Model–Based Deconvolution

Massachusetts Institute of Technology

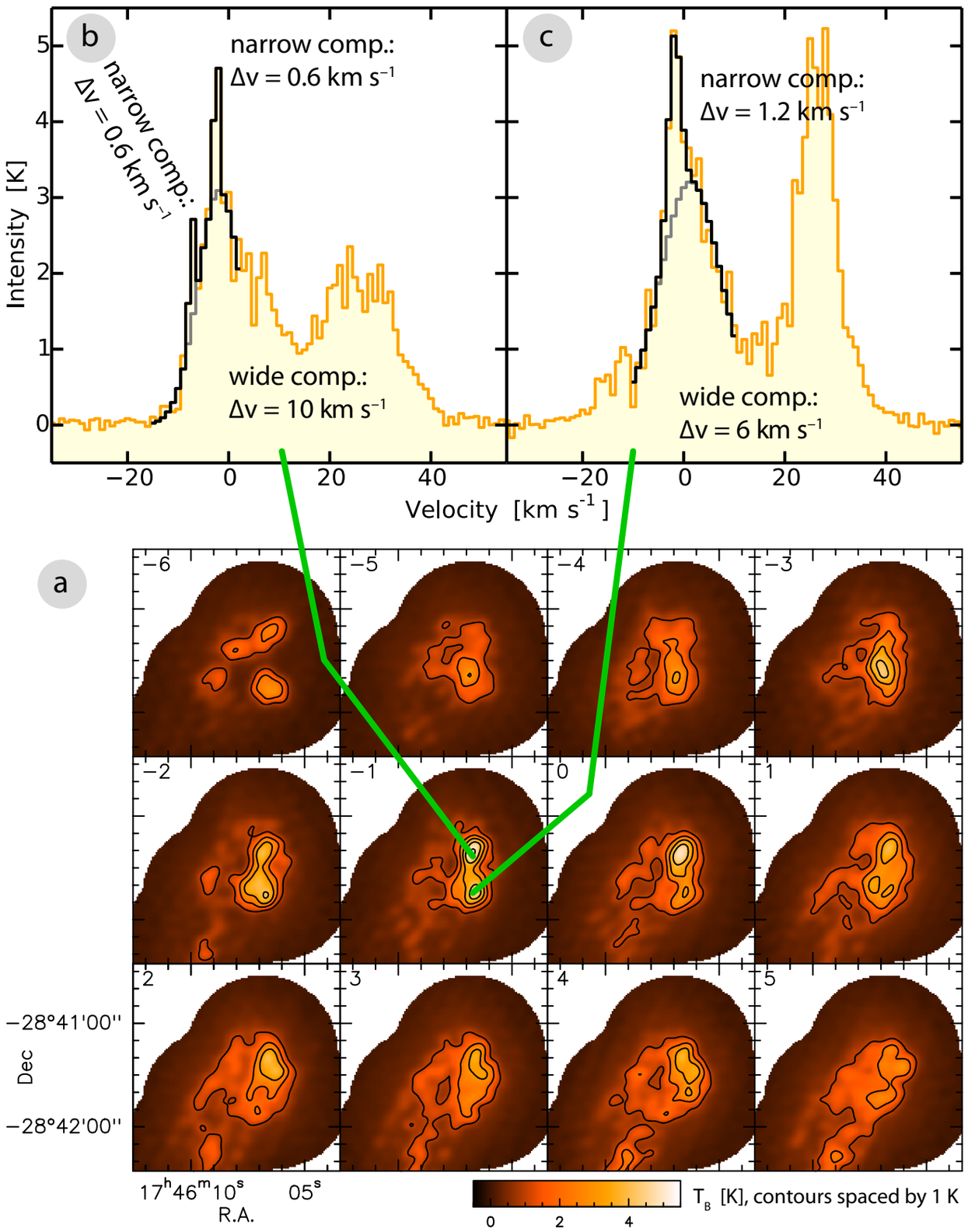
Jens Kauffmann

MIT Haystack Observatory

Haystack Observatory, Massachusetts Institute of Technology

Image Fidelity Workshop, Lorentz Center Leiden • 2019 Aug. 13

Example I: Galactic Center Molecular Clouds with SMA

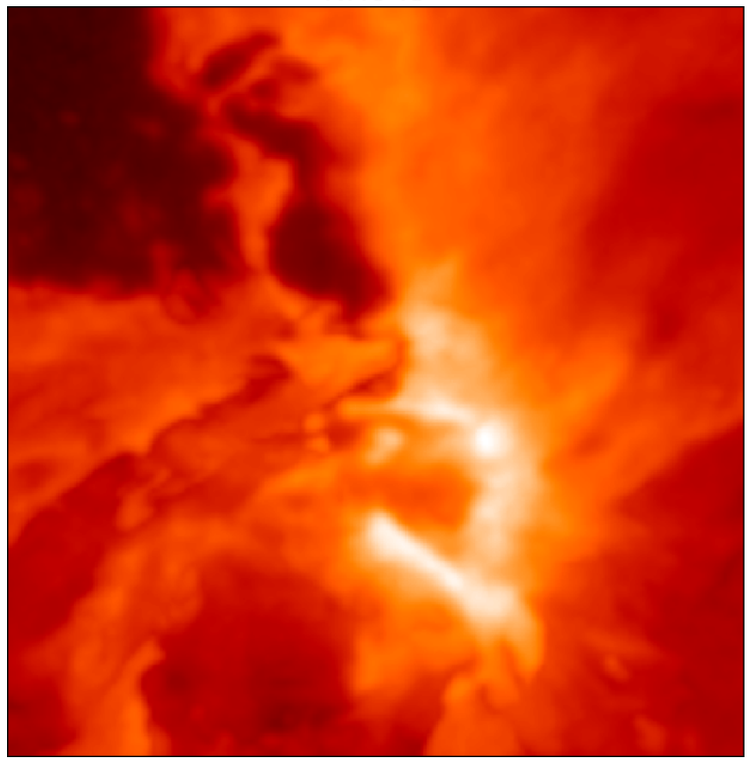


SMA + APEX

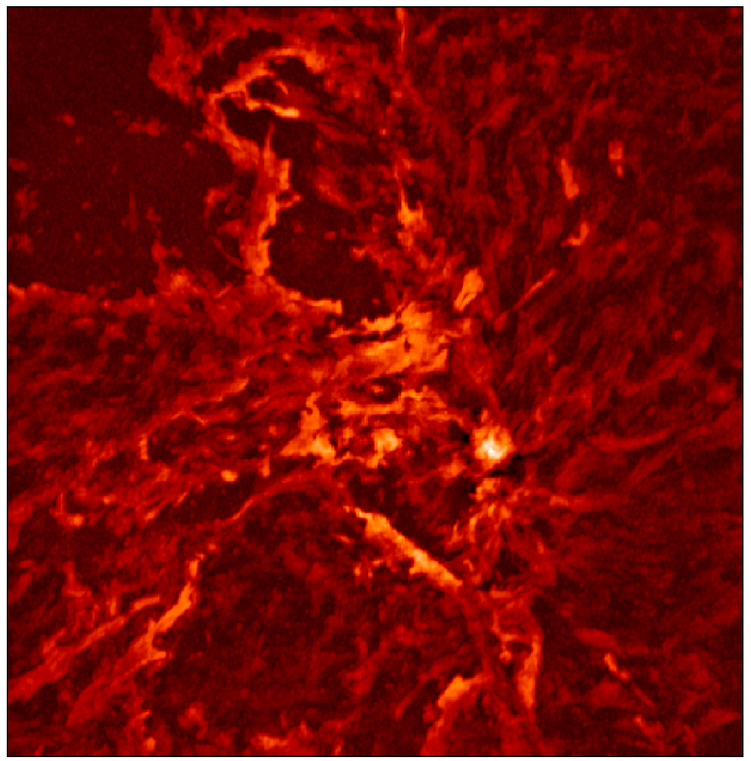
Kauffmann et al. (2017a)

