

High Dynamic Range Image Display Using Asinh Stretch

W. D. Cotton (NRAO) September 9, 2026

Abstract—The Inverse hyperbolic sine (Asinh) stretch for image displays is explored and compared with linear and square root stretches. The asinh stretch is shown to represent structures over a wider range of brightness than linear and square root stretches on a high dynamic range image.

Index Terms—Image display, asinh

I. INTRODUCTION

The display of high dynamic range astronomical images generally requires a nonlinear stretch. The gold standard these days ¹ is the inverse hyperbolic sine stretch [1]. This memo describes the incorporation of this technique into the Obit package [2]² image display ObitView and compare several alternates on a high dynamic range image.

II. ASINH

The inverse hyperbolic sine (asinh) function is defined as

$$\operatorname{asinh}(x) = \ln(x + \sqrt{x^2 + 1}).$$

The asinh image stretch is defined by [1] as

$$I_{stretch} = \frac{\operatorname{asinh}(\frac{I-b}{a})}{\operatorname{asinh}(\frac{1-b}{a})}$$

where I is the pixel intensity, normalized to 0-1, a is the “stretch” factor and b is the “black point background cutoff”. The denominator normalizes the value to between 0 and 1. By convention a is the off-source σ of the pixel distribution and b is the mode of the distribution minus 1σ .

At low levels this stretch is linear with pixel values becoming logarithmic at high levels. This behavior is described in more detail by the google AI assistant in the Appendix.

III. OBITVIEW IMPLEMENTATION

The Obit image viewer, ObitView, uses a color map with 256 levels. The various stretches map pixel values onto these brightness levels. An asinh stretch has been added. The user interface is the Options panel which can be selected from the Options menu. This panel, displayed in Figure 1, allows selection of the stretch and an optional range of values to display; values below this range are given the minimum brightness and those above this range are saturated. The stretch type is selected by radio buttons.

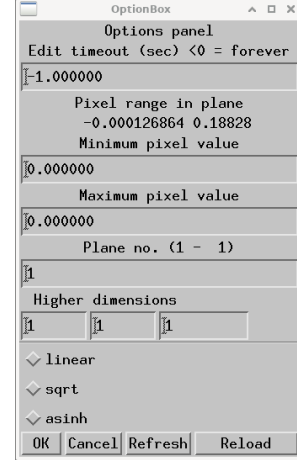


Fig. 1. Screen shot of the ObitView “Options” for selecting display stretch.

A. Examples

The default range of pixel values (zeroes in the appropriate text fields) depend on the stretch type. The default minimum is the mode of the pixel distribution minus one sigma of the off-source RMS. Figure 2 shows examples from the MeerKat image of the galaxy IC 4296 [3]. The dynamic range of this image is 18,000:1, peak to RMS.

B. Linear

The default maximum pixel brightness for this stretch is the pixel mode plus $13 \times \sigma$. The top panel in Figure 2 uses the default range for linear stretch.

C. Square root

The default maximum pixel brightness for this stretch is the pixel mode plus $64 \times \sigma$. The second panel in Figure 2 uses the default range for the square root stretch.

D. Asinh

The default maximum pixel brightness for this stretch is the maximum in the image. The third panel in Figure 2 uses this default range for the asinh stretch, here mode to 0.19 Jy. The bottom panel in Figure 2 uses a lower peak value to emphasize the intermediate levels.

