# Acquiring, Processing and Displaying JWST NIRCam Images

Max Oakes

# Project Overview

- Automate the processing pipeline of JWST NIRCam image data
  - Automatically download the data given search parameters
  - Perform any needed processing to the images to make them 'cleaner'
  - Attempt to combine any data and display the resulting image to the same caliber as the ones released by NASA
  - All done via a single Python script
- Images released by NASA are processed by hand by multiple artists via photo editing software
  - Use FITS Liberator to open and level-adjust images
  - Photoshop to combine and enhance

WR 124

Credit: NASA via
https://www.nasa.gov/feature/goddard/2023/nasa-s-webb-telescope-captures-rarely-seen-prelude-to-supernova

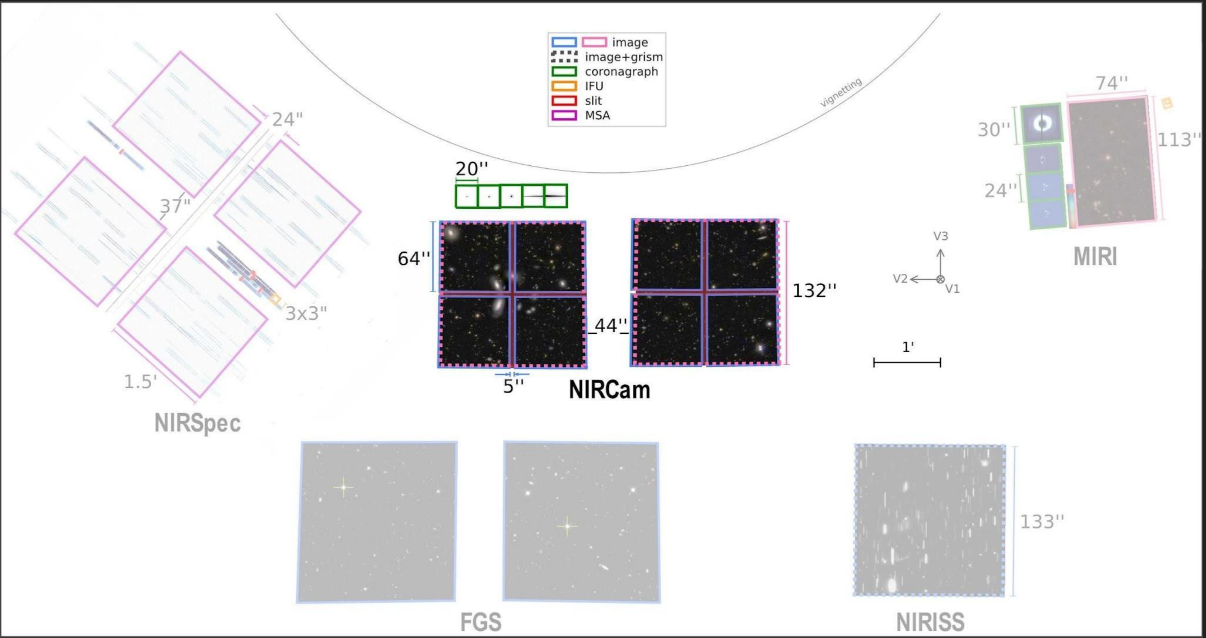
# Background

# James Webb Space Telescope

- Launched on December 25, 2021
- Successor to Hubble Space Telescope
- Mission to examine every phase of cosmic history
- Primary goals:
  - Use sensitive infrared imaging to see first stars and the faintest of galaxies
  - See through clouds of dust and gas to observe how planetary systems are born
  - Study atmospheres of extrasolar planets in hopes of finding building blocks of life

Webb NIRCam composite image of Jupiter from three filters – F360M (red), F212N (yellow-green), and F150W2 (cyan) – and alignment due to the planet's rotation. Credit: NASA, ESA, CSA, Jupiter ERS Team; image processing by Judy Schmidt. Via https://blogs.nasa.gov/webb/2022/08/22/webbs-jupiter-images-showcase-auroras-hazes/

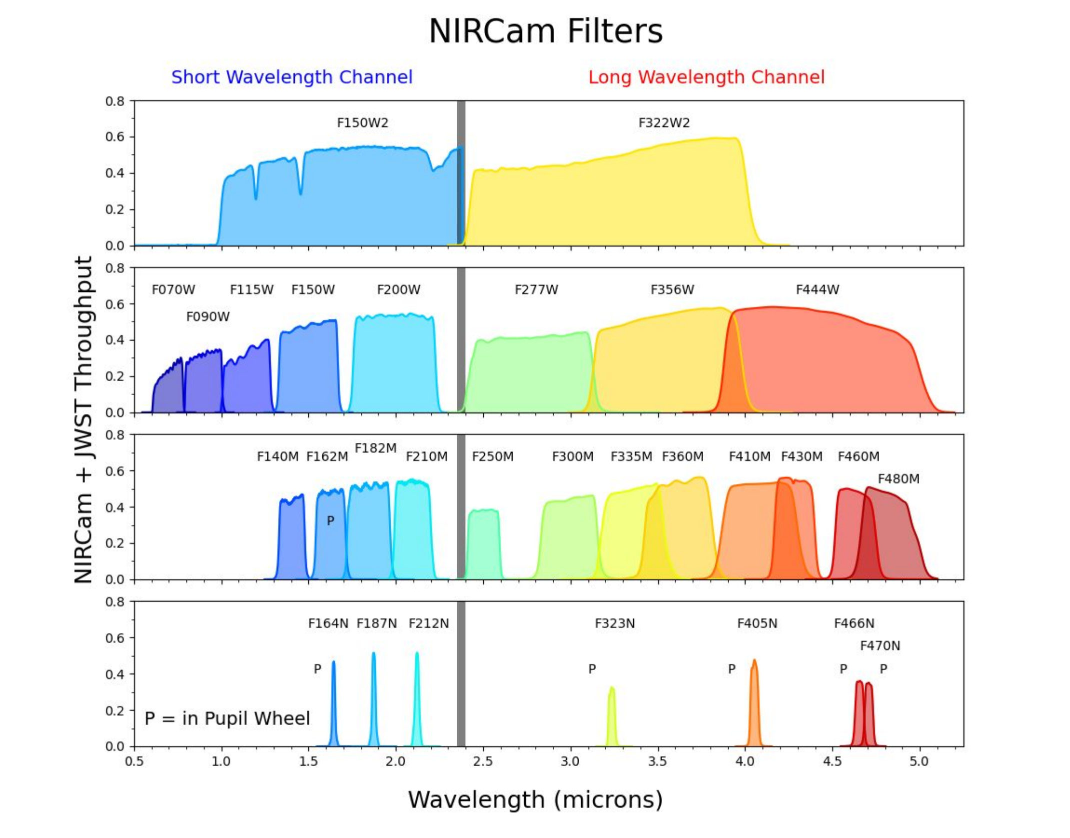
# James Webb Space Telescope Imaging Instruments




- MIRI Imager
  - Imaging Range: 5.6 - 25.5μm
  - 9 filters
  - FOV: 1.7 x 1.3 arcmin
- NIRCam Infrared Camera
  - Imaging Range: 0.6 - 5.0 μm
  - 29 available filters
  - FOV: 9.7 arcmin$^2$ = 0.000006253 steradian
  - Offers occlusion masks
- NIRISS Imaging
  - Range: 0.8 - 2.5μm
  - FOV: 2.2 arcmin$^2$
- NIRSPEC Spectroscopy
  - Range: 0.6 - 5.3μm
  - FOV: 2.4 x 3.6 arcmin

*JWST's Fine Guidance Sensor (FGS) provides data for science attitude determination, fine pointing, and attitude stabilization using guide stars in the JWST focal plane.*

# James Webb Space Telescope Filters




- Through all instruments on all modes:
  - 0.6 − 28.8μm
- For NIRCam only:
  - Small bandwidth: 0.6 - 2.3μm
  - Large bandwidth: 2.4 - 5.0μm

# FITS File Format

- Used to store astronomical data
- Each filter exposure image is in one FITS file
- Consists of HDUs (Header + Data units)
    - Each header unit can contain a table data structure that resembles a dictionary
        - KEYNAME = value / comment string
    - Also can contain a data unit
        - N-dimensional array of one data type (signed, unsigned ints, or floats)

# FITS File Format and Preview Image

Preview image that comes with each FITS file: represents the data unit

Example printout of the header records of a FITS file

```
REFFRAME= 'EME2000 '           / Ephemeris reference frame
EPH_TYPE= 'Definitive'         / Definitive or Predicted
EPH_TIME=    59733.75763888889 / [d] MJD time of position and velocity vectors
JWST_X  =   -45906090.46482842 / [km] barycentric JWST X coordinate at MJD_AVG
JWST_Y  =   -134176460.6176983 / [km] barycentric JWST Y coordinate at MJD_AVG
JWST_Z  =   -58398896.70939019 / [km] barycentric JWST Z coordinate at MJD_AVG
OBSGEO-X=     183610120.497275 / [m] geocentric JWST X coordinate at MJD_AVG
OBSGEO-Y=    -1402718728.79393 / [m] geocentric JWST Y coordinate at MJD_AVG
OBSGEO-Z=   -873646274.2810251 / [m] geocentric JWST Z coordinate at MJD_AVG
JWST_DX =    28.13761177082785 / [km/s] barycentric JWST X velocity at MJD_AVG
JWST_DY =   -8.142775878133994 / [km/s] barycentric JWST Y velocity at MJD_AVG
JWST_DZ =   -3.682592721118733 / [km/s] barycentric JWST Z velocity at MJD_AVG
OBSGEODX=     150.891951281282 / [m/s] geocentric JWST X velocity at MJD_AVG
OBSGEODY=     29.3392716433158 / [m/s] geocentric JWST Y velocity at MJD_AVG
OBSGEODZ=    -139.653416027791 / [m/s] geocentric JWST Z velocity at MJD_AVG
PA_APER =    111.6826959921155 / [deg] Position angle of aperture used
VA_SCALE=   0.9999352710703433 / Velocity aberration scale factor
BUNIT   = 'MJy/sr  '           / physical units of the array values

        Photometry information

PHOTMJSR=    3.043400049209595 / Flux density (MJy/steradian) producing 1 cps
PHOTUJA2=    71.53338505933402 / Flux density (uJy/arcsec2) producing 1 cps
PIXAR_SR=             2.31E-14 / Nominal pixel area in steradians
PIXAR_A2= 0.000981399999999999 / Nominal pixel area in arcsec^2

        Information about the coordinates in the file

RADESYS = 'ICRS    '           / Name of the coordinate reference frame

        Spacecraft pointing information

RA_V1   =    151.9074755207031 / [deg] RA of telescope V1 axis
DEC_V1  =   -40.51189863758767 / [deg] Dec of telescope V1 axis
PA_V3   =    111.9049856699223 / [deg] Position angle of telescope V3 axis

        WCS parameters

WCSAXES =                    2 / number of World Coordinate System axes
CRPIX1  =    2395.953857306228 / axis 1 coordinate of the reference pixel
CRPIX2  =    2394.276833537035 / axis 2 coordinate of the reference pixel
CRVAL1  =    151.7570120163398 / first axis value at the reference pixel
CRVAL2  =   -40.43665561653189 / second axis value at the reference pixel
CTYPE1  = 'RA---TAN'           / Axis 1 type
CTYPE2  = 'DEC--TAN'           / Axis 2 type
CUNIT1  = 'deg     '           / Axis 1 units
CUNIT2  = 'deg     '           / Axis 2 units
CDELT1  = 8.69997906407455E-06 / Axis 1 coordinate increment at reference point
CDELT2  = 8.69997906407455E-06 / Axis 2 coordinate increment at reference point
PC1_1   =   0.3694661313117234 / linear transformation matrix element
PC1_2   =   0.9292441970836021 / linear transformation matrix element
PC2_1   =   0.9292441970836021 / linear transformation matrix element
PC2_2   =  -0.3694661313117234 / linear transformation matrix element
S_REGION= 'POLYGON ICRS  151.721470971 -40.448319128 151.772509901 &'
```

# Methodology

# Downloading the Data

- Astroquery library allows for querying and downloading from the MAST archive
- Query Mode:
    - Allow the user to query for a target, mission title or proposal ID
    - Automatically runs a query to the MAST archive for stage 3 data for JWST NIRCam, then writes a CSV to the disk
- Run Mode:
    - Enter a proposal ID, then automatically download all associated FITS files for that mission and reference those files in the future
    - Images are put into a python dictionary with the filter as the key
        - (filter name, np array)

# Processing the Data: Value Clamping

- Clamp the brightness
  - All images are high dynamic range between ~0 and upwards of >10000
  - Find the 99.9% highest value, and use that as the upper clamp
  - Then normalize them to between 0 and 1 as floats
    - Need a reasonable range for when the image is converted to color

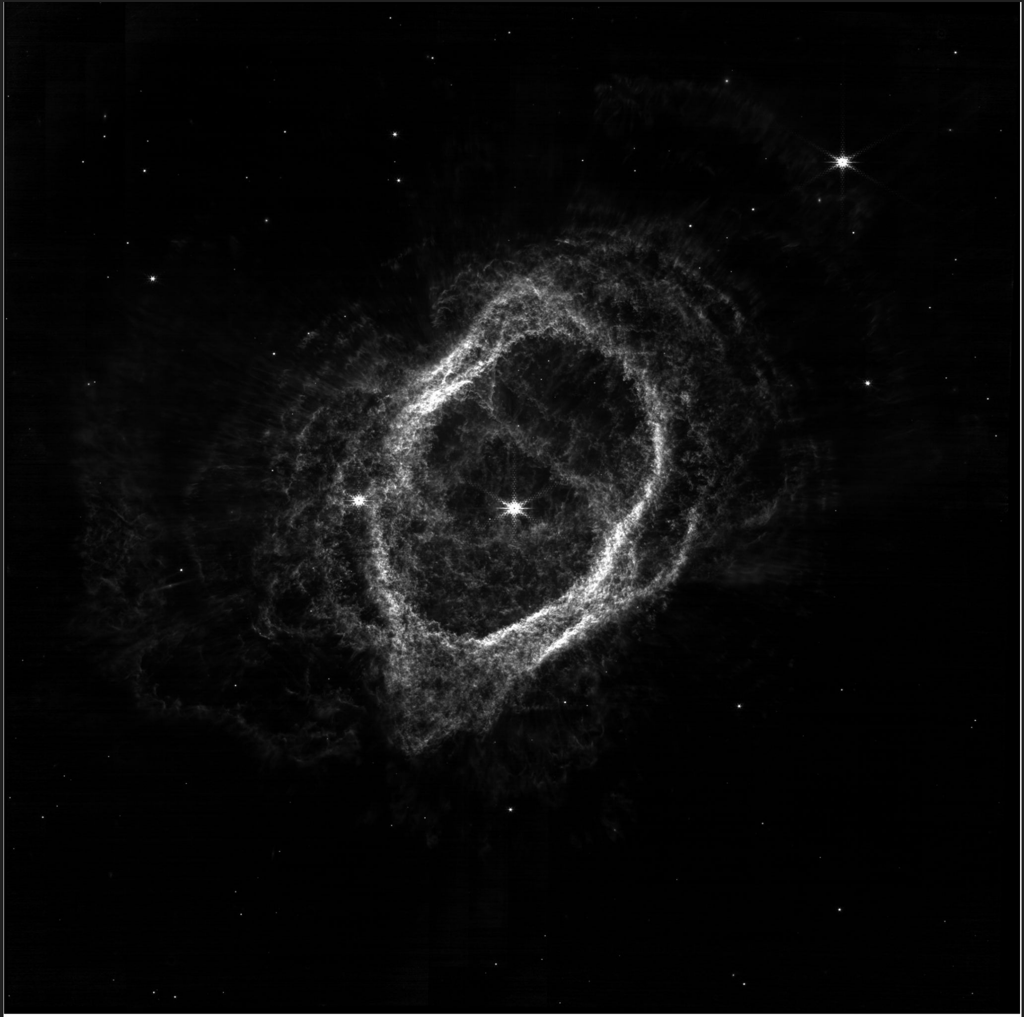
```
Smallest resolution image is [2348, 2356]: F356W/CLEAR
F090W/CLEAR: Min: -1.6, Max: 5316.87, Avg: 0.71, Median: 0.23, Quartile:15.0
F090W/CLEAR clipped to [0.0, 1.0]
F187N/CLEAR: Min: -7.63, Max: 19204.82, Avg: 2.81, Median: 0.32, Quartile:55.0
F187N/CLEAR clipped to [0.0, 1.0]
F212N/CLEAR: Min: -13.5, Max: 25527.65, Avg: 1.98, Median: 0.5, Quartile:43.0
F212N/CLEAR clipped to [0.0, 1.0]
F356W/CLEAR: Min: -0.01, Max: 396.43, Avg: 0.42, Median: 0.14, Quartile:8.0
F356W/CLEAR clipped to [0.0, 1.0]
F444W/F405N: Min: -1.63, Max: 1907.88, Avg: 1.76, Median: 0.17, Quartile:34.0
F444W/F405N clipped to [0.0, 1.0]
F444W/F470N: Min: -2.18, Max: 6497.36, Avg: 2.8, Median: 0.6, Quartile:55.0
F444W/F470N clipped to [0.0, 1.0]
```

# Processing the Data: Alignment

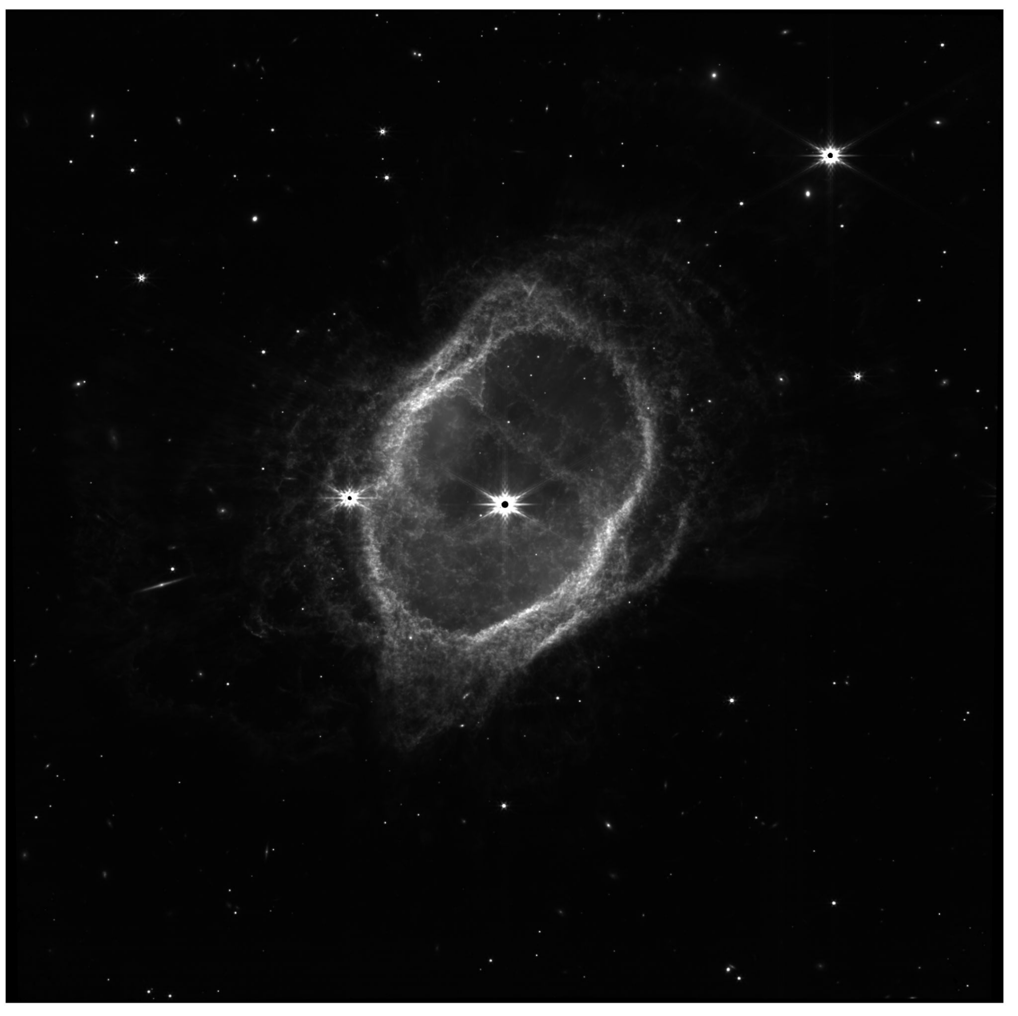
- Each filter image is a different resolution, need to make them the same to proceed
- Naive way:
  - Assume that the images all show the same exact spatial area
  - Find the smallest resolution of the images, then resize all images to that size
    - Pad the images if needed so there is no stretching
- Result:
  - The images are not aligned
  - If the captures are done around the same time, they have about the same rotation
    - If they are not the same rotation, the spacecraft orientation can be referenced to correct the rotation

Filter: F090W, clear

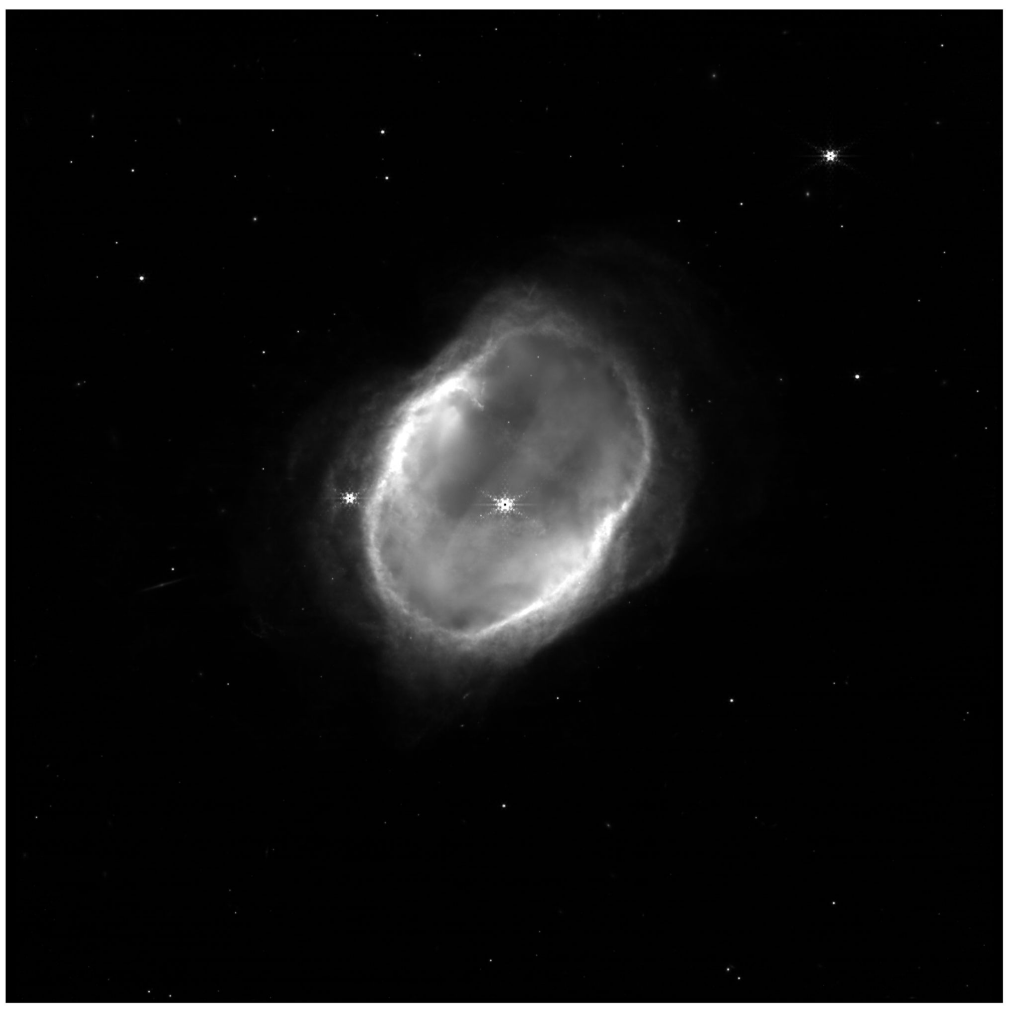
Filter: F187N, clear



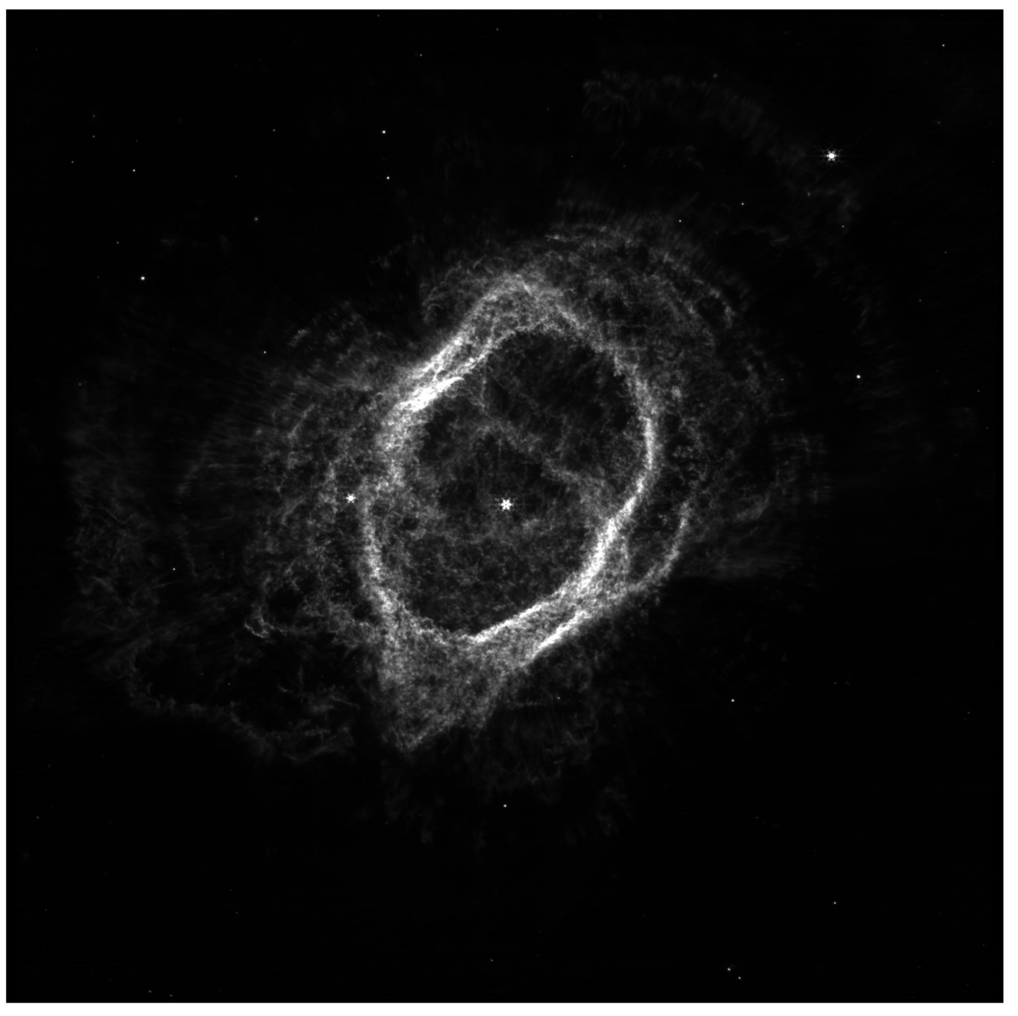
Filter: F212N, clear



Filter: F356W, clear



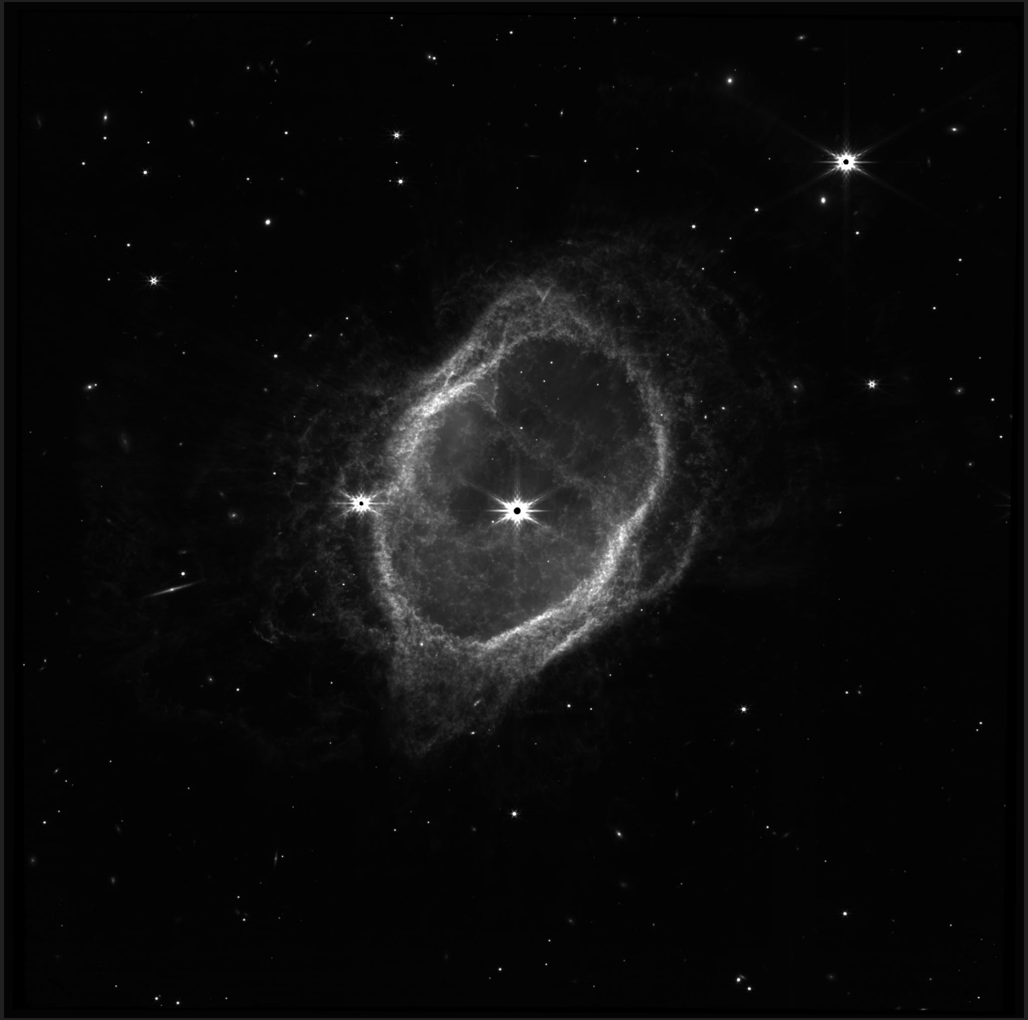
Filter: F444W, F405N pupil



Filter: F444W, F470N pupil

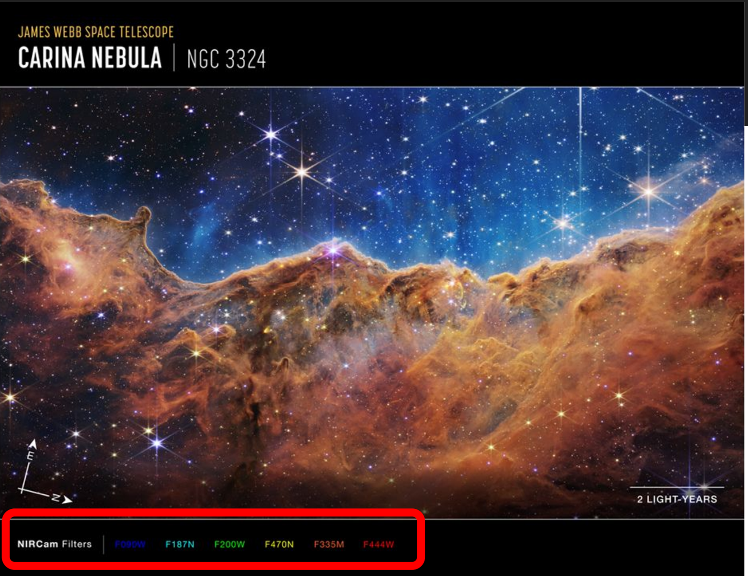
# Processing the Data: Alignment

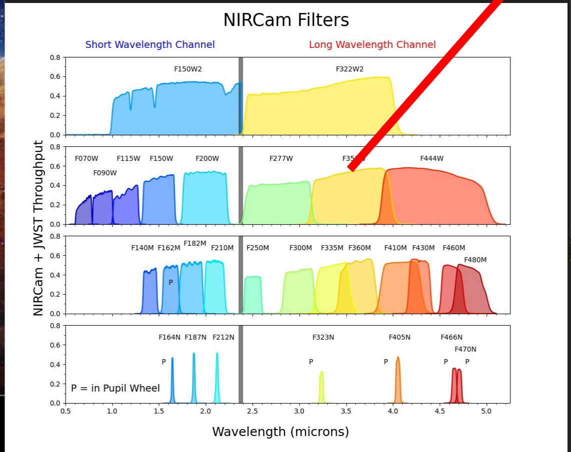
- Astroalign way:
  - Choose a target image
  - Automatically attempt to align each image to the target image via triangulation
- Result:
  - Perfectly aligned images
  - Alignment on some images may fail, only option is to scrap failing images



# Processing the Data: Assigning Colors

- Develop some set of false-color assignment
- Assign a color to each image
  - Use NIRCam filter chart as reference
  - JWST team appears to use that same color reference

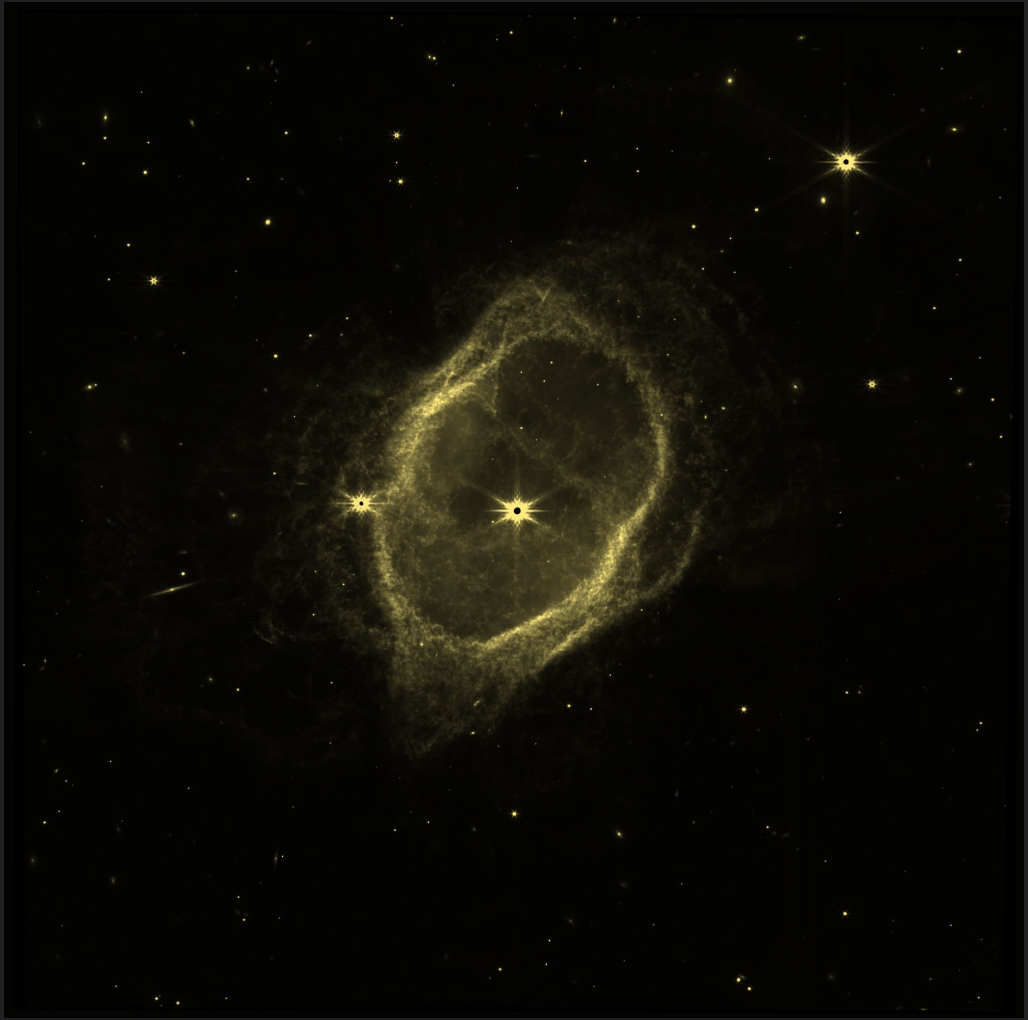


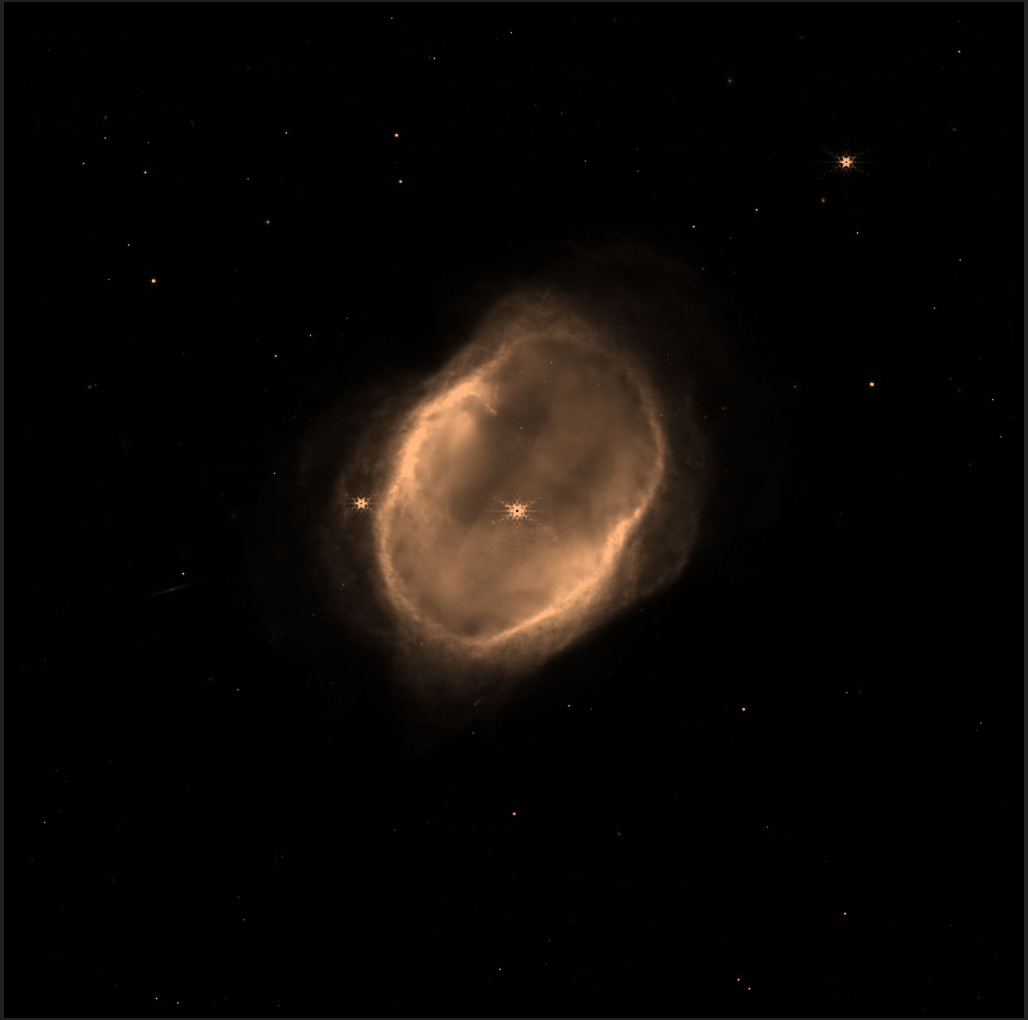

```python
nasa_colors_dict = {
    "F150W2/CLEAR": (127/255,205/255,255/255),
    "F070W/CLEAR": (127/255,127/255,216/255),
    "F090W/CLEAR": (127/255,127/255,240/255),
    "F115W/CLEAR": (127/255,135/255,255/255),
    "F150W/CLEAR": (127/255,175/255,255/255),
    "F200W/CLEAR": (127/255,231/255,255/255),
    "F140M/CLEAR": (127/255,165/255,255/255),
    "F150W2/F162M": (127/255,191/255,255),
    "F182M/CLEAR": (127/255,215/255,255/255),
    "F210M/CLEAR": (128/255,243/255,249/255),
    "F150W2/F164N": (127/255,193/255,255/255),
    "F187N/CLEAR": (127/255,219/255,255/255),
    "F212N/CLEAR": (131/255,247/255,246/255),
    "F322W2/CLEAR": (255/255,242/255,127/255),
    "F277W/CLEAR": (191/255,255/255,186/255),
    "F356W/CLEAR": (255/255,234/255,127/255),
    "F444W/CLEAR": (255/255,147/255,127/255),
    "F250M/CLEAR": (165/255,255/255,212/255),
    "F300M/CLEAR": (210/255,255/255,167/255),
    "F335M/CLEAR": (244/255,255/255,133/255),
    "F360M/CLEAR": (244/255,255/255,133/255),
    "F410M/CLEAR": (255/255,181/255,127/255),
    "F430M/CLEAR": (255/255,158/255,127/255),
    "F460M/CLEAR": (238/255,127/255,127/255),
    "F480M/CLEAR": (216/255,127/255,127/255),
    "F356W/F323N": (233/255,255/255,144/255),
    "F444W/F405N": (255/255,182/255,127/255),
    "F444W/F466N": (234/255,127/255,127/255),
    "F444W/F470N": (227/255,127/255,127/255)
}
```

```python
nasa_colors_dict = {
    "F150W2/CLEAR": (127/255,205/255,255/255),
    "F070W/CLEAR": (127/255,127/255,216/255),
    "F090W/CLEAR": (127/255,127/255,240/255),
    "F115W/CLEAR": (127/255,135/255,255/255),
    "F150W/CLEAR": (127/255,175/255,255/255),
    "F200W/CLEAR": (127/255,231/255,255/255),
    "F140M/CLEAR": (127/255,165/255,255/255),
    "F150W2/F162M": (127/255,191/255,255),
    "F182M/CLEAR": (127/255,215/255,255/255),
    "F210M/CLEAR": (128/255,243/255,249/255),
    "F150W2/F164N": (127/255,193/255,255/255),
    "F187N/CLEAR": (127/255,219/255,255/255),
    "F212N/CLEAR": (131/255,247/255,246/255),
    "F322W2/CLEAR": (255/255,242/255,127/255),
    "F277W/CLEAR": (191/255,255/255,186/255),
    "F356W/CLEAR": (255/255,234/255,127/255),
    "F444W/CLEAR": (255/255,147/255,127/255),
    "F250M/CLEAR": (165/255,255/255,212/255),
    "F300M/CLEAR": (210/255,255/255,167/255),
    "F335M/CLEAR": (244/255,255/255,133/255),
    "F360M/CLEAR": (244/255,255/255,133/255),
    "F410M/CLEAR": (255/255,181/255,127/255),
    "F430M/CLEAR": (255/255,158/255,127/255),
    "F460M/CLEAR": (238/255,127/255,127/255),
    "F480M/CLEAR": (216/255,127/255,127/255),
    "F356W/F323N": (233/255,255/255,144/255),
    "F444W/F405N": (255/255,182/255,127/255),
    "F444W/F466N": (234/255,127/255,127/255),
    "F444W/F470N": (227/255,127/255,127/255)
}
```

```
nasa_colors_dict = {
    "F150W2/CLEAR": (127/255,205/255,255/255),
    "F070W/CLEAR": (127/255,127/255,216/255),
    "F090W/CLEAR": (127/255,127/255,240/255),
    "F115W/CLEAR": (127/255,135/255,255/255),
    "F150W/CLEAR": (127/255,175/255,255/255),
    "F200W/CLEAR": (127/255,231/255,255/255),
    "F140M/CLEAR": (127/255,165/255,255/255),
    "F150W2/F162M": (127/255,191/255,255),
    "F182M/CLEAR": (127/255,215/255,255/255),
    "F210M/CLEAR": (128/255,243/255,249/255),
    "F150W2/F164N": (127/255,193/255,255/255),
    "F187N/CLEAR": (127/255,219/255,255/255),
    "F212N/CLEAR": (131/255,247/255,246/255),
    "F322W2/CLEAR": (255/255,242/255,127/255),
    "F277W/CLEAR": (191/255,255/255,186/255),
    "F356W/CLEAR": (255/255,234/255,127/255),
    "F444W/CLEAR": (255/255,147/255,127/255),
    "F250M/CLEAR": (165/255,255/255,212/255),
    "F300M/CLEAR": (210/255,255/255,167/255),
    "F335M/CLEAR": (244/255,255/255,133/255),
    "F360M/CLEAR": (244/255,255/255,133/255),
    "F410M/CLEAR": (255/255,181/255,127/255),
    "F430M/CLEAR": (255/255,158/255,127/255),
    "F460M/CLEAR": (238/255,127/255,127/255),
    "F480M/CLEAR": (216/255,127/255,127/255),
    "F356W/F323N": (233/255,255/255,144/255),
    "F444W/F405N": (255/255,182/255,127/255),
    "F444W/F466N": (234/255,127/255,127/255),
    "F444W/F470N": (227/255,127/255,127/255)
}
```



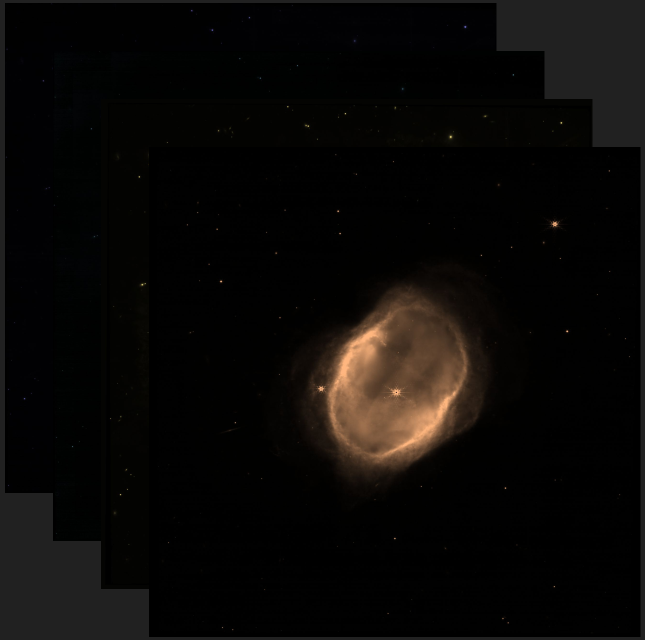
```python
nasa_colors_dict = {
    "F150W2/CLEAR": (127/255,205/255,255/255),
    "F070W/CLEAR": (127/255,127/255,216/255),
    "F090W/CLEAR": (127/255,127/255,240/255),
    "F115W/CLEAR": (127/255,135/255,255/255),
    "F150W/CLEAR": (127/255,175/255,255/255),
    "F200W/CLEAR": (127/255,231/255,255/255),
    "F140M/CLEAR": (127/255,165/255,255/255),
    "F150W2/F162M": (127/255,191/255,255),
    "F182M/CLEAR": (127/255,215/255,255/255),
    "F210M/CLEAR": (128/255,243/255,249/255),
    "F150W2/F164N": (127/255,193/255,255/255),
    "F187N/CLEAR": (127/255,219/255,255/255),
    "F212N/CLEAR": (131/255,247/255,246/255),
    "F322W2/CLEAR": (255/255,242/255,127/255),
    "F277W/CLEAR": (191/255,255/255,186/255),
    "F356W/CLEAR": (255/255,234/255,127/255),
    "F444W/CLEAR": (255/255,147/255,127/255),
    "F250M/CLEAR": (165/255,255/255,212/255),
    "F300M/CLEAR": (210/255,255/255,167/255),
    "F335M/CLEAR": (244/255,255/255,133/255),
    "F360M/CLEAR": (244/255,255/255,133/255),
    "F410M/CLEAR": (255/255,181/255,127/255),
    "F430M/CLEAR": (255/255,158/255,127/255),
    "F460M/CLEAR": (238/255,127/255,127/255),
    "F480M/CLEAR": (216/255,127/255,127/255),
    "F356W/F323N": (233/255,255/255,144/255),
    "F444W/F405N": (255/255,182/255,127/255),
    "F444W/F466N": (234/255,127/255,127/255),
    "F444W/F470N": (227/255,127/255,127/255)
}
```



```
nasa_colors_dict = {
    "F150W2/CLEAR": (127/255,205/255,255/255),
    "F070W/CLEAR": (127/255,127/255,216/255),
    "F090W/CLEAR": (127/255,127/255,240/255),
    "F115W/CLEAR": (127/255,135/255,255/255),
    "F150W/CLEAR": (127/255,175/255,255/255),
    "F200W/CLEAR": (127/255,231/255,255/255),
    "F140M/CLEAR": (127/255,165/255,255/255),
    "F150W2/F162M": (127/255,191/255,255),
    "F182M/CLEAR": (127/255,215/255,255/255),
    "F210M/CLEAR": (128/255,243/255,249/255),
    "F150W2/F164N": (127/255,193/255,255/255),
    "F187N/CLEAR": (127/255,219/255,255/255),
    "F212N/CLEAR": (131/255,247/255,246/255),
    "F322W2/CLEAR": (255/255,242/255,127/255),
    "F277W/CLEAR": (191/255,255/255,186/255),
    "F356W/CLEAR": (255/255,234/255,127/255),
    "F444W/CLEAR": (255/255,147/255,127/255),
    "F250M/CLEAR": (165/255,255/255,212/255),
    "F300M/CLEAR": (210/255,255/255,167/255),
    "F335M/CLEAR": (244/255,255/255,133/255),
    "F360M/CLEAR": (244/255,255/255,133/255),
    "F410M/CLEAR": (255/255,181/255,127/255),
    "F430M/CLEAR": (255/255,158/255,127/255),
    "F460M/CLEAR": (238/255,127/255,127/255),
    "F480M/CLEAR": (216/255,127/255,127/255),
    "F356W/F323N": (233/255,255/255,144/255),
    "F444W/F405N": (255/255,182/255,127/255),
    "F444W/F466N": (234/255,127/255,127/255),
    "F444W/F470N": (227/255,127/255,127/255)
}
```

# Processing the Data: Combining Images

- Merge each layer with a lighten operation
  - For each RGB color channel and for each assigned color filter
  - *result* = max(*nextImage* * factor, *thisImage*)
- Output as uint8 PNG



# Processing the Data: Displaying the Images

Output of my script with NASA's color assignment (using Astroalign alignment)

Output of my script with NASA's color assignment
(using Astroalign alignment)
*PLUS* quick color curve enhancement in photo editor

What NASA released

# Conclusions

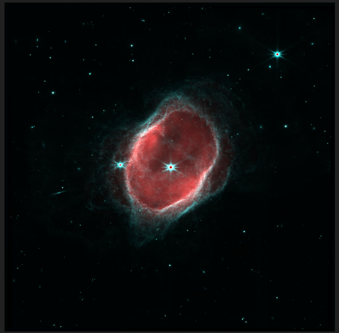
- Using a script to successfully download and (eventually) align the images would be helpful
  - My solution makes it very easy to download mission data to organized directories
  - Once this script can automatically align all images, it will be helpful
- Programatically coloring the final image is limited
  - Ultimately up to the editor to determine what looks aesthetically pleasing
  - Publicly released images go through a team of artists to combine and enhance the images
  - Color choice might depend on scene

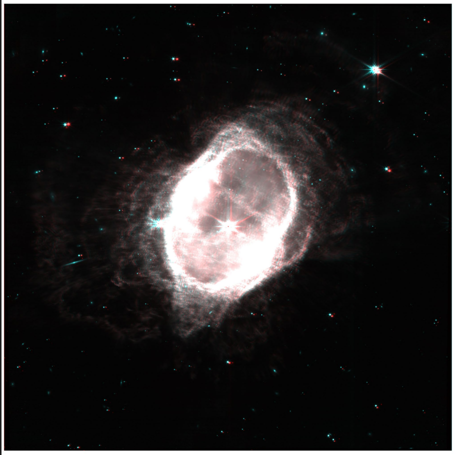
# Future Work

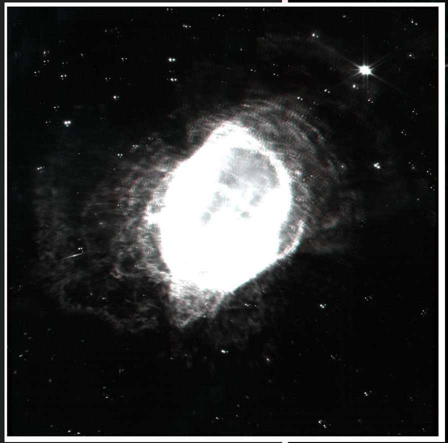
- Align all images using the given photo metadata
  - In the current version, no header data is used other than the filter
- Add further color adjustments to script
  - Currently only perform lighten operation on each filter, then combine them
  - Possibility of adding color curve adjustment to add more vivid color
- Fill in occluded areas
- Test more target scenes
  - Only tested NGC 3132
  - This script should in theory work for other images, since no scene-specific parameters were used

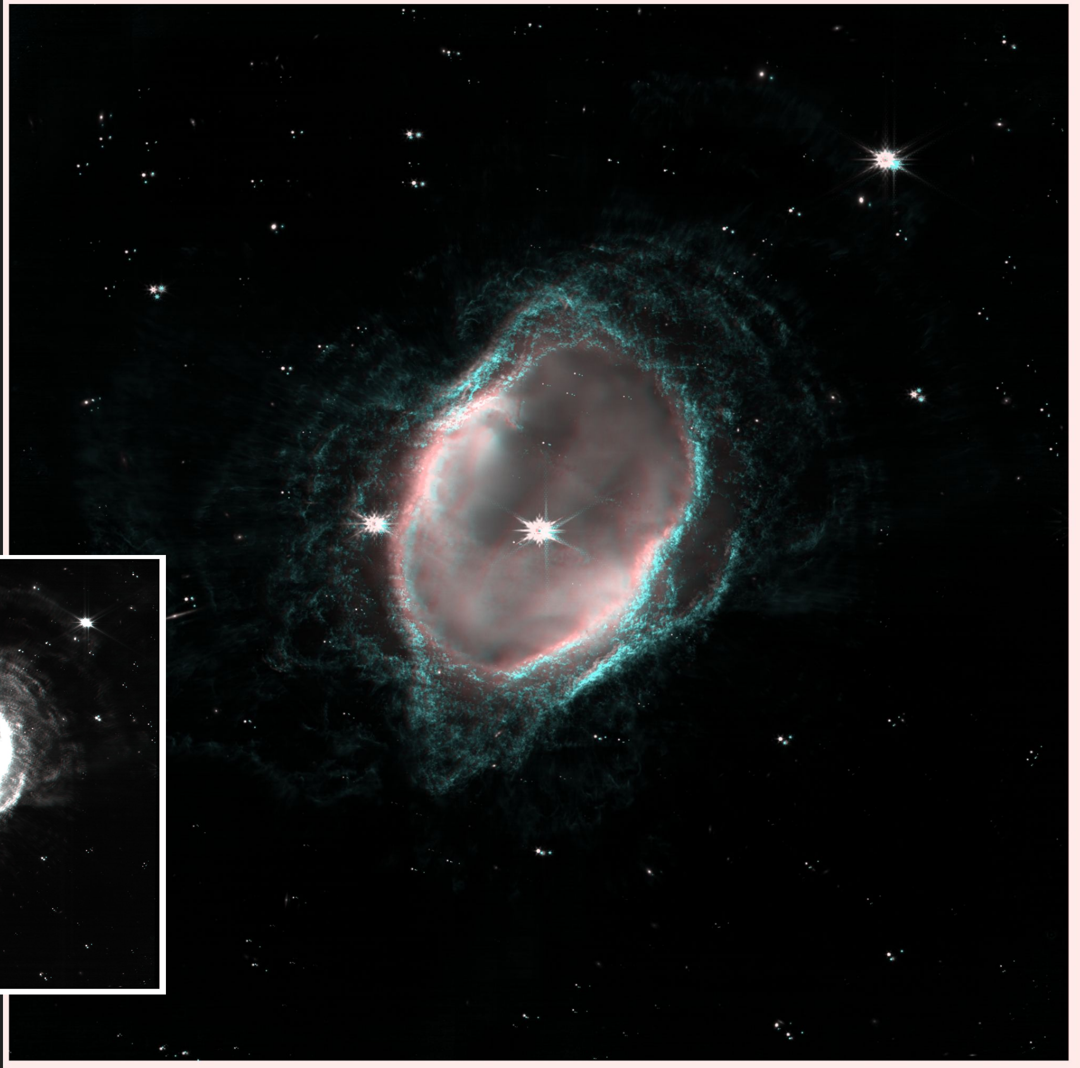
# Questions?

# Final results of bad alignments and bad blending modes









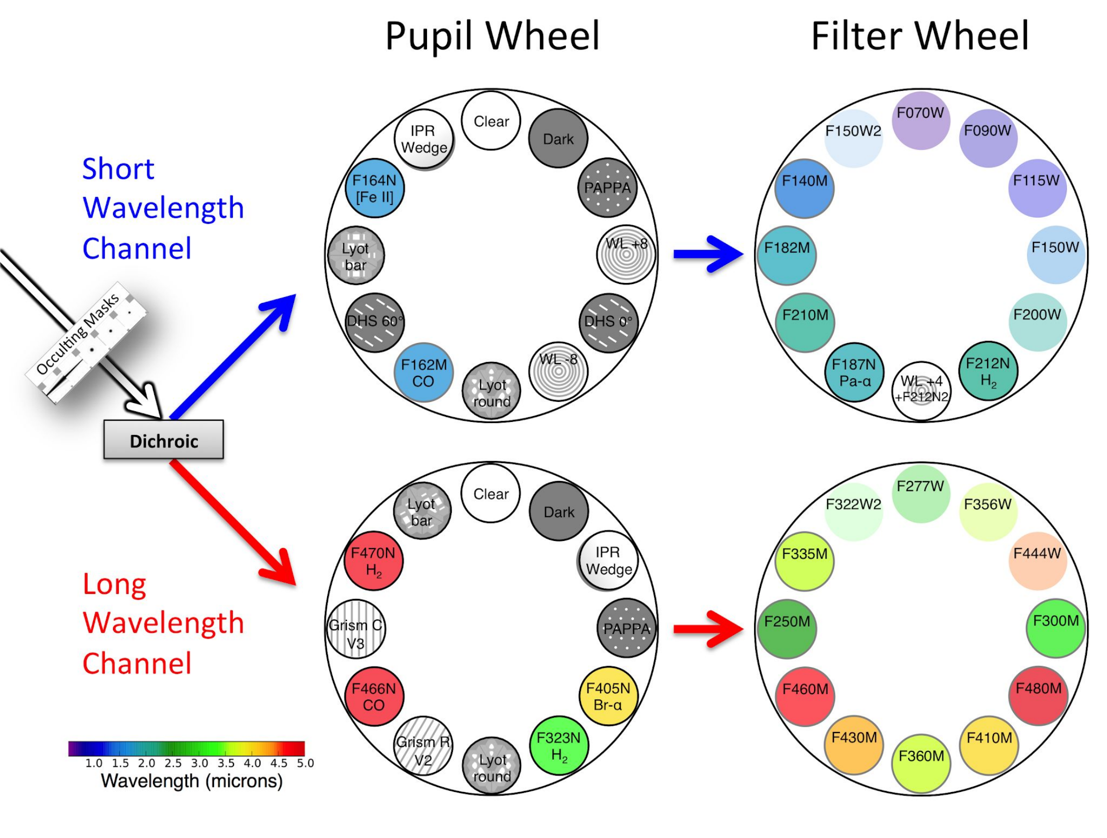
# Image Data

- All data can be found on the MAST Archive, alongside data from any other sources
- JWST data has 4 processing stages:
  - Raw FITS Files (Stage 0): Raw spacecraft output packaged to FITS file format
  - Detector Corrected Exposures (Stage 1): Output count-rate images after applied detector-level corrections to raw data
  - Flux Calibrated Exposures (Stage 2): Fully-calibrated individual exposures, and associated files, to all imaging/spectroscopy data and modes
  - Calibrated Combined Products + Associations (Stage 3): Combined final data products specific to each observing mode

# FITS File Format

- Used to store astronomical data
- One FITS file per filter image
- Consists of HDUs (Header + Data units)
  - Each header unit can contain a table data structure that resembles a dictionary
    - KEYNAME = value / comment string
  - Also can contain a data unit
    - N-dimensional array of one data type (signed, unsigned ints, or floats)
- JWST stage 3 data contains several HDUs
  - Primary: contains no data unit, but has a header unit of records with information about the data capture and processing already completed
  - SCI: 2-dimensional array that I use as the image for a wavelength capture. Records detailing nearly every detail about that capture
  - ERR, VAR, others that show error for each pixel value for the SCI table

# James Webb Space Telescope Pupil and Filter Wheels




- Two wheels that can be rotated to allow different combinations of filters

# Good Resources

- How the composed images are made, and what the raw images look like
  - https://webbtelescope.org/contents/articles/how-are-webbs-full-color-images-made
- NIRCam filters:
  - https://jwst-docs.stsci.edu/jwst-near-infrared-camera/nircam-instrumentation/nircam-filters
- 