Example II: Orion with CARMA

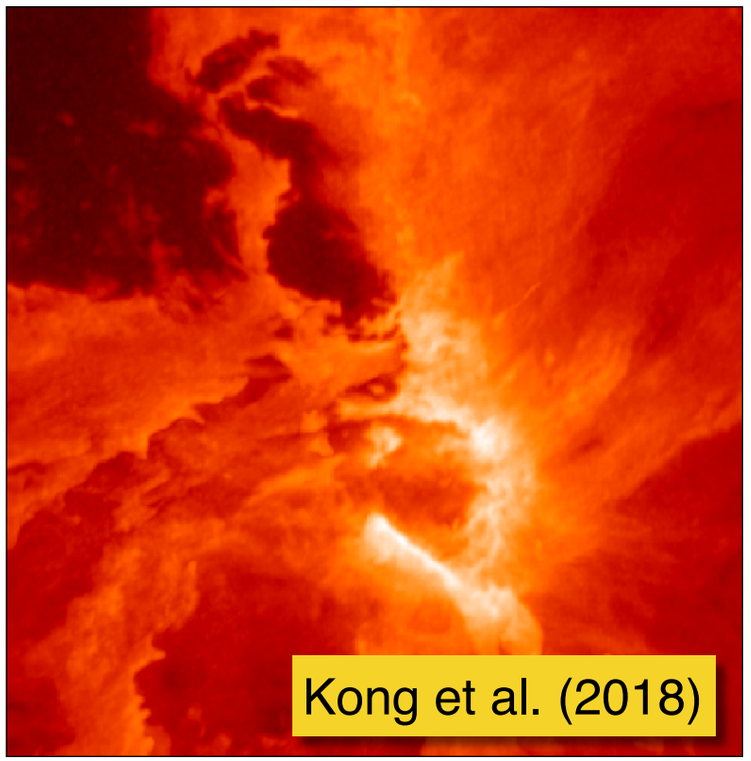
NRO45



CARMA

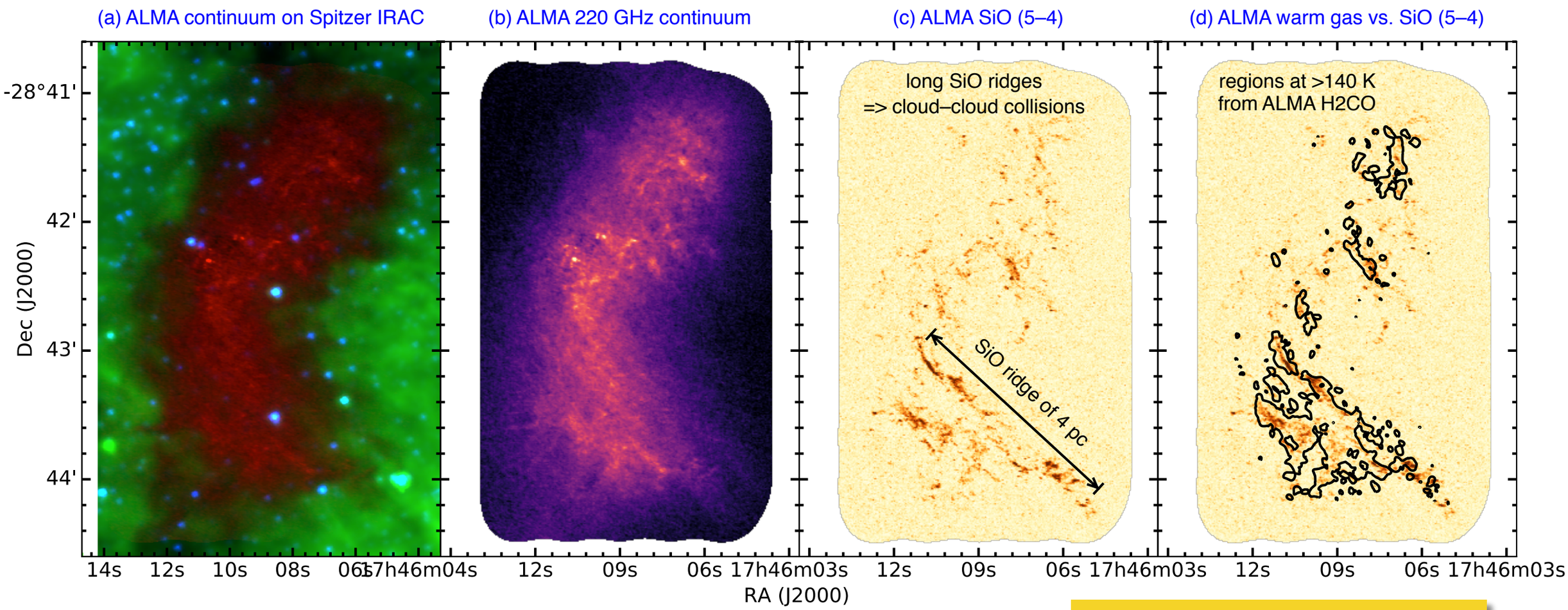


NRO45+CARMA



CARMA + Nobeyama 45m–telescope

Example III: Galactic Center Molecular Clouds with ALMA

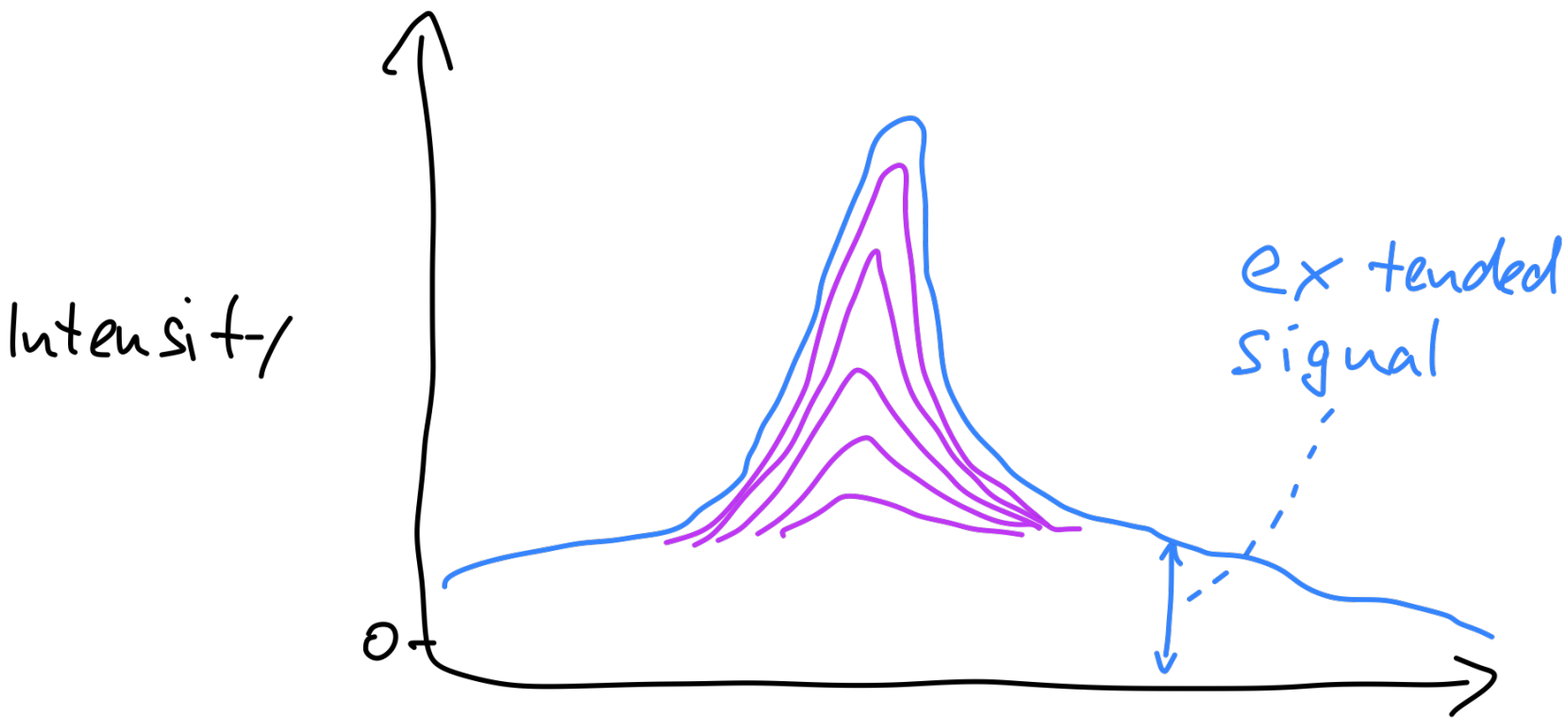


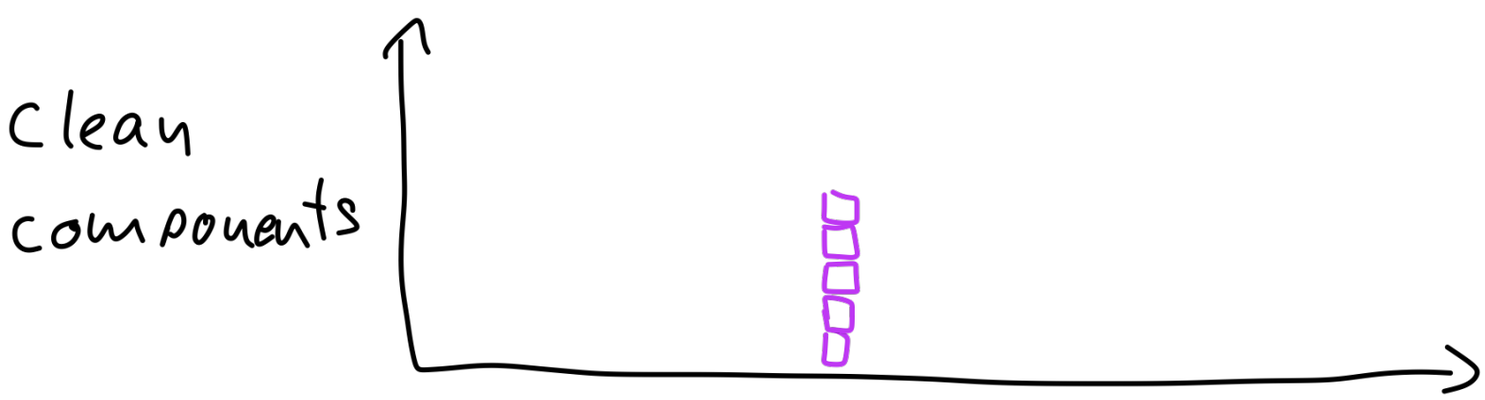
Kauffmann et al. (in prep.)

ALMA 12m + ALMA ACA + APEX

Cleaning Extended Emission

Cleaning Emission





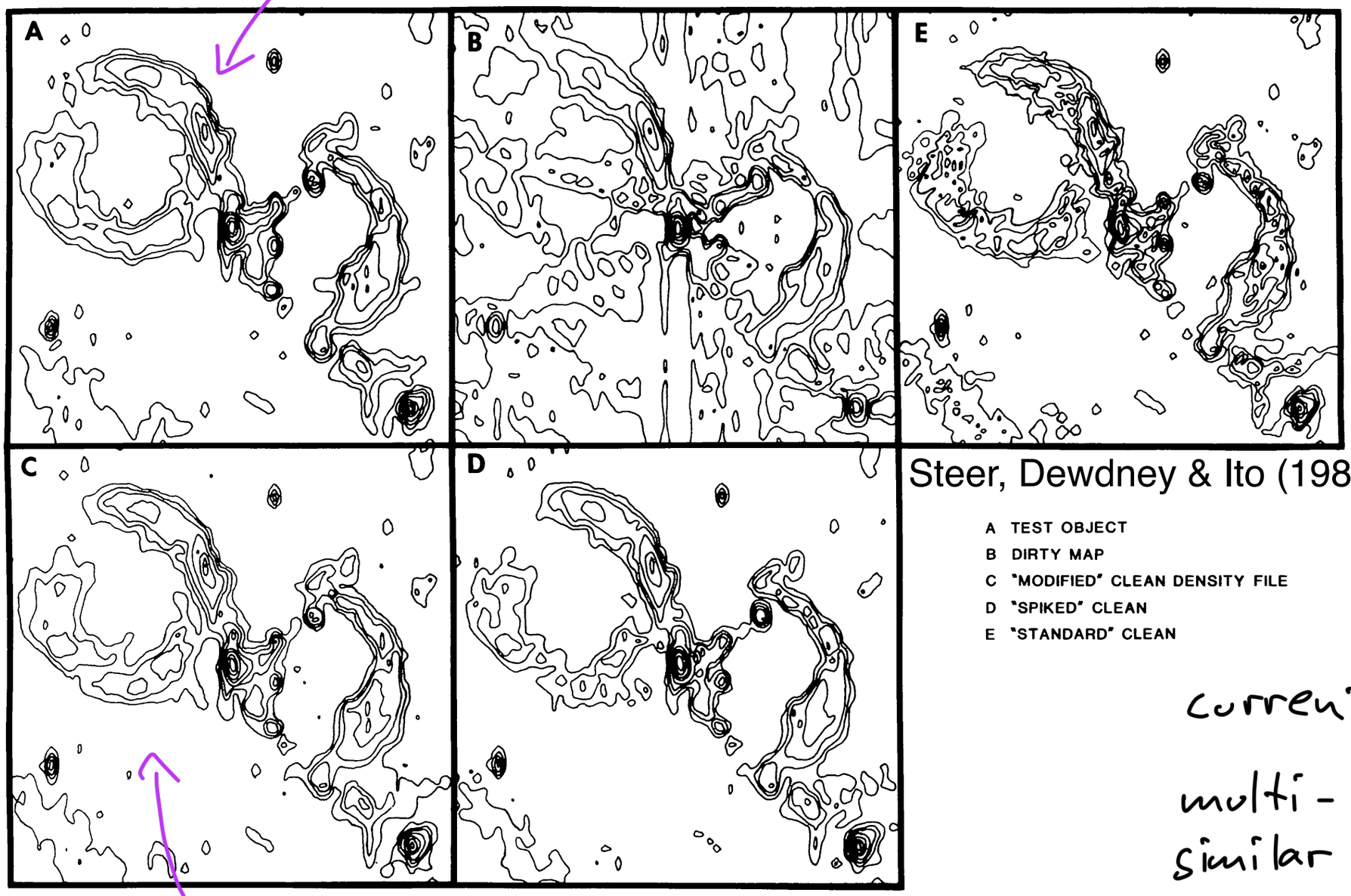
problem:
extended emission not well represented by δ-functions

⇒ ① modify clean to subtract ext. comp. (Steer, Dewdney & Ito 1984)

② subtract (estimated) model of extended emission from data before clean

Steer–Dewdney–Ito Clean

original



Steer, Dewdney & Ito (1984)

A TEST OBJECT
B DIRTY MAP
C "MODIFIED" CLEAN DENSITY FILE
D "SPIKED" CLEAN
E "STANDARD" CLEAN

currently not in CASA (?)

multi-scale might offer similar capabilities

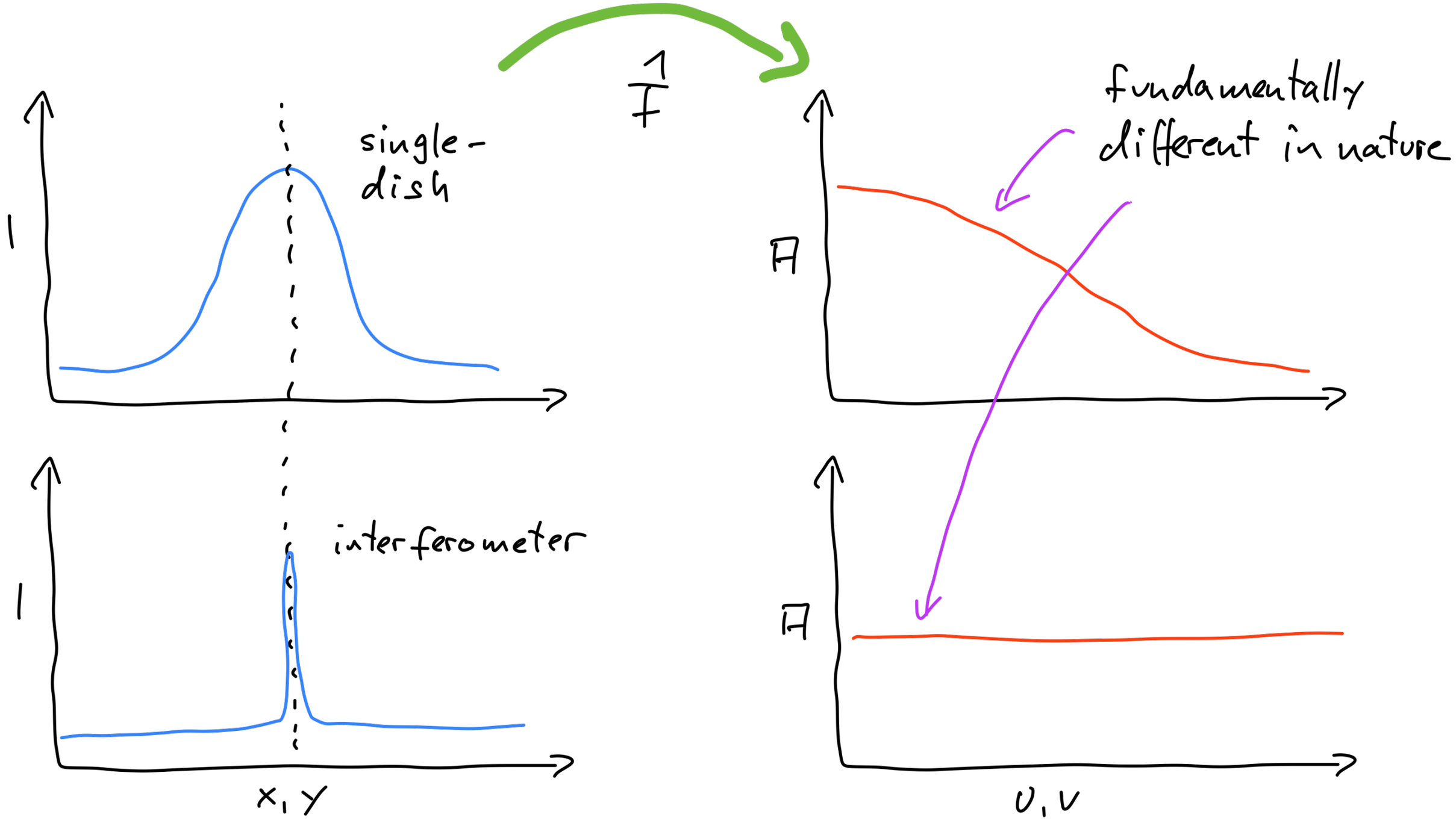
steer clean

substantial residuals in MIRIAD

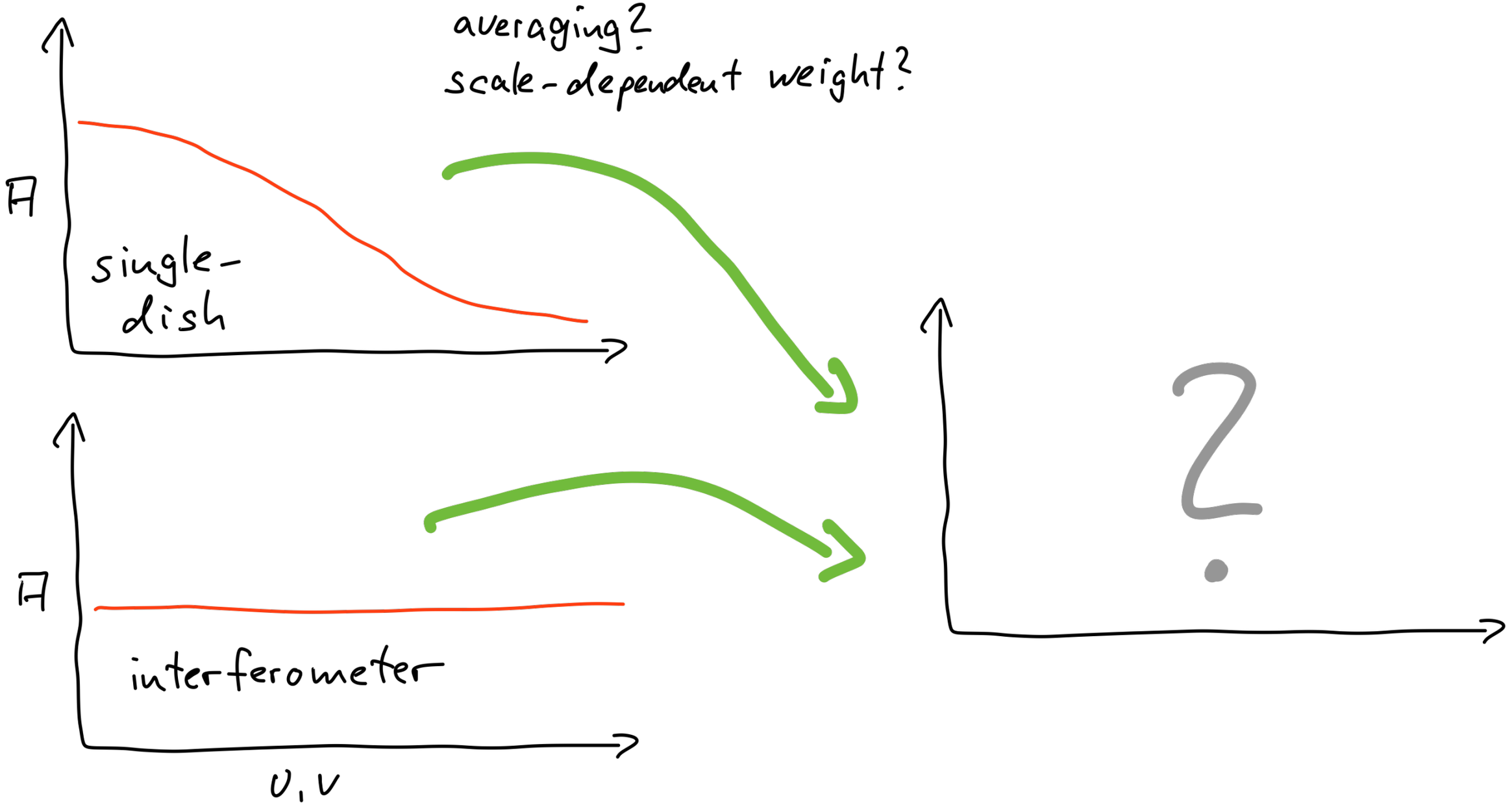
=> merge residual with clean comp.

Point Sources in Zero–Spacing Data

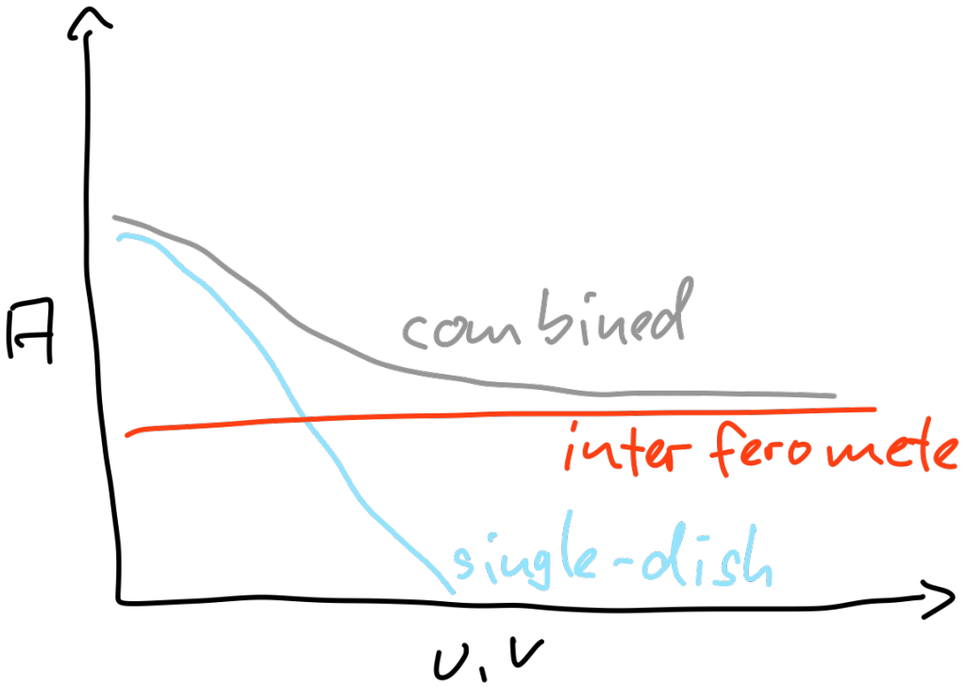
Point Sources in Single–Dish and Interferometer Data

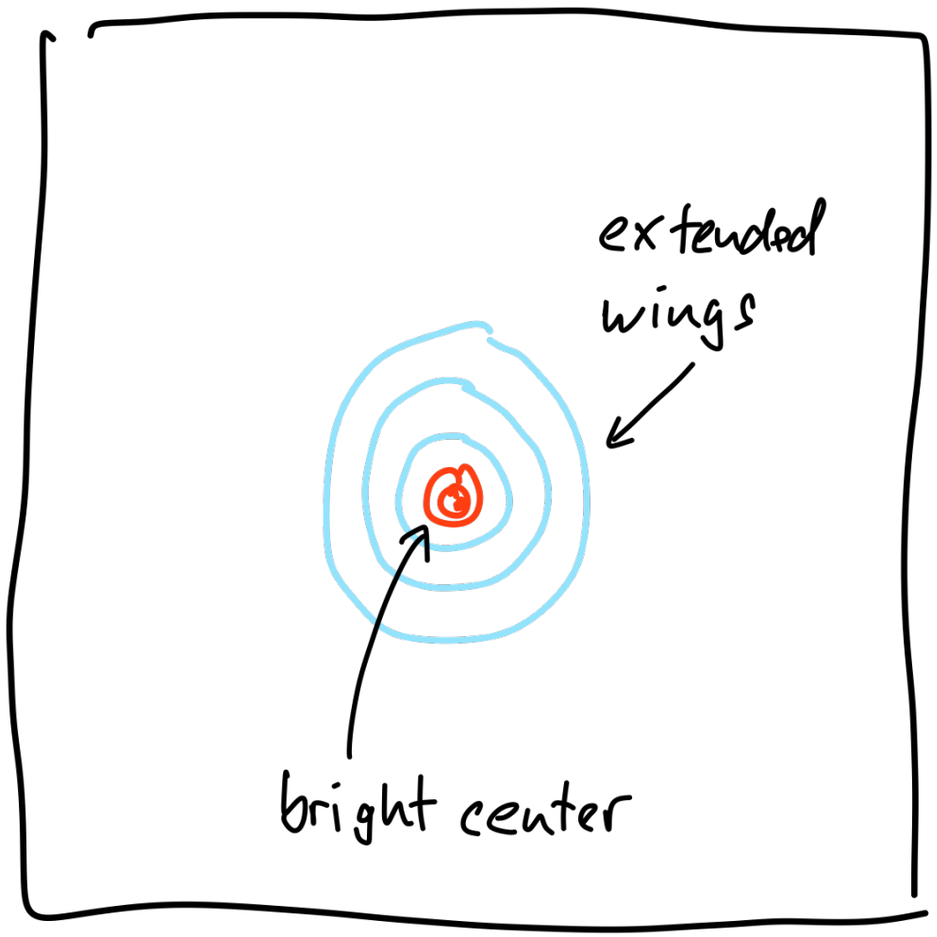


Data Combination



Unfortunate Combinations

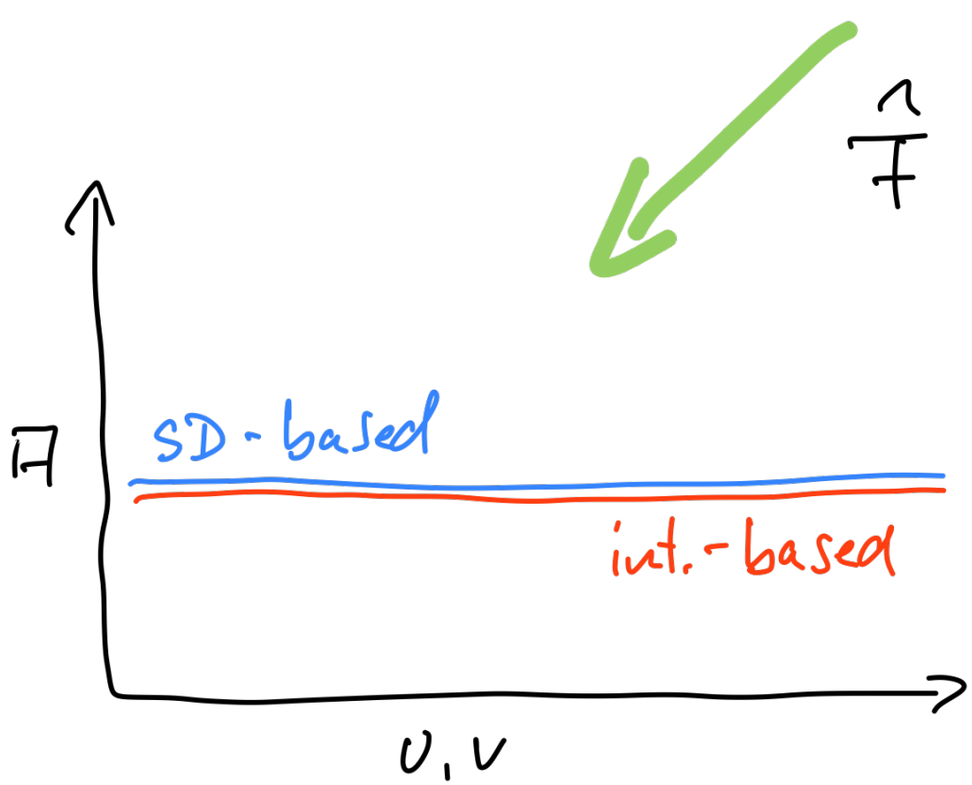




plain combination

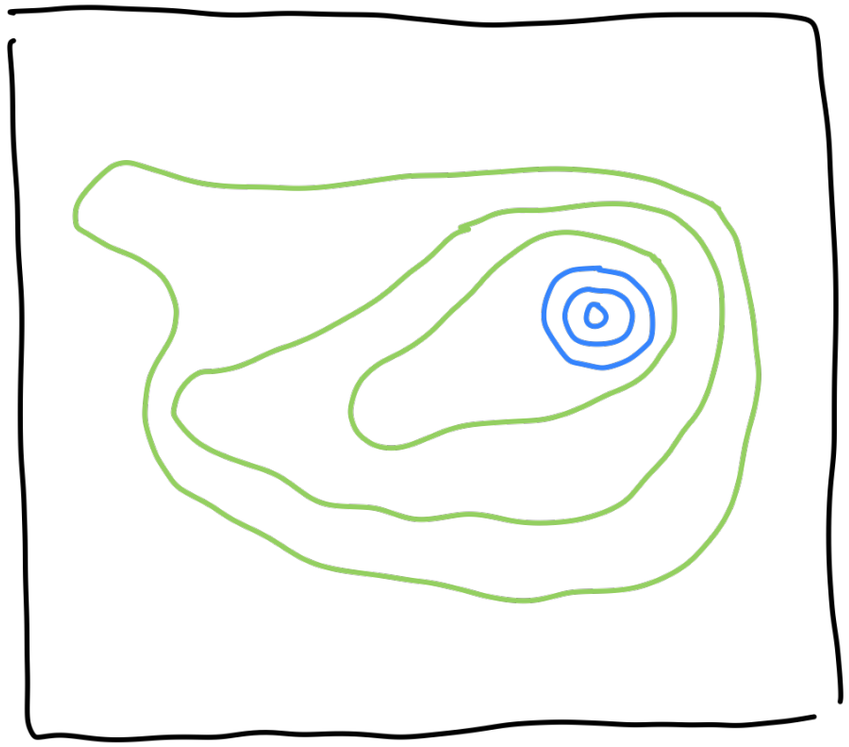
⇒ source represented twice, by compact and extended signal

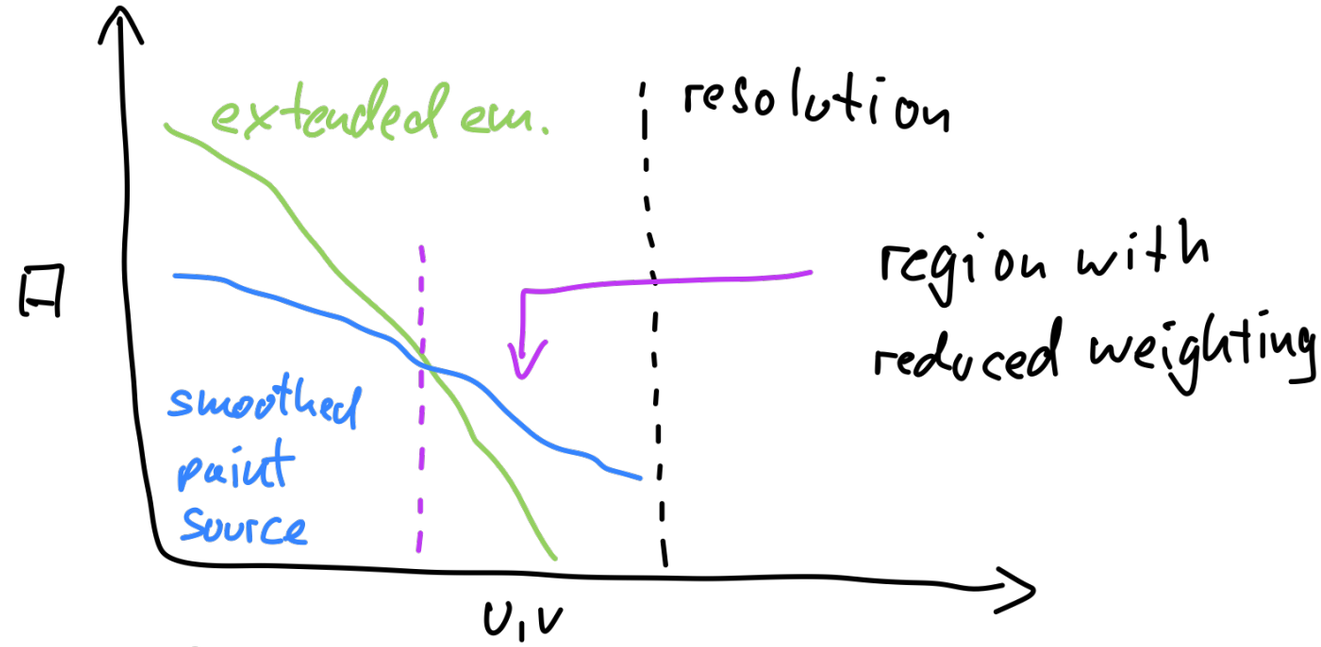
Solution I: Clean Single–Dish Data



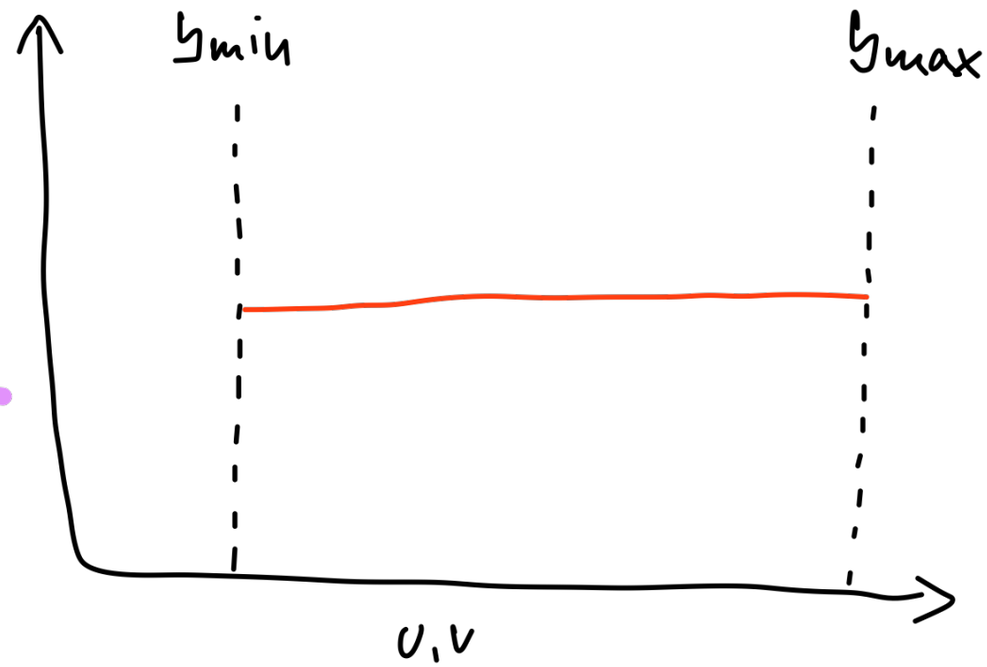
Solution II: Scale–Dependent Weighting of Data

SD : extended and point-like

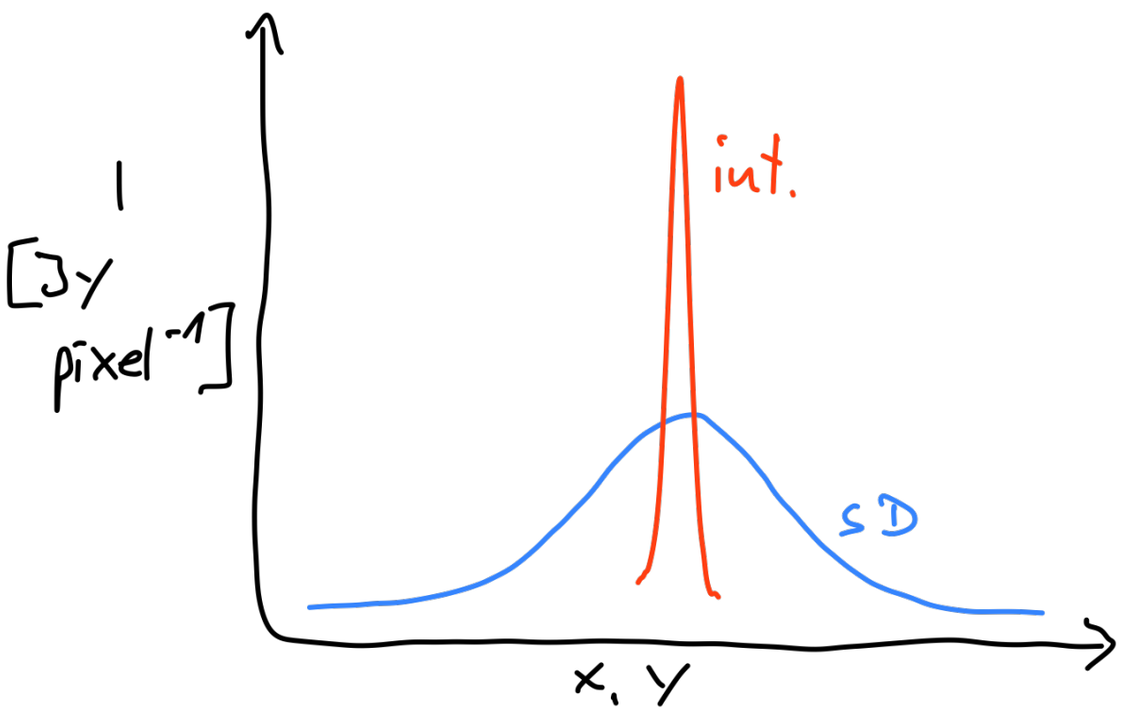


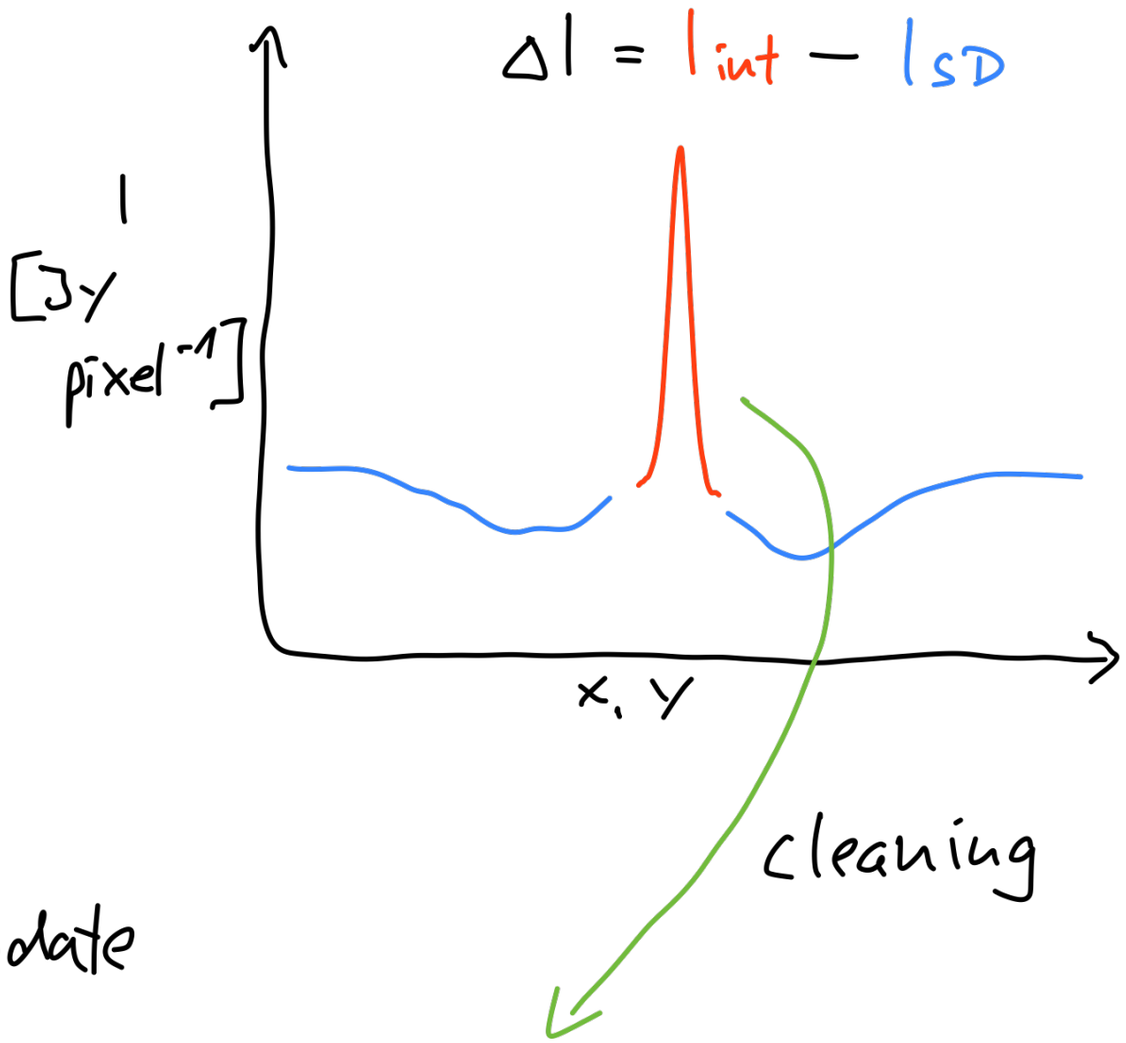


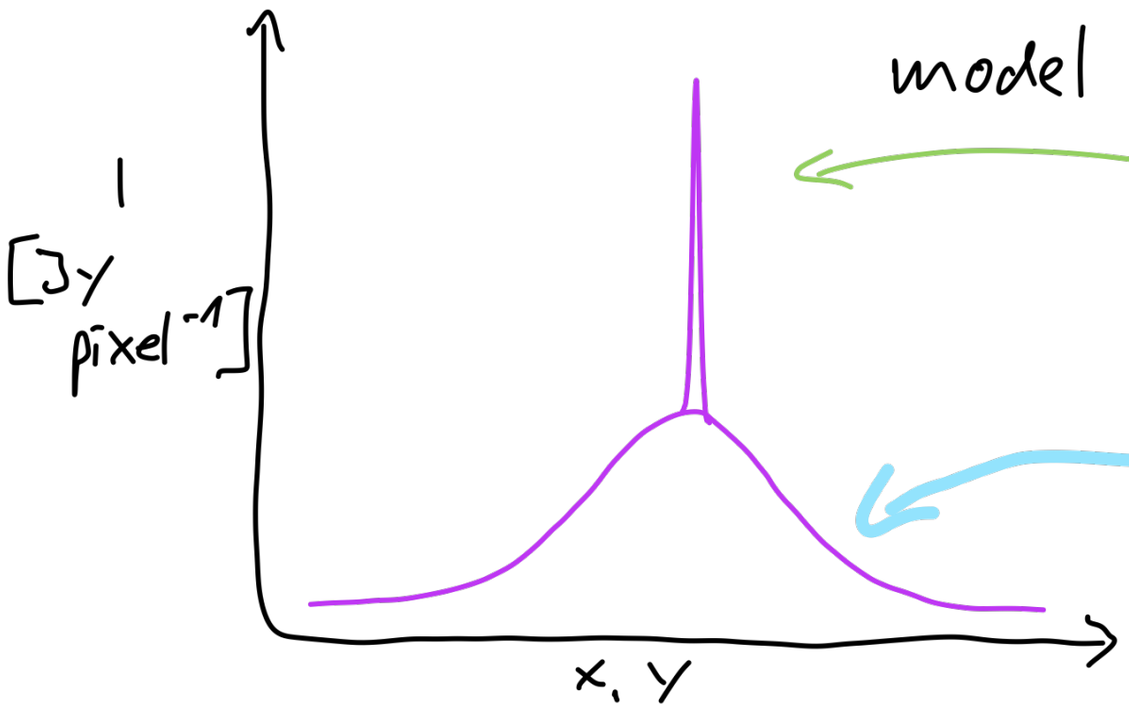
feathering,
joint deconvol.



Solution III: Iterative Model Estimation







model update

components in excess of SD-based model

Not just sum! Feathering, e.g., can reduce ext. signal from point source.

A CASA Implementation

Cleaning with Models in CASA

this material is a bit dated

input uv-data from interferometer

```
clean(vis = '../data_uv/uv_casa/' + UVFileNaming + '.ms',
      imagename = 'deconvolved-sdinput',
      modelimage = 'apex_trans_regrid_scaled_unmasked_noneg.image',
      ...)
```

input model

output image
(this is what we get)

see http://tinyurl.com/zero-spacing for details

Getting Single–Dish Data into Shape

```
#
# prepare single-dish data
#

# convert APEX map
importfits(fitsimage = '../data_sd/' + SDImage,
           imagename='apex.image',
           overwrite=True)
imhead('apex.image',
       mode='put',
       hdkey='telescope', hdvalue='ALMA')


# smooth the velocity axis of the APEX data
SDCellSizeX = str(abs(imhead('apex.image', mode='get', hdkey='cdelt1')['value'] * \
                      180.*3600./pi)) + 'arcsec'
SDCellSizeY = str(abs(imhead('apex.image', mode='get', hdkey='cdelt2')['value'] * \
                      180.*3600./pi)) + 'arcsec'

os.system('rm -rf apex_vsmooth.image')
ia.open('apex.image')
im2 = ia.sepconvolve(outfile='apex_vsmooth.image',
                     axes=[0,1,2], types=['box','box','box'],
                     widths=[SDCellSizeX,SDCellSizeY,FrequencyResolution],
                     overwrite=True)
im2.done()
ia.close()


# beat the stokes axis into proper format
os.system('rm -rf apex_vsmooth_stokes.image')
ia.open('apex_vsmooth.image')
im2 = ia.regrid(dropdeg=True)
im3 = im2.adddegaxes(outfile='apex_vsmooth_stokes.image', overwrite=True,
                     stokes='I')
im2.done()
im3.done()
ia.close()

imhead('apex_vsmooth_stokes.image')

os.system('rm -rf apex_trans.image')
imtrans(imagename='apex_vsmooth_stokes.image',
        outfile='apex_trans.image',
        order='0132')

imhead('apex_trans.image')


# regrid single-dish image
os.system('rm -rf apex_trans_regrid.image')
ReferenceFrame = imregrid('deconvolved.image', template='get')
imregrid(imagename='apex_trans.image',
         template=ReferenceFrame,
         output='apex_trans_regrid.image',
         asvelocity=True,
         interpolation='linear',
         overwrite=True)

imhead('apex_trans_regrid.image')

imhead('deconvolved.image')


# flux conversion scaling for single-dish image
JyperKelvin = 0.817 * (RestFrequencyGHz/100.)**2. * \
              (SingleDishResolutionArcsec/10.)**2.
FactorJyperPixel = CellSizeArcsec**2. / \
                   (1.133 * SingleDishResolutionArcsec**2.)
ConversionFactor = FudgeScale * JyperKelvin * FactorJyperPixel / SDEfficiency

os.system('rm -rf apex_trans_regrid_scaled.image')
im1 = ia.imagecalc(outfile='apex_trans_regrid_scaled.image',
                   pixels=str(ConversionFactor)+'*apex_trans_regrid.image',
                   overwrite=True)
im1.done()
ia.close()

imhead('apex_trans_regrid_scaled.image',
       mode='put',
       hdkey='bunit', hdvalue='Jy/pixel')


# remove blanking for single-dish data
os.system('rm -rf apex_trans_regrid_scaled_unmasked.image')
os.system('cp -r apex_trans_regrid_scaled.image ' +
          'apex_trans_regrid_scaled_unmasked.image')
ia.open('apex_trans_regrid_scaled_unmasked.image')
ia.replacemaskedpixels(0., update=True)
ia.close()


# remove negative values in single-dish image
os.system('rm -rf apex_trans_regrid_scaled_unmasked_noneg.image')
immath(outfile='apex_trans_regrid_scaled_unmasked_noneg.image',
       imagename=['apex_trans_regrid_scaled_unmasked.image'],
       mode='evalexpr',expr='iif(IM0 >= 0.00, IM0, 0.0)')
```

read SD data into CASA

grid SD data into velocity channels of interferometer data

data formatting...

grid SD data into pixels of interferometer image

scale SD data into Jy pixel^{-1}

unblank SD data

remove negative intensities in SD data

Calculating an improved Source Model

```
#
# produce a first model of combined interferometer and SD data
#

# clean the UV data
os.system('rm -rf deconvolved-sdinput*')
clean(vis = '../data_uv/uv_casa/' + UVFileNaming + '.ms',
      imagename = 'deconvolved-sdinput',
      modelimage = 'apex_trans_regrid_scaled_unmasked_noneg.image',
      spw='',
      restfreq = RestFrequency,
      imagermode = 'mosaic',
      mode = 'velocity',
      outframe='lsrk',
      width = VelocityResolution,
      start = StartVelocity,
      nchan = NumberChannels,
      interactive = False,
      imsize = ImageSize,
      cell = CellSize,
      phasecenter = SourceLocation,
      weighting = 'briggs',
      niter = NumberIterations,
      cyclefactor = 5.,
      threshold = CleaningThreshold,
      usescratch = True)


# get the positive interferometer-only clean components
os.system('rm -rf deconvolved-sdinput.intmodel')
immath(outfile='deconvolved-sdinput.intmodel',
   imagename=['deconvolved-sdinput.model',
              'apex_trans_regrid_scaled_unmasked_noneg.image'],
   mode='evalexpr',
   expr='iif((IM0-IM1) >= 0.00, IM0-IM1, 0.0)')


# remove those components if they are at the map edge
os.system('rm -rf deconvolved-sdinput-masked.intmodel')
immath(outfile='deconvolved-sdinput-masked.intmodel',
   imagename=['deconvolved-sdinput.intmodel',
              'deconvolved-sdinput.flux.pbcoverage'],
   mode='evalexpr',
   expr='iif((IM1) >= 0.25, IM0, 0.0)')

imhead('deconvolved-sdinput-masked.intmodel',
       mode='put',
       hdkey='bunit', hdvalue='Jy/pixel')


# smooth the interferometer-only components to the synthesized beam
SynthBeamMaj = imhead('deconvolved-sdinput.image', mode='get', hdkey='bmaj')['value']
SynthBeamMin = imhead('deconvolved-sdinput.image', mode='get', hdkey='bmin')['value']
SynthBeamPA = imhead('deconvolved-sdinput.image', mode='get', hdkey='bpa')['value']

os.system('rm -rf deconvolved-sdinput.intimage')
imsmooth(imagename='deconvolved-sdinput-masked.intmodel',
         outfile='deconvolved-sdinput.intimage',
         kernel='gauss',
         major=str(SynthBeamMaj)+'arcsec',
         minor=str(SynthBeamMin)+'arcsec',
         pa=str(SynthBeamPA)+'deg')


# produce a non-negative single-dish map in Jy/beam
os.system('rm -rf apex_trans_regrid_scaled_unmasked_noneg_perbeam.image')
immath(outfile='apex_trans_regrid_scaled_unmasked_noneg_perbeam.image',
   imagename=['apex_trans_regrid_scaled_unmasked_noneg.image'],
   mode='evalexpr', expr='IM0/'+str(FactorJyperPixel))

imhead('apex_trans_regrid_scaled_unmasked_noneg_perbeam.image',
       mode='put',
       hdkey='bunit', hdvalue='Jy/beam')
imhead('apex_trans_regrid_scaled_unmasked_noneg_perbeam.image',
       mode='put',
       hdkey='bmaj', hdvalue=str(SingleDishResolutionArcsec)+'arcsec')
imhead('apex_trans_regrid_scaled_unmasked_noneg_perbeam.image',
       mode='put',
       hdkey='bmin', hdvalue=str(SingleDishResolutionArcsec)+'arcsec')
imhead('apex_trans_regrid_scaled_unmasked_noneg_perbeam.image',
       mode='put',
       hdkey='bpa', hdvalue='0deg')


# feather the data
os.system('rm -rf deconvolved-combi.model')
feather(imagename = 'deconvolved-combi.model',
        highres = 'deconvolved-sdinput.intimage',
        lowres = 'apex_trans_regrid_scaled_unmasked_noneg_perbeam.image',
        effdishdiam = SDCutoffScaleMeter,
        lowpassfiltersd = True)


# clean up and keep the final model
os.system('rm -rf combimodel.image')
os.system('mv deconvolved-combi.model combimodel.image')
```

clean interferometer data with input mode

identify clean components not in SD model map

remove artifacts at outskirts of map

smooth interferometer-only clean components to synthesized beam

produce non-negative map of SD data in Jy beam^{-1}

feather SD data and interferometer-only components

filenaming

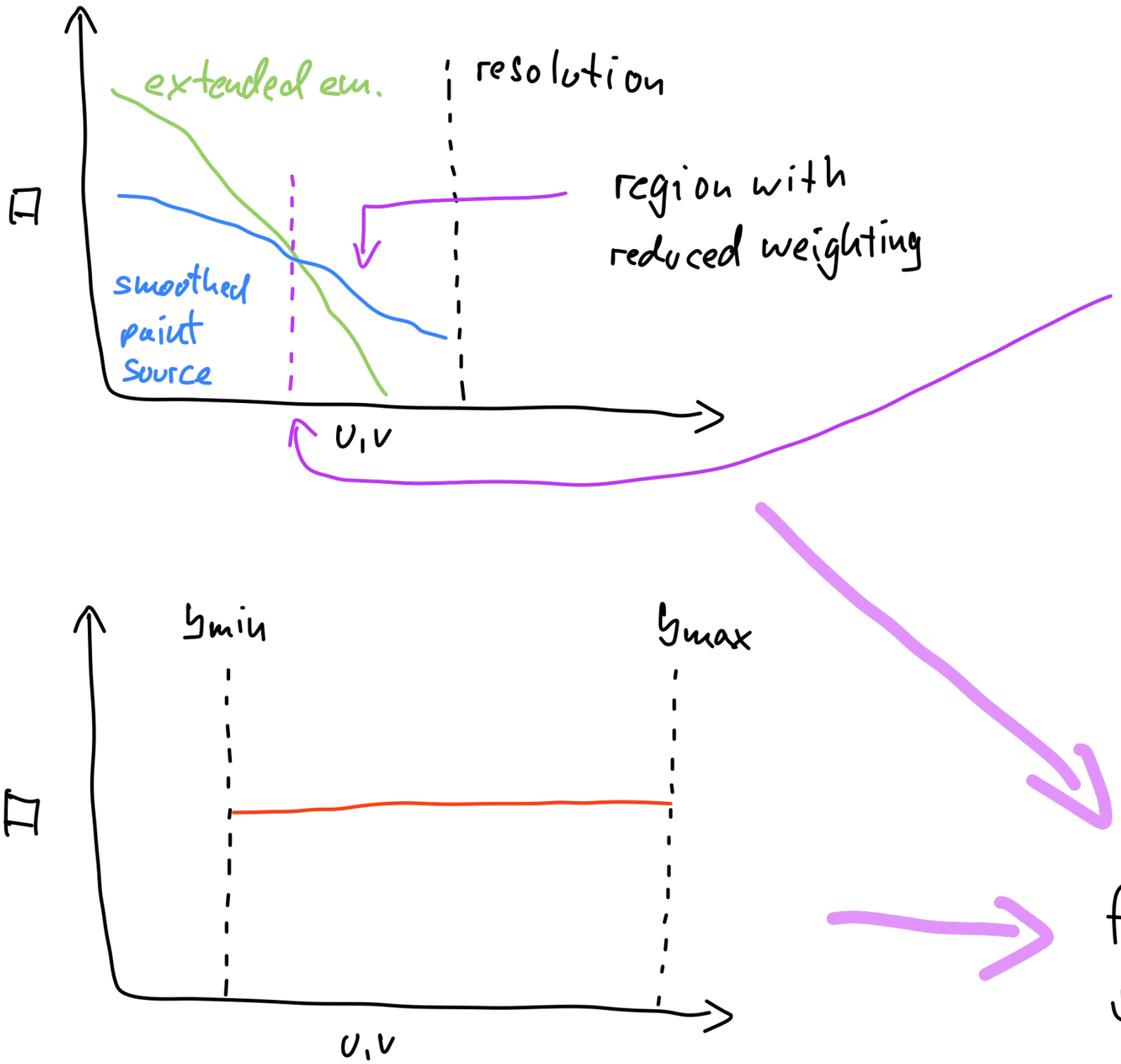
Final Cleaning

```
#
# second and final cleaning with new model
#

# clean the UV data
os.system('rm -rf deconvolved-sdinput*')
clean(vis = '../data_uv/uv_casa/' + UVFileNaming + '.ms',
      imagename = 'deconvolved-sdinput',
      modelimage = 'combimodel.image',
      spw='',
      restfreq = RestFrequency,
      imagermode = 'mosaic',
      mode = 'velocity',
      outframe='lsrk',
      width = VelocityResolution,
      start = StartVelocity,
      nchan = NumberChannels,
      interactive = False,
      imsize = ImageSize,
      cell = CellSize,
      phasecenter = SourceLocation,
      weighting = 'briggs',
      niter = NumberIterations,
      cyclefactor = 5.,
      threshold = CleaningThreshold,
      usescratch = True)

viewer('deconvolved-sdinput.image')
```

further cleaning, now with improved model as input

Other Lessons

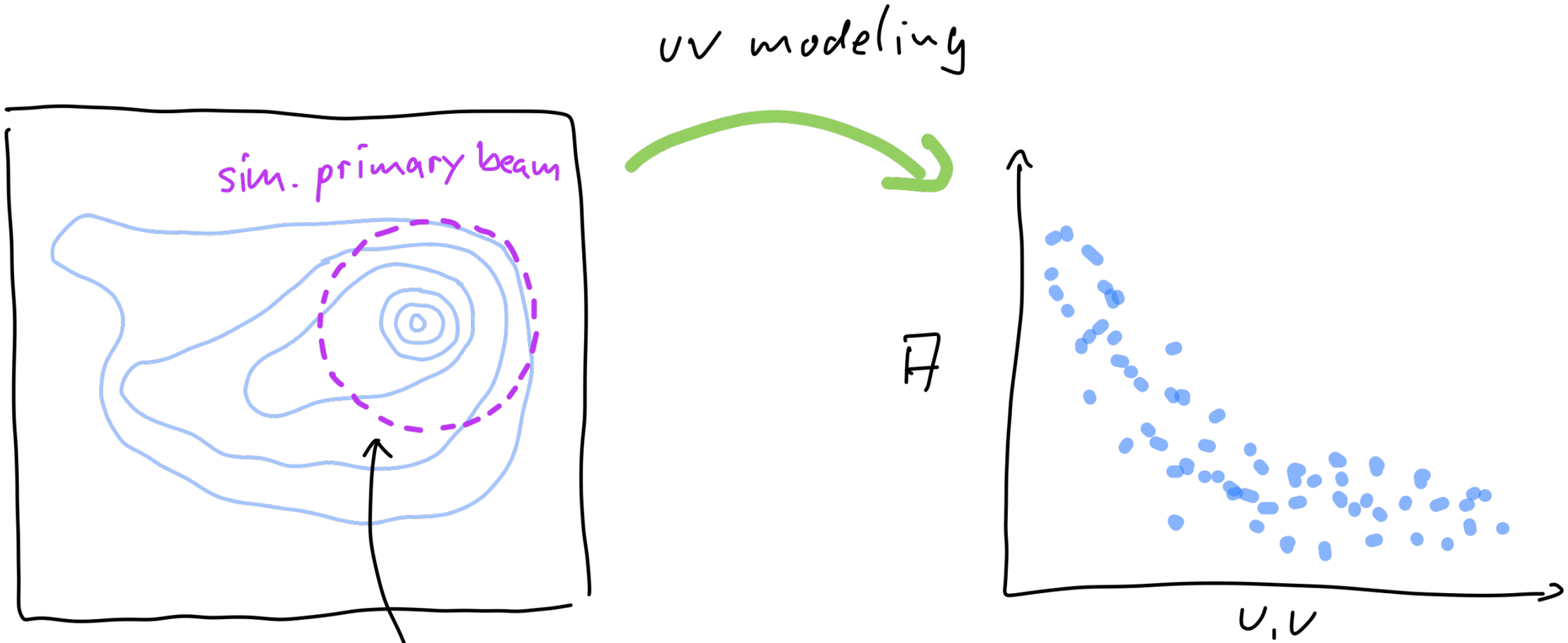
Transition Scale in Feather: Not set by Instrumentation



example for scale not strictly set by telescope parameters

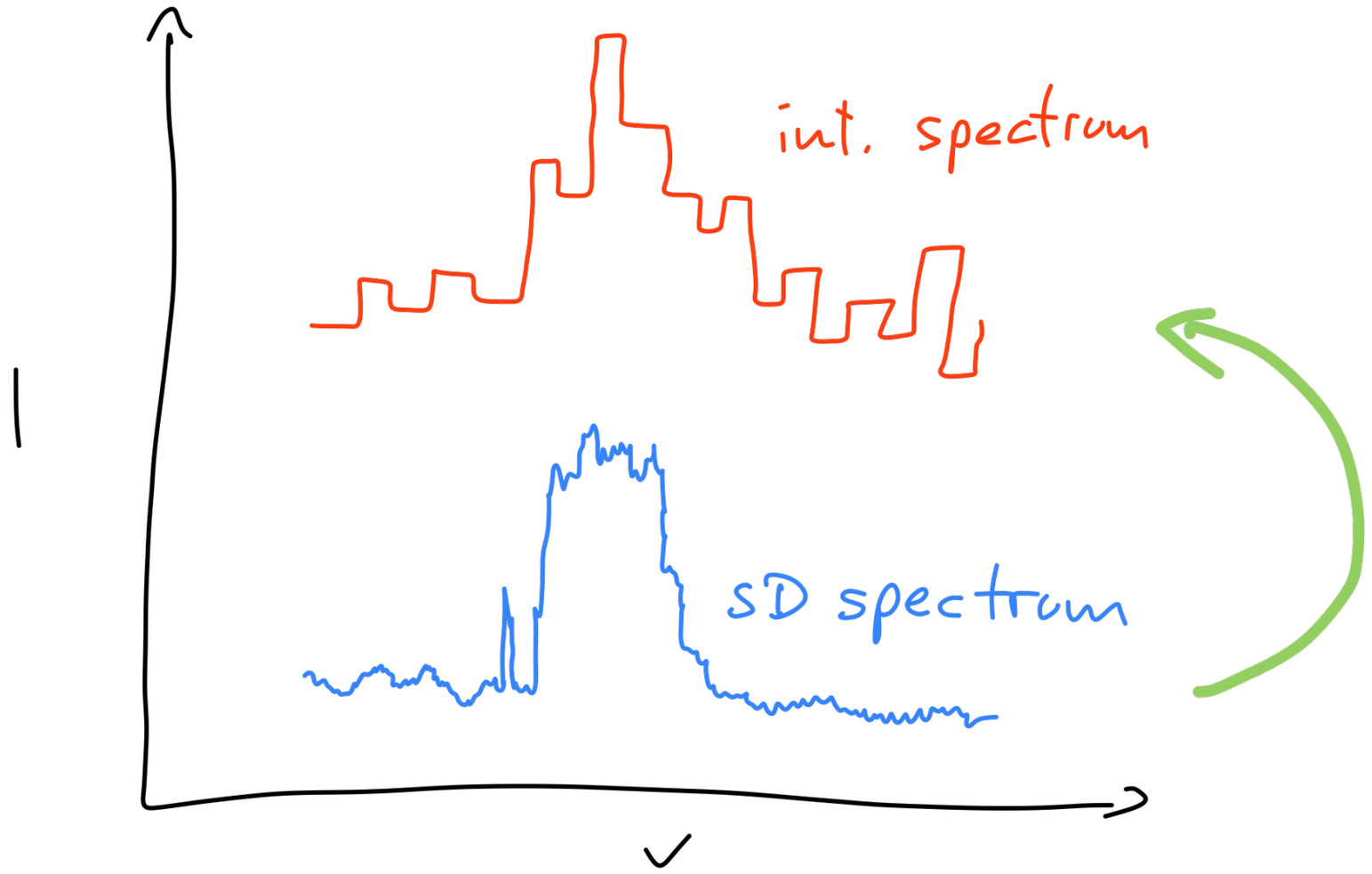
feathering,
joint deconvol.

Primary Beams in Simulated uv–Data



no definitive choice for size,
in particular in heterogeneous arrays

Sampling of Velocity Channels



Sampling to interferometer gridding

not identical to simple Gaussian smoothing to lower resolution

Remain Alert & Check Intermediate Products

often artifacts in practical experiments:

* unexpected behavior
* bona-fide bugs

⇒ * remain vigilant
* check intermediate products

impact on community:

* make sure pipelines produce decent data
* but alert users that non-standard useage can produce spectacular failures