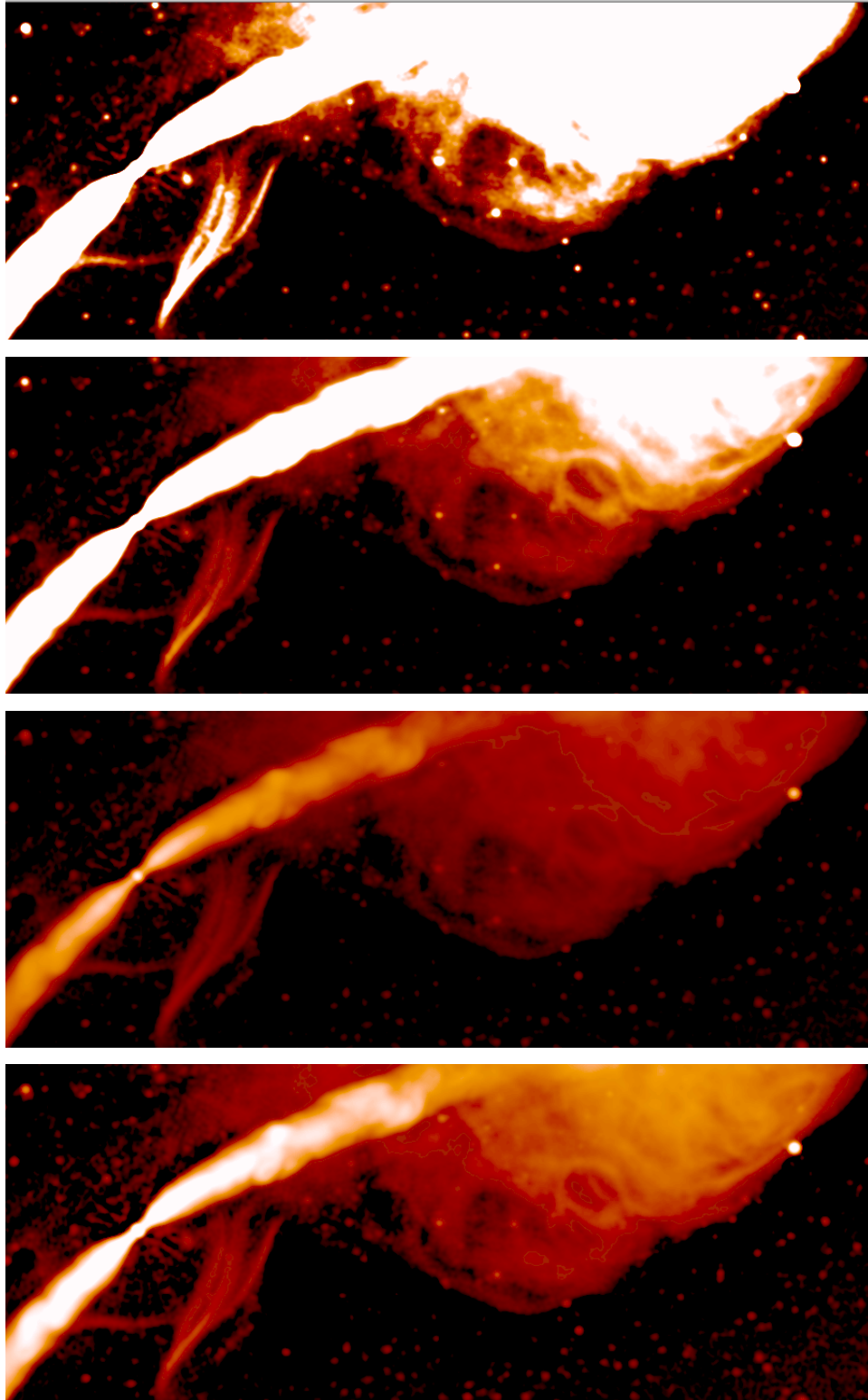


Fig. 2. Screen shot of a portion of the MeerKAT image of IC 4296 [3] from ObitView with Pseudo Flame coloration and default settings for
Top: linear stretch,
Second: Square root stretch,
Third: Asinh stretch, full pixel range (max 0.19 Jy)
Bottom: Asinh stretch, clipped above 0.01 Jy

IV. DISCUSSION

The comparison of linear, square root and asinh renditions of a high dynamic range (18,000:1) image in Figure 2 shows that asinh can show a wider range of brightness than the others. The entire range is visible although the intermediate level structures are more visible with a lower upper limit on the display range. Using the upper brightness limit of 0.01 Jy with the linear and square root stretches better show the intermediate brightnesses but essentially lose the lower brightness features.

APPENDIX

This appendix gives the responses from the Google AI assistant to questions about the asinh stretch.

The Gospel according to Google AI

A. What is inverse hyperbolic sine stretch for image display?

An inverse hyperbolic sine (asinh) stretch is an image processing function that brightens faint background details while keeping bright highlights and star colors from blowing out.

1) How It Works:

- **Behavior:** It acts like a linear scale for very bright objects, but changes to a logarithmic scale for very dark areas. [Note: google got this backwards.]
- **Control:** It uses a stretch factor and a black point to safely pull out hidden features, such as the faint outer halos of galaxies or dim nebulas.
- **Color Safety:** Unlike standard curve stretches, asinh preserves the natural RGB color ratios in color images so stars do not lose their true color or bloat.

2) Common Uses:

- **Astrophotography:** Popular in software like Siril Documentation to process deep-sky stacked images.
- **Wide Dynamic Range:** Useful whenever an image has extreme contrast differences between the darkest shadows and brightest light sources.

3) *The Core Equation:* The logarithmic definition of the inverse hyperbolic sine function is:

$$\text{asinh}(x) = \ln(x + \sqrt{x^2 + 1}).$$

When applied to an image pixel value, the standard stretching formula introduces two main parameters — a stretch factor (a) and a black point (b):

$$I_{\text{stretch}} = \frac{\text{asinh}(\frac{I-b}{a})}{\text{asinh}(\frac{1-b}{a})}$$

where

I is the original normalized pixel intensity (0 to 1),

b is the black point background cutoff and

a is the stretch or softening factor.

The denominator normalizes the output back to a 0-1 range so the brightest pixels (I=1) remain at maximum brightness.

B. Dual-Scale Behavior (Why it works)

The asinh(x) function transitions smoothly between two different mathematical behaviors depending on the value of x. See Figure 3.

1) Linear Behavior for Faint Pixels:

For small values of x (where $x \sim 0$), the square root term $\sqrt{x^2 + 1} \sim 1$. The Taylor series expansion shows that:

$$\text{asinh}(x) \sim x$$

Because it behaves linearly near the black point, it boosts faint structures evenly without creating harsh, noisy contrast transitions in your dark background.

2) Logarithmic Behavior for Bright Pixels:

For large values of x (where $x \gg 1$), the +1 inside the square root becomes negligible, meaning $\sqrt{x^2 + 1} \sim x$. The equation simplifies to:

$$\text{asinh}(x) \sim \ln(2x)$$

Because it behaves logarithmically for bright values, it compresses high dynamic range data. This prevents bright regions (like the core of a galaxy or a bright star) from saturating (clipping to pure white).

C. Preserving Chrominance (Color Safety)

In traditional RGB curve stretching (like a histogram transformation), if you apply a non-linear stretch to the Red, Green, and Blue channels independently, the ratios between the channels break. This causes bright stars to turn white and lose their color saturation. To solve this, asinh algorithms calculate a single scaling factor based on the luminance (L) or intensity of the pixel, and then multiply all three channels by that same factor:

- 1) Calculate total intensity: $L = (R+G+B)/3$ (or a weighted luminance).
- 2) Calculate the scaling factor: $f(L) = \text{asinh}(L)/L$
- 3) Apply to channels:
 $R_{\text{new}} = R \times f(L)$
 $G_{\text{new}} = G \times f(L)$
 $B_{\text{new}} = B \times f(L)$

Because the R,G,B ratios remain completely unchanged, the true colors and hue are perfectly preserved across the entire brightness gradient.

D. Show me how to calculate the ideal stretch factor (a) based on my image's noise floor.

To calculate the ideal stretch factor (a), you need to find the point where the image transitions from linear behavior (preserving faint details and noise without blowing them up) to logarithmic behavior (compressing the bright highlights). The rule of thumb in astrophotography is to set the stretch factor a equal to the standard deviation of the background noise (σ_{bg}) after subtracting the median background level.

