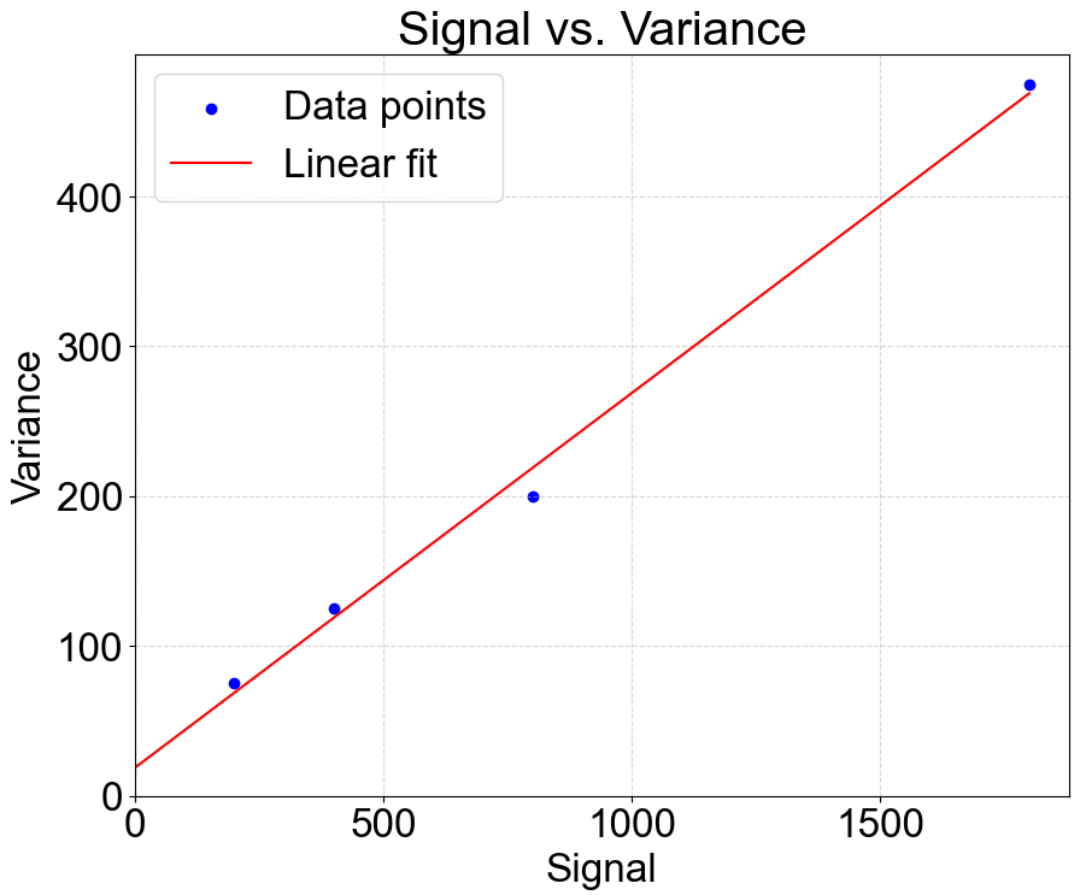
2. A sequence of flat-field observations is obtained at different exposure levels. The following signal and variance values were found: 200DN, 75; 400DN, 125; 800DN, 200; 1,800DN, 475.
Plot the data and derive the gain factor g (electrons/DN) and the readout noise R.



$$V_M = \frac{1}{g} S_M + \left(\frac{R}{g}\right)^2$$

g = 1/slope, R = g * sqrt(intercept)

slope = 0.25, intercept = 18.75

→ g = 4, R = 4 * sqrt(18.75) ~ 17.3

8. Describe the major steps needed to calibrate a high-resolution spectrograph with a CCD or IR array detector.

Identify Dispersion Direction: Determine if increasing pixel numbers correspond to increasing wavelength, setting up a clear understanding of how data is organized.

Interpolate Over Dead Pixels: Fill in data gaps caused by dead pixels or columns to avoid skewing analysis results. Keep a record of these locations since they do not contain actual data.

Flat-Field Calibration: Combine flat-field images, ensuring they are normalized. These images may be captured with the slit wide open, creating uniform CCD illumination using, for example, a quartz lamp against a white screen.

Apply Flat-Field Correction: Divide the observed stellar spectra by the flat-field image to correct for pixel-to-pixel sensitivity variations.

8. Describe the major steps needed to calibrate a high-resolution spectrograph with a CCD or IR array detector.

Order Identification: If using software like IRAF, manually identify spectral orders on the CCD, potentially using a brighter star if the target star is too faint. This allows the software to correctly locate and extract spectra.

Spectrum Extraction: Isolate rectangular sections of CCD data containing the stellar spectrum and arc lamp spectra, using routines like "apsum" in IRAF.

White-Light Spectrum Division: Divide the flat-fielded stellar spectra by the white-light spectrum acquired with the normal slit width. Ensure the white-light spectrum is itself flat-fielded to remove fringe effects, focusing on overall variations rather than pixel-specific corrections.

Wavelength Calibration: Identify emission lines from an arc lamp, typically a thorium-argon lamp, to establish an accurate pixel-to-wavelength transformation. This step is critical for precise wavelength measurements.

Introduction to IDL for Image Processing

- **IDL (Interactive Data Language)** is a comprehensive **visualization package** for scientific, engineering, and medical data.
- **Array-oriented language** enables complex functions, procedures, and applications without the need for traditional programming (e.g., FORTRAN or C).

→ Data simulation and modeling, **Commercial i.e. not for free**

Introduction to IDL for Image Processing

- Executes IDL commands as soon as they are entered.
- Operators and functions work on entire data arrays.
- Image display & graphics are integrated with computation for effective visualization of results.
- IDL Widgets enables creation of Graphical User Interfaces (GUIs), making it especially useful for instrument builders.
- Supported on UNIX, VMS, Microsoft Windows, and Macintosh systems.

Handling FITS Files in IDL: procedures

- IDL loads FITS into a string that contains FITS header and a matrix of real numbers which are the pixel values.
- The Space Telescope Science Institute (STScI) provides IDL scripts
- To read a FITS file (myfile.fits) and display data:

IDL>image = readfits("myfile.fits", header)

IDL>tvscl, image

- IDL commands are either procedures or functions
- "tvscl" is a procedure that will plot a matric with auto-scaling.
- Procedure commands have two kinds of parameters: arguments and keywords
- To write data in a matrix "image" to the FITS "write.fits" with header info in "hdr_str" (optional argument) with undefined pix set to zero (NaN is a keyword for undefined pixel)

IDL>writefits, "outfile.fits", image, hdr_str, NaNvalue = 0

Procedure	Argument	Keyword

Handling FITS Files in IDL: functions

- readfits is a kind of function that creates new variables.

IDL> new_variables= function(parameters)

- Users can create their own procedures and functions using IDL programming lang.
- Full programming lang like FORTRAN or C++, but with higher level interface.
- Make guessing right syntax easy: to print to the screen is "print"
- Programs are files with ".pro" extensions loaded by

IDL> .run myprogram

- Comments begin with ";"
- Large blocks of programs, such as the body of a loop, are started with "begin" and finished with "end"

Example) IDL Program "subflat.pro"

- This program subtract the bias from a series of images that have a common root to their name followed by a number.
- It will then optionally divide by a flat-field image.
- Output will be written to fits files with a different root name

This file loaded at the command line

IDL>.run subflat

; Define the procedure

;"$" allows the definition to continue on the next line

; Capitalization and indentation are stylistic, but not required

PRO SUBFLAT, root, out, first, last, $

BIASNAME=biasname, FLATNAME=flatname, DIVFLAT=divflat

Example) IDL Program "subflat.pro"

- This program subtract the bias from a series of images that have a common root to their name followed by a number.
- It will then optionally divide by a flat-field image.
- Output will be written to fits files with a different root name

```
; Check to see if bias and flat are defined.
; If not, they will have the default names ; of "bias" and "flat"

IF NOT KEYWORD_SET(BIASNAME) THEN biasname = "bias"
IF NOT KEYWORD_SET(FLATNAME) THEN flatname = "flat"

;Define loop variable to start with first
I = first

;read in the bias data
biasdata = READFITS(biasname + ".fts")
```

Example) IDL Program "subflat.pro"

```
;next loop through the images.
;Note that the body of the loop begins with "begin" and ; ends with "end"
WHILE (I le last) DO BEGIN
  infits =root + STRTRIM(i,1)+".fts" ; make name for of input fits file
  data = READFITS(infits, hdr) ;read that file into data and hdr
  outdata= data biasdata ;subtract off the bias from data
IF KEYWORD_SET(DIVFLAT) THEN BEGIN
  outdata= outdata / flatdata ;divide by  flat if requested
END
outfits= out+STRTRIM(i,1)+".fts" ;construct output filename
WRITEFITS, outfits, outdata, hdr ;write data, w/ same header
 I= I+1 ;increment the loop variable
END
 ;end the procedure
END
```

Example) IDL Program "subflat.pro"

- This program could then be loaded at the command line with

IDL>.run sub ¯at (the .pro is assumed).

- Then it could be executed on im1.fts, im2.fts, im3.fts, with bias.fts and flat.fts, to make out1.fts, out2.fts, and out3.fts, with

IDL>subflat, "im", "out", 1, 3, bi= "bias", fl = "flat", /div

The IDL Astronomy User's Library, at the Goddard Space Flight Center, contains over 500 astronomically oriented routines (see http://idlastro.gsfc.nasa.gov/homepage.html for more information).

There is also a handy IDL-based interactive display tool called ATV (ATV.pro) written and maintained by Aaron Barth at University of California, Irvine (see http://www.physics.uci.edu/~barth/atv/)

FITS Liberator: Enhancing Image Quality

- FITS Liberator: Free software for importing and processing FITS images in Adobe Photoshop.
- Preserves grayscale values for high-quality editing.
- Create multi-color images by layering different wavelengths or spectral bands data.
- Adjust levels to enhance features invisible to the naked eye.
- Download FITS Liberator and practice importing a FITS file.
- Use Photoshop to adjust contrast and layer images for a composite view.

10.3 PRINCIPLES OF IMAGE ANALYSIS AND PROCESSING
10.3.1 Displaying images

- Displaying images = First step!
- Images are displayed with true intensity values in the screen range (e.g., 0-255).
- Look-Up Table (LUT): weakest = 0, brightest = 255, intermediate signals are binned into the intermediate levels
- Software with a cursor has advantages to read pixel coordinates and actual intensity, not just scaled values.
- If dinamic range is large (i.e. Imax >> Imin), linear mapping is not good contrast
- Set white level (255) to 10% of peak

→faint feature are visible

- Any "window" of signal levels can be stretched from 0 to 255 display levels

10.3 PRINCIPLES OF IMAGE ANALYSIS AND PROCESSING
10.3.1 Displaying images

Stretch Techniques:

(c) Different transfer functions for separate intensity ranges (e.g., steeper slopes for faint signals).

(d) Logarithmic:

Enhances faint details by steeply rising at low intensities, then compressing bright details.

(f) Sawtooth or wraparound:

Repetitively maps gray levels to create a wraparound effect.

(other) Histogram Equalization:

enhance visibility of faint objects near the background level.

