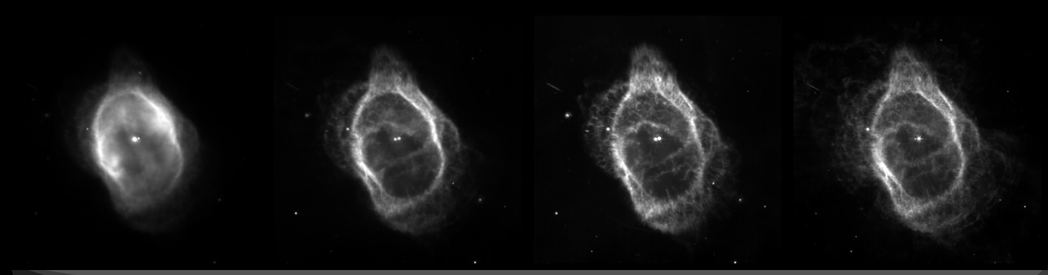
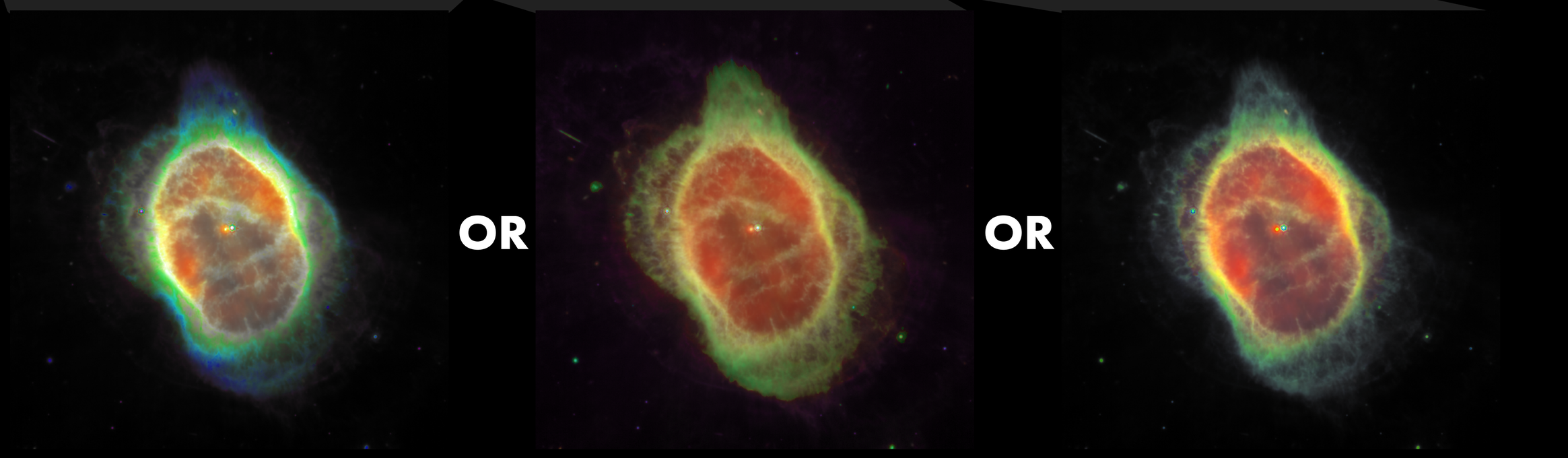
A New Method for Visualizing Multiband Astronomical Data in the Hue-Saturation-Value Colour Space

Hansen Jiang, Mubdi Rahman, Jennifer Scora, Tai Withers

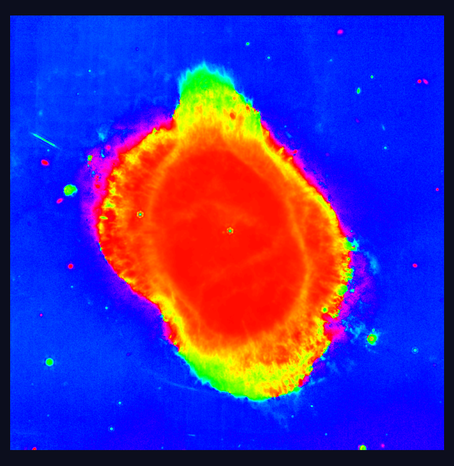


The visualization of astronomical data remains a key process in the modern astronomy workflow, providing the ability to generate diagnostics, identify features, and share results. By processing data in the HSV colour space, we expose the scientific and visual choices traditionally made implicitly in the RGB colour space. We present one such method of calculation in the HSV colour space for multiband images, for each of the three Hue-Saturation-Value channels. **We represent relative spectral luminosity via the Hue channel, the spread of luminosity via the Saturation channel, and the combined luminosity via the Value channel.** This method depicts noise, luminosity, and spectral differences so as to be able to better diagnose observation issues, identify and classify features, and depict multiband images when compared to traditional RGB methods.



For this method of calculation we use JWST-ERO observations of NGC3132, with four filters: f770w, f1130w, f1280w, f1800w. This method transforms calibrated and pixel-aligned multiband images into a single image using the HSV colour space. Each of the hue, saturation, and value channels are calculated separately, but all of the information from each filter is used in each channel.

Hue



The hue channel depicts the pure colour active at each pixel, represented by an angle. We calculate this colour from a weighted mean.

First, select the "base" colour for each filter in the range between red and blue, omitting non-spectral magenta and wrap-around reds.

$$\theta_k = f(\lambda_k) \quad \text{where } \theta_k \in [0, 240^\circ]$$

To portray relative wavelength distance, use the central wavelengths to linearly map each filter to the full colour range. To maximize contrast instead, use equal spacing over the range.

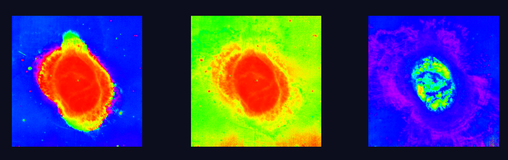
$$\theta_k = \frac{4\pi}{3} \cdot \frac{\lambda_k - \lambda_{\min}}{\lambda_{\max} - \lambda_{\min}}$$



$$\text{or} \quad \theta_k = (k-1) \cdot \frac{4\pi}{3(N-1)}$$

Next, calculate the weights. For each filter, clip the intensity to its q_1[th] and q_2[th] percentiles to remove the effects of outliers.

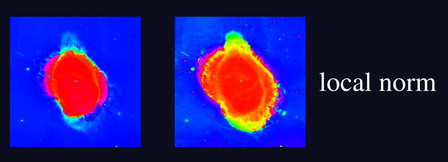
$$w_k(x,y) = \max\left(q_{1,k}, \min\left(q_{2,k}, I_k(x,y)\right)\right)$$



$(0, 100)$ $(0.001, 99.999)$ $(0.1, 99.9)$

To preserve the relative intensity between filters, use these values as the weights. To instead depict the structure of each filter in full dynamic range, min-max normalize these values to [0, 1].

$$W(x,y) = \{w_1(x,y), w_2(x,y), \ldots, w_N(x,y)\}$$



Then, using the weights w and angles θ for each filter, calculate the weighted circular mean with arctan2.

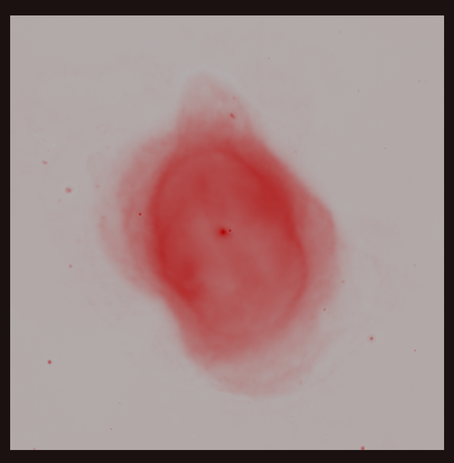
$$X(x,y) = \sum_{k=1}^{N} w_k(x,y) \cdot \cos(\theta_k)$$
$$Y(x,y) = \sum_{k=1}^{N} w_k(x,y) \cdot \sin(\theta_k)$$
$$\theta_{\text{mean}}(x,y) = \text{atan2}(Y(x,y), X(x,y)) \pmod{2\pi}$$

To compress the dynamic range, apply a stretch function, such as log, asinh, or sqrt.

Finally, min-max normalize to the 8-bit output (0 to 255).

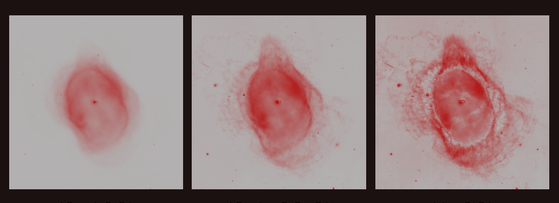
$$H(x,y) = 255 \cdot \frac{\theta_{\text{mean}}(x,y)}{2\pi}$$

Saturation



The saturation channel depicts the vividness of the colour active at each pixel, represented by a linear scale. We calculate this strength from a measure of spread.

First, calculate the intensities using the same algorithm from the hue channel weights: clip to the q_1[th] and q_2[th] percentiles of each filter, then min-max normalize per-filter (or keep as is to preserve relative intensity).



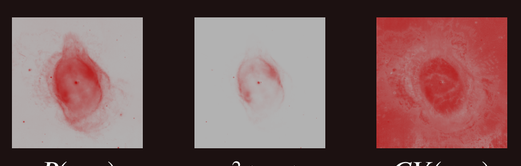
$(0, 100)$ $(0.1, 99.9)$ $(1, 99)$

Next, calculate the spread. To depict the full dynamic range, use the peak-to-peak distance. To depict dispersion from the average, use the variance. To show structure smoothness, use the coefficient of variation.

$$S_{\text{stretch}}(x,y) = P(x,y) = \max(W(x,y)) - \min(W(x,y))$$

$$\text{or} \quad S_{\text{stretch}}(x,y) = \sigma^2(x,y) = \frac{\sum_{k=1}^{N} (w_k(x,y) - \mu_w(x,y))^2}{N}$$

$$\text{or} \quad S_{\text{stretch}}(x,y) = CV(x,y) = \frac{\sigma(x,y)}{\mu_w(x,y)}$$



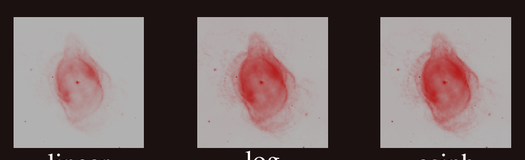
$P(x,y)$ $\sigma^2(x,y)$ $CV(x,y)$

To compress the dynamic range, apply a stretch function.

$$S_{\text{stretch}}(x,y) = S_{\text{spread}}(x,y)$$

$$\text{or} \quad S_{\text{stretch}}(x,y) = \log\left(\alpha \cdot S_{\text{spread}}(x,y) + 1\right)$$

$$\text{or} \quad S_{\text{stretch}}(x,y) = \text{asinh}\left(\frac{S_{\text{spread}}(x,y)}{\beta}\right)$$



linear log asinh

Finally, min-max normalize to the 8-bit output (0 to 255).

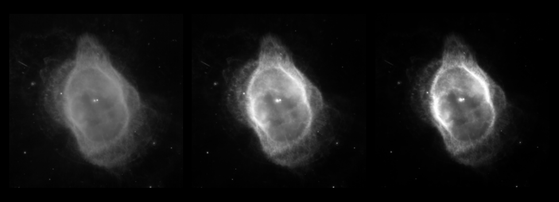
$$S(x,y) = 255 \cdot \frac{S_{\text{stretch}}(x,y) - \min(S_{\text{stretch}})}{\max(S_{\text{stretch}}) - \min(S_{\text{stretch}})}$$

Value

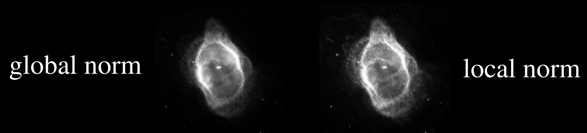


The value channel depicts the brightness of the colour active at each pixel, represented by a linear scale. We calculate this brightness by stacking the intensity information from each filter.

First, calculate the intensities using the same algorithm from the hue channel weights: clip to the q_1[th] and q_2[th] percentiles of each filter, then min-max normalize per-filter (or keep as is to preserve relative intensity).



$(0, 100)$ $(0.1, 99.9)$ $(1, 99)$



Next, using exposure times as weights, calculate the weighted mean of these intensities.

$$I_{\text{mean}}(x,y) = \frac{\sum_{k=1}^{N} w_k \cdot I_k(x,y)}{\sum_{k=1}^{N} w_k}$$

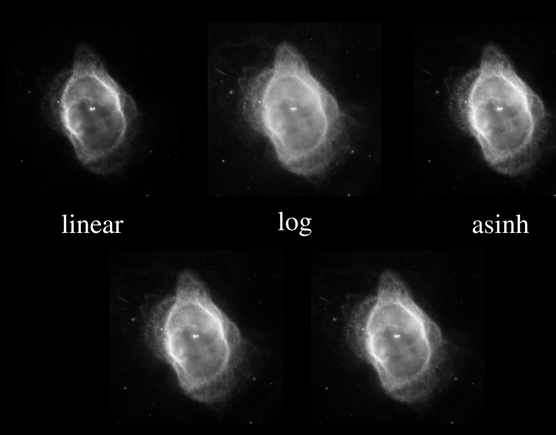
To compress the dynamic range, apply a stretch function.

$$I_{\text{stretch}}(x,y) = \log(\alpha \cdot I_{\text{mean}}(x,y) + 1)$$

$$\text{or} \quad I_{\text{stretch}}(x,y) = \text{asinh}\left(\frac{I_{\text{mean}}(x,y)}{\beta}\right)$$

$$\text{or} \quad I_{\text{stretch}}(x,y) = \sqrt{I_{\text{mean}}(x,y)}$$

$$\text{or} \quad I_{\text{stretch}}(x,y) = I_{\text{mean}}(x,y)^{\gamma}$$



linear log asinh sqrt power

Finally, min-max normalize to the 8-bit output (0 to 255).

$$V(x,y) = 255 \cdot \frac{I_{\text{stretch}}(x,y) - \min(I_{\text{stretch}})}{\max(I_{\text{stretch}}) - \min(I_{\text{stretch}})}$$

SIDRAT RESEARCH

Image Credit: Data from JWST-ERO Team / K. Pontoppidan (STScI) / MAST Archive (DOI: 10.17909/67ft-nb86).