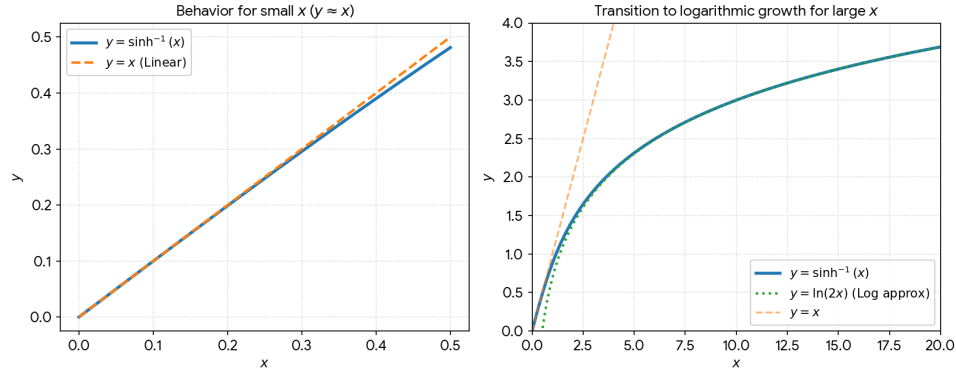


Fig. 3. Behavior of asinh stretch at low and high levels

1) *Step-by-Step Calculation Framework*: To calculate this programmatically or analytically, use the following three steps:

- 1) **Measure the Background Statistics** First, isolate a small, clean patch of sky that contains no stars, nebulae, or galaxies. Calculate two values from this background region:
 - Median_{bg}: The background pedestal level.
 - σ_{bg} : The standard deviation (the RMS noise floor).
- 2) **Set the Black Point (b)** The black point b should clip away the bulk of the empty background without clipping real data. Typically, it is set slightly below or exactly at the median background:

$$b = \text{Median}_{bg} - k \cdot \sigma_{bg}$$

(Where k is usually a small safety factor between 0 and 1, ensuring faint background structures aren't aggressively clipped to pure black).

- 3) **Determine the Stretch Factor (a)** To ensure that noise remains linear and doesn't get distorted into harsh artifacts, you set a relative to the remaining noise floor:

$$a = \sigma_{bg}$$

2) *Why This Math Protects Your Noise Floor*: Recall the Taylor expansion for the numerator: when the input value is very small, $\text{asinh}(x) \approx x$. By setting $a = \sigma_{bg}$, look at what happens to a pixel I that sits exactly one standard deviation above the black point ($I = b + \sigma_{bg}$):

$$\frac{I - b}{a} = \frac{(b + \sigma_{bg}) - b}{\sigma_{bg}} = 1$$

- Pixels where $I - b < \sigma_{bg}$: The input argument to asinh is less than 1. The function stays in its linear regime, meaning background noise is amplified uniformly without being warped or stretched into ugly structural patterns.
- Pixels where $I - b > \sigma_{bg}$: The argument becomes greater than 1. The function begins its smooth transition into the logarithmic regime, beginning to compress bright stars and galaxy cores so they don't saturate.

ACKNOWLEDGMENT

The google AI assistant was used extensively in researching this technique.

REFERENCES

- [1] R. Lupton, J. E. Gunn, and A. Szalay, "Preparing red-green-blue (rgb) images from ccd data," *The Astronomical Journal*, vol. 127, no. 1, pp. 351–354, jan 2004. [Online]. Available: <https://doi.org>
- [2] W. D. Cotton, "Obit: A Development Environment for Astronomical Algorithms," *PASP*, vol. 120, pp. 439–448, 2008.
- [3] J. J. Condon, W. D. Cotton, S. V. White, S. Legodi, S. Goedhart, K. McAlpine, S. M. Ratcliffe, and F. Camilo, "Threads, Ribbons, and Rings in the Radio Galaxy IC 4296," *ApJ*, vol. 917, no. 1, p. 18, Aug. 2021.