 3.6</td>
<td>0.1 - 2.2</td>
<td></td>
</tr>
</tbody>
</table>

- **Range** from synthesized beam to maximum angular scale (MAS)
- **Smooth** structures larger than MAS begin to be resolved out.
- All flux on scales larger than $\lambda/B_{\text{min}}$ (~2 x MAS) completely resolved out.
Output of \textit{clean}

Minimally:

- `my_image.flux`  
  Relative sky sensitivity
- `my_image.image`  
  Cleaned and restored image
- `my_image.mask`  
  Clean “boxes”
- `my_image.model`  
  Clean components
- `my_image.psf`  
  Dirty beam
- `my_image.residual`  
  Residual
Some of the capabilities of clean ...
Imaging spectral lines
# clean :: Invert and deconvolve images with selected algorithm
vis = 'ngc3256_co.ms.contsub' # Name of input visibility file
imagename = 'ngc3256_co' # Pre-name of output images
outlierfile = '' # Text file with image names, sizes, centers for outliers
field = '' # Field Name or id
spw = '0:38:87' # Spectral windows e.g. '0=3', '' is all
selectdata = False # Other data selection parameters
mode = 'channel' # Spectral gridding type (mfs, channel, velocity, frequency)
 nchan = 50 # Number of channels (planes) in output image: -1 = all
 start = '' # Begin the output cube at the frequency of this channel in the MS
 width = '' # Width of output channel relative to MS channel (# to average)
 interpolation = 'linear' # Spectral interpolation (nearest, linear, cubic). Use nearest for mode=channel
 chaniter = False # Clean each channel to completion (True), or all channels each cycle (False)
 outframe = '' # velocity frame of output image

restfreq = '115.271201800GHz' # Rest frequency to assign to image (see help)
Imaging spectral lines: continuum subtraction

- Generally would like to subtract continuum emission (we will see how to identify line-free channels in hands-on session)
- Use `uvcontsub` to do the subtraction in uv plane.

```python
casa <11>: inp
----------> inp()
  # uvcontsub :: Continuum fitting and subtraction in the uv plane
  vis = 'ngc3256_co.ms'  # Name of input MS. Output goes to vis + ",.contsub"
  field = '*'            # Select field(s) using id(s) or name(s)
  fitspw = '0:20"53;71"120'  # Spectral window; channel selection for fitting the continuum
  combine = '*'          # Data axes to combine for the continuum estimation (none, or spw and/or scan)
  solint = 'int'        # Continuum fit timescale (int recommended!)
  fitorder = 0          # Polynomial order for the fits
  spw = ' '             # Spectral window selection for output
  want_cont = False     # Create vis + ",.cont" to hold the continuum estimate.
  async = False         # If true the taskname must be started using uvcontsub(...)```
Mosaics

Example: SMA 1.3 mm observations: 5 pointings
- Primary beam ~1’
- Resolution ~3”

In Cycle 1, the number of pointings <= 150.

Petitpas et al.

AAS Splinter Session, Jan 11, 2011
Imaging mosaics

```
imagermode = 'mosaic'
mosweight = False
ftmachine = 'ft'
scaletype = 'SAULT'
cyclefactor = 1.5
cyclespeedup = -1
flatnoise = True
```

- `ftmachine = “ft”` : shift and add in image plane
- `ftmachine = “mosaic”` : add in uv plane and invert together
multiscale clean

multiscale  = [0, 5, 15]  # Deconvolution scales (pixels); [] = standard clean
Interactive Clean

- residual image in viewer
- define a mask with R-click on shape type
- define the same mask for all channels
- or iterate through the channels with the tape deck and define separate masks
Interactive Clean

• perform N iterations

• and return – every time the residual is displayed is a major cycle

• continue until #cycles or threshold reached, or user stop
Combining with single-dish or other interferometric maps

If you have only images:
  feather

If you have an image and a MS:
  use clean with the image as “modelimage”
  or feather
Combining with other data: feather

```
# feather :: Combine two images using their Fourier transforms
imagename     = ''   # Name of output feathered image
highres       = ''   # Name of high resolution (interferometer) image
lowres        = ''   # Name of low resolution (single dish) image
async         = False # If true the taskname must be started using feather(...)
```

- 32-antenna ALMA
- 6-antenna ACA
- feather
Combining with other data: modelimage

```plaintext
# clean :: Invert and deconvolve images with selected algorithm

modelimage = ''  # Name of model image(s) to initialize cleaning
```
Multi-Frequency Synthesis (mfs)

```
mode = mfs
n terms = 1
reffreq =

'mfs'
# Spectral gridding type (mfs, channel, velocity, frequency)
# Number of Taylor coefficients to model the sky frequency dependence
# Reference frequency (n terms > 1),'' uses central data-frequency
```
higher order mfs

- MFS (mode mfs)
  - nterm=2 compute spectral index, 3 for curvature etc.
  - needed for bandwidths ~5% or more (S/N dependent)
  - tt0 average intensity, tt1 alpha*tt0, alpha images output
  - takes at least nterms longer (image size dependent)
Miscellaneous

- clean will restart from existing files
  - will first recompute residuals from model
  - be sure this is what you want

- mask image in particular can be reused but be careful of imsize

- total cleaned flux not reported until end

- don’t do ^C while imaging – can do bad things to your MS
OK, Let’s give it a try!

The imaging script we will follow is here:

~/distrib/imaging/scripts/basic_imaging/Basic_imaging.py

You can get more information and background on the data and methods here: