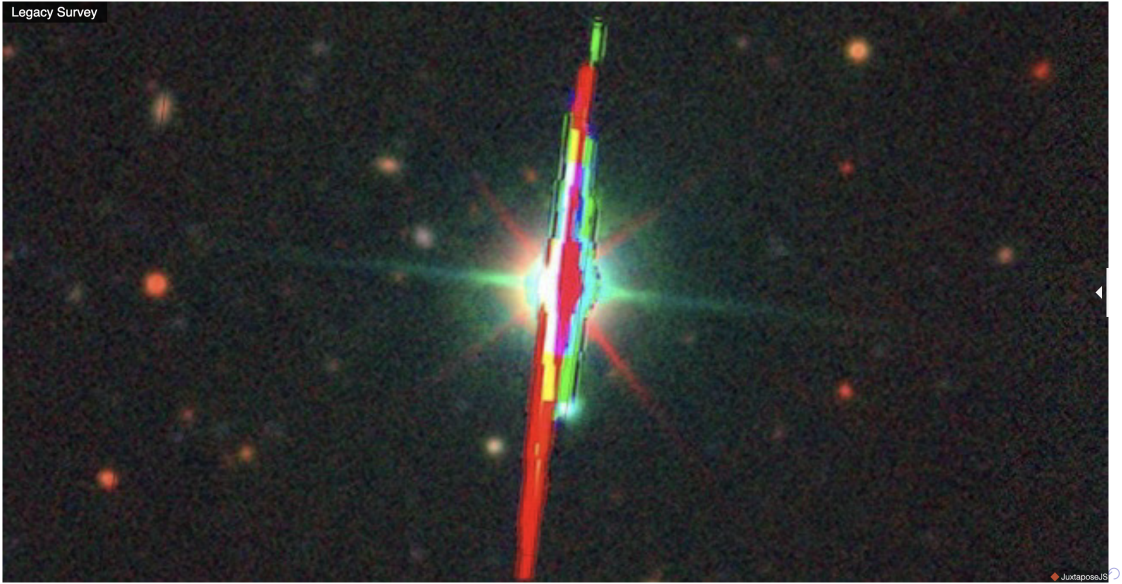
Finding and Seeing Structure in Images

Deconvolution & Stretching

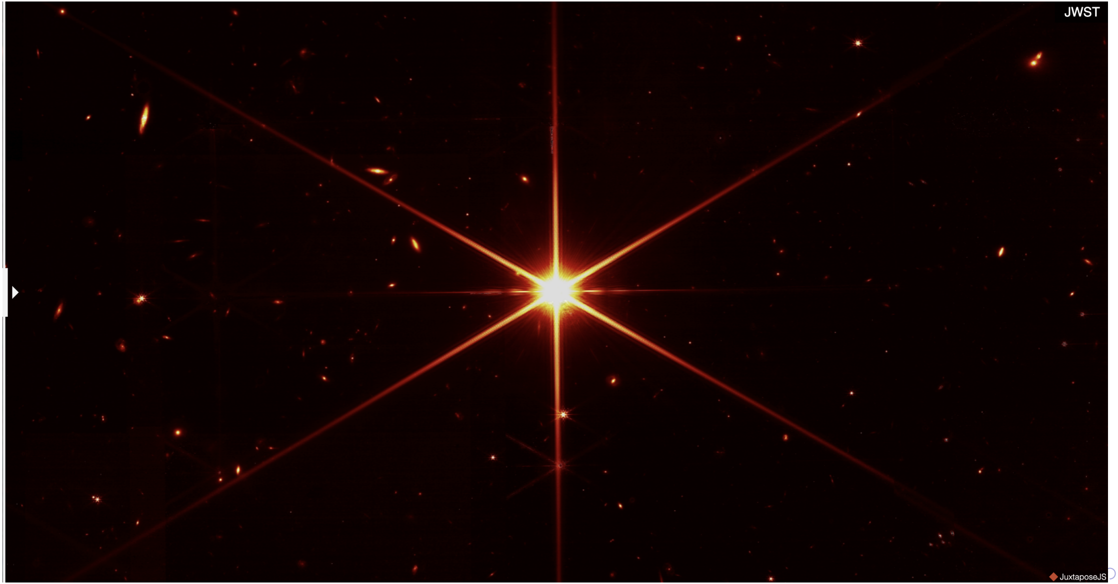
February 13, 2024

University of Rochester

The effect of seeing



The effect of seeing



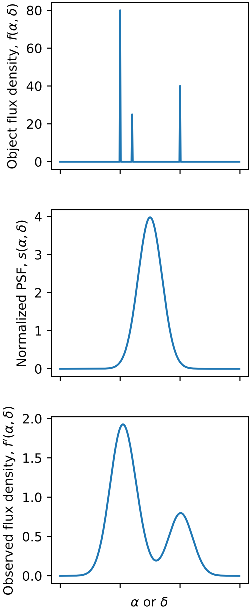
Sharpening images: deconvolution

If your images have very high S/N, you can recover some of the angular resolution that was lost from our normally not-very-good seeing.

- Atmospheric turbulence broadens what should look like a diffraction spot, in the manner of convolution by a Gaussian:

$$f'(\alpha,\delta) = \iint f(x,y)s(\alpha - x, \delta - y)\, dx\, dy$$
$$\cong f(\alpha,\delta) * s(\alpha,\delta)$$
$$s(x,y) = \frac{1}{2\pi\sigma^2}\exp\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

  where s is the point-spread function (PSF), a.k.a. "seeing disk."
- At Mees, the diffraction limit is 0.22 arcsec in the G filter, while the seeing is usually 2 arcsec or so.



Deconvolution

The best way to eliminate the blurring effects of seeing — which are due to different phase shifts of light that takes slightly different paths through the atmosphere — are to

- Put the telescope in outer space (VERY expensive)
- Correct the phases in real time with the help of a reference object in the same field of view as the target. This method is called adaptive optics (brief intro here).

The first of these methods is the only one that works perfectly.

- Adaptive optical systems are complex and expensive.
- They are even more complex and expensive if a field much bigger than about an arcminute needs correction. (Recall that our system is 15.4 arcmin square.)
- Currently, they do not work well at visible wavelengths. At least those used by astronomers.

Deconvolution

The blur can be ameliorated a bit in the data processing.

- If we knew exactly what the point-spread function s is, and if there was no such thing as noise or systematic error, then we could determine the object's flux density unambiguously because of the convolution theorem:

Convolution theorem

If $f'(\alpha,\delta) = f(\alpha,\delta) * s(x,y)$, and F', F, and S are respectively the Fourier transforms of f', f, and s, then $F'(u,v) = F(u,v)S(u,v)$.

- Thus, we could deconvolve the image:
  - Fourier-transform the observations (f') and the point-spread function (s), divide the two results to produce F, then Fourier-transform F to produce the object's real flux density f, unsmeared by seeing.
  - Which would be limited by diffraction, of course.

Deconvolution

However, life is rarely so simple. There is noise and systematic error, and it messes everything up. Separating the measurable functions into noiseless/error-free terms and noise/error, as

$$f'(\alpha, \delta) + \Delta f'(\alpha, \delta) = f(\alpha, \delta) * [s(\alpha, \delta) + \Delta s(\alpha, \delta)]$$

we see that the convolution theorem could still help, but additional constraints on the noise terms would be necessary.

- There are rarely enough constraints to do this algebraically, e.g. N equations in N unknowns solvable by linear-algebra techniques.

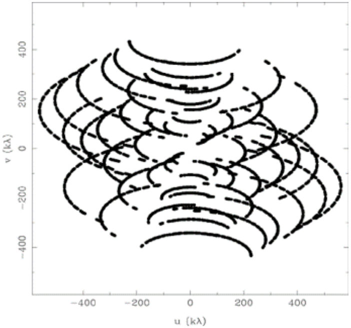
If we know what our PSF shape is, and we measure the structures that are ever so much broader than that at high S/N, we are justified in saying that we know something about those structures on scales smaller than the PSF.

How can we use such observations, in cases in which we cannot do a simple deconvolution?

Aside: How CLEAN works

Classic reference: Hogbom (1974)
Nice compact intro: Wilner (2018), from which some of the following images are taken.

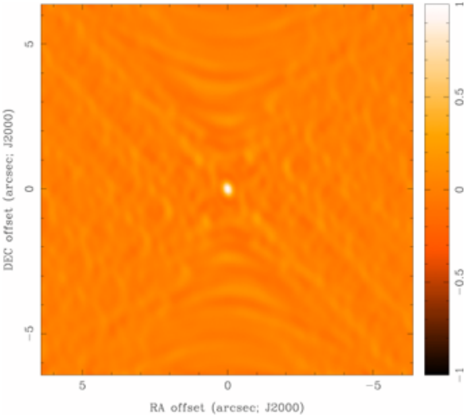
- In radio interferometry, telescopes and instruments measure the amplitude and phase of light from the target field.
- Because both the amplitude and phase are measured, signals from an array of telescopes can be combined as if they were "facets" on one much larger telescope, with a size equal to the separation of the array elements: a larger **aperture** is **synthesized** from the array.



Distances, projected onto the plane of the sky, between pairs of an eight-telescope array (the SMA). Continuous trails result from the changing aspect of each pair's baseline as the target is tracked across the sky. From Wilner (2018).

Aside: How CLEAN works

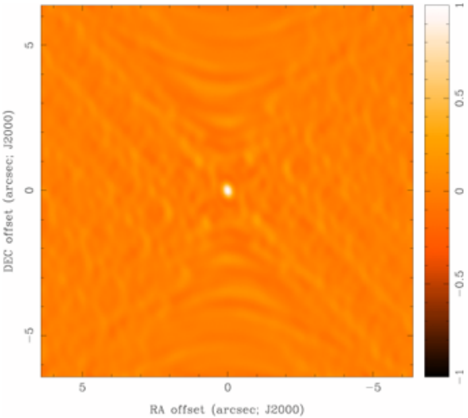
- It turns out that mutual coherence of the signals from two telescopes in the array, which can be calculated from those signals, is one Fourier component of the light-intensity distribution in the target field.
  - This is the Van Cittert-Zernike theorem.
- Thus, the Fourier transform of the mutual coherence of the pairwise signals from all of the telescopes in the array give the image of the target field at the diffraction-limited resolution of the **array size**, not the telescope size.



2D Fourier transform of a uniform-amplitude and phase at those distances, i.e. the point spread function. This is called the dirty beam in radio-astronomy parlance. From Wilner (2018).

Aside: How CLEAN works

- So far, this is not different from how a diffraction-limited telescope would work.
- But, the diffraction pattern of the array of "facets" is not the same as that from a completely filled aperture.
- In particular, the sparse coverage by the telescope array of the synthetic aperture leads to sidelobes:
    - Diffraction peaks that are much brighter, relative to the central peak, than the outer rings are relative to the peak in a single-telescope diffraction pattern.



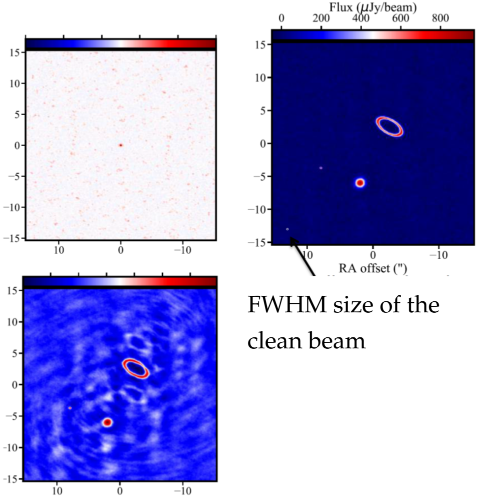
The clutter of sidelobes is why this is called the dirty beam, and why its cure was called CLEAN. From Wilner (2018).

Aside: How CLEAN works

To remove the sidelobes, CLEAN does the following:

1. Locates the bright spot in the dirty image.
2. Subtracts from that spot a dirty-beam-shaped intensity, some fraction (usually 0.1) of the peak intensity.
3. Keeps track of where it subtracted that from. This is the list of CLEAN components.
4. Repeat until a preset brightness limit is reached.
5. Then restore the signal of a clean beam at the position of each clean component, where a clean beam is a Gaussian fit to the main peak of the dirty beam.

The resulting clean image is considered to be the true intensity distribution plus the original noise. (Though, annoyingly, the procedure has never been **proven** to converge on the true intensity distribution.)



Artificial ALMA observations of a model image plus noise, and the result of CELANing (Wilner 2018).

Deconvolution

- Our problem is that we do not know what our dirty beam shape is, noiselessly.
- So, generally, we have too few measurements to solve for all of the unknowns in

$$f'(\alpha,\delta) + \Delta f'(\alpha,\delta) = f(\alpha,\delta) * [s(\alpha,\delta) + \Delta s(\alpha,\delta)]$$

- But not all is lost: we can instead determine the *range* of solutions for f consistent with what constraints there are, and then select among the range for the solution which is most probable.
- This reduces the question to: How does one rank the solutions by probability?
- There is no best way to do this, nor — once again — a way so far which has been proven to converge on the exact solution for f.
- Generally, we proceed by defining a measurable property of the image related to sharpness, and finding the solution corresponding to the maximum or minimum of that property. Routines like this include…

Deconvolution routines

Maximum-likelihood (Lucy-Richardson) deconvolution Related to CLEAN, as it decomposes objects into a sum of PSFs, but it searches for the most likely coefficients in the *restored*-PSF sum under the assumption that they are Poisson-distributed, like shot noise. So it improves the PSF according to the S/N.

Positivity-constrained deconvolution Like Lucy-Richardson, but it rectifies the results as it goes along to prevent the final answer from having unphysically-negative flux densities.

Maximum-entropy (MEM) deconvolution Maximizes the "image entropy" $S = -\sum_i p_i \log_2 p_i$, where p_i is the probability that the difference in DNs between adjacent pixels has the value i.

- The function has nothing to do with physical entropy, $S = -k_B \ln \Omega$; it is named for the resemblance to this formula as the Stirling approximation applies to it. But it is a good analogy to physical entropy, as those who have taken a data-science course know.

Deconvolution

CCDStack v2.9 does positivity-constrained and maximum-entropy deconvolution, and Photoshop CC has all three.

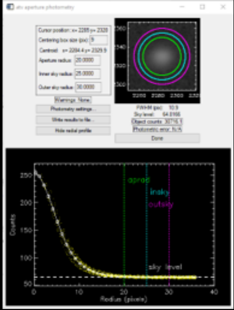
Caveats:

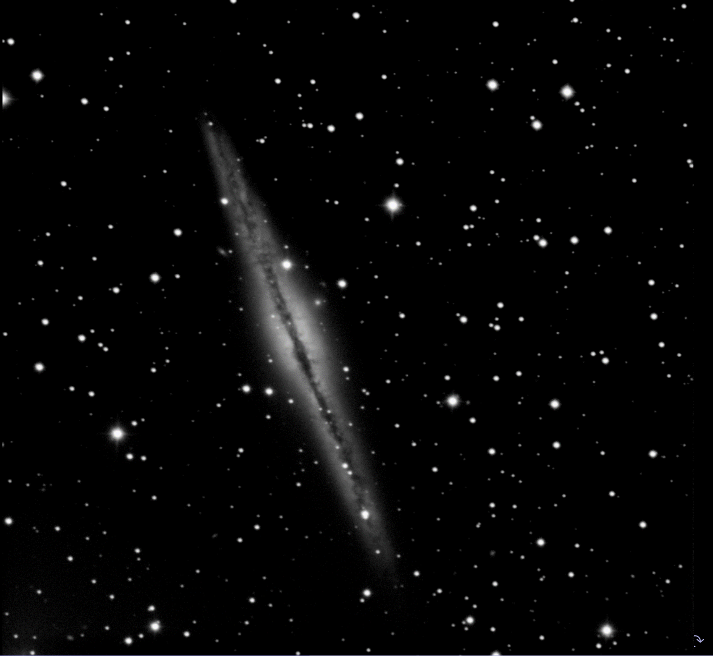
- None of these methods will improve the resolution of images by much more than a factor of two, and that only at very high S/N.
- The resolution will be seen to vary across the image, being better (sharper features, smaller stars) where S/N is higher — unlike the original.
- Maximum entropy deconvolution **does not conserve energy**: the resulting image will have a different total flux density than the original.
  - L-R and positivity deconvolution are booby-trapped against that.

So NEVER use them, particularly maximum entropy, on images with which you want to do photometry.

NGC 891

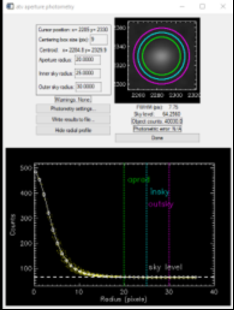
L, average of 24 5-minute frames

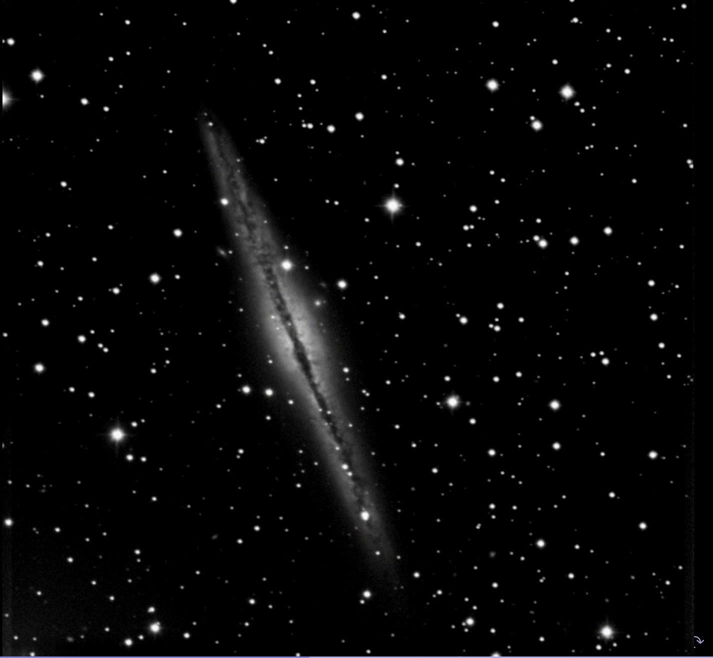




NGC 891, MEM deconvolve

Note the higher contrast in the dust lanes and the filaments perpendicular to the disk. The faint stars are also brighter.





Stretching images

CCDs have DN linearly proportional to the power collected; each pixel's signal is sent out as a 16-bit floating point number and stored in the computer as a 64-bit number. None of this matches displays very well.

- Computer monitors only display eight bits of brightness (0–256), for the very good reason that eyes can only resolve that many shades of gray. Printing on paper is similar.
- Thus, the huge range of astronomical brightness must be stretched (or compressed, really) to lie within an eight-bit range for display.
- Photographic emulsion has a useful compressive feature built into it, called reciprocity failure: Hypersensitized emulsion produces image density linear in power at low light levels, but the response becomes much more like logarithmic with brighter lights.

Stretching functions

Each of the image analysis programs that we use has a variety of ways to stretch images, notably

Gamma This maps power exponentially into signal:

$$DN_{\text{display}} = A(DN_{\text{image}})^{\gamma} + B$$

$\gamma = 1$ is, of course, the same as the original image, but $\gamma < 1$ compresses the display brightness into a smaller range. A ("brightness") and B ("background") can be set separately.

Logarithmic

$$DN_{\text{display}} = A \log(DN_{\text{image}}) + B$$

An attempt to mimic what the eye does, but it has obvious problems if it is possible for the pixels to have DN values of zero.

Stretching functions

Digital development process (DDP) Developed by a professional physicist who happened to be an amateur astronomer, this is an algorithm that is meant to mimic the useful features of reciprocity failure.
The algorithm is iterative and replicates the stages seen in an exposed photographic plate immersed in developer, so no equation.

arcsinh stretch At large values, $\text{arcsinh}(x)$ converges to $\ln(x)$; at small values, it is linear (like $\ln(x)$), but $\text{arcsinh}(0) = 0$ (unlike $\ln(x)$). So it satisfies all the constraints and embodies several useful properties.

- It also behaves very much like reciprocity failure in hypersensitized emulsion. The DDP stretch is very similar to the arcsinh stretch, as was first noticed long before the invention of DDP by fitting functions to the response of "analog" developed emulsion (Bunsen & Roscoe 1862, Schwarzschild 1899, Kron 1913).
- Not sure that I would put DDP into my image-processing software if I knew that…

Demonstration of stretching

Linear

DDP

Demonstration of stretching

arcsinh

DDP