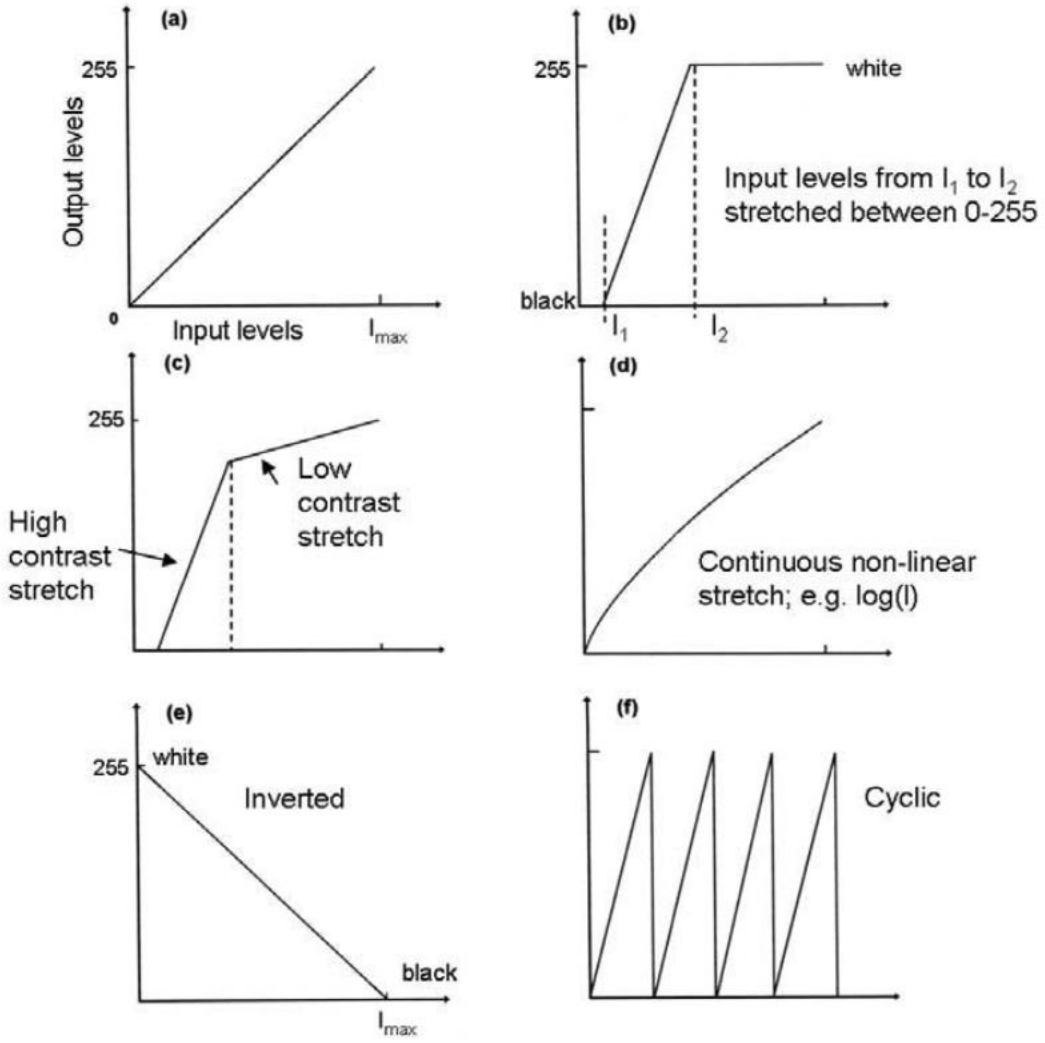


Figure 10.3. Examples of six look-up tables (LUTs) or display-stretching transformations: (a) linear; (b) linear between two intensities; (c) two-step linear; (d) logarithmic; (e) inverse; and (f) sawtooth or wraparound.

10.3 PRINCIPLES OF IMAGE ANALYSIS AND PROCESSING
10.3.1 Displaying images

(other) Histogram Equalization:

- equalizes the histogram of signal values vs the number of pixels with that signal
- enhance visibility of faint objects near the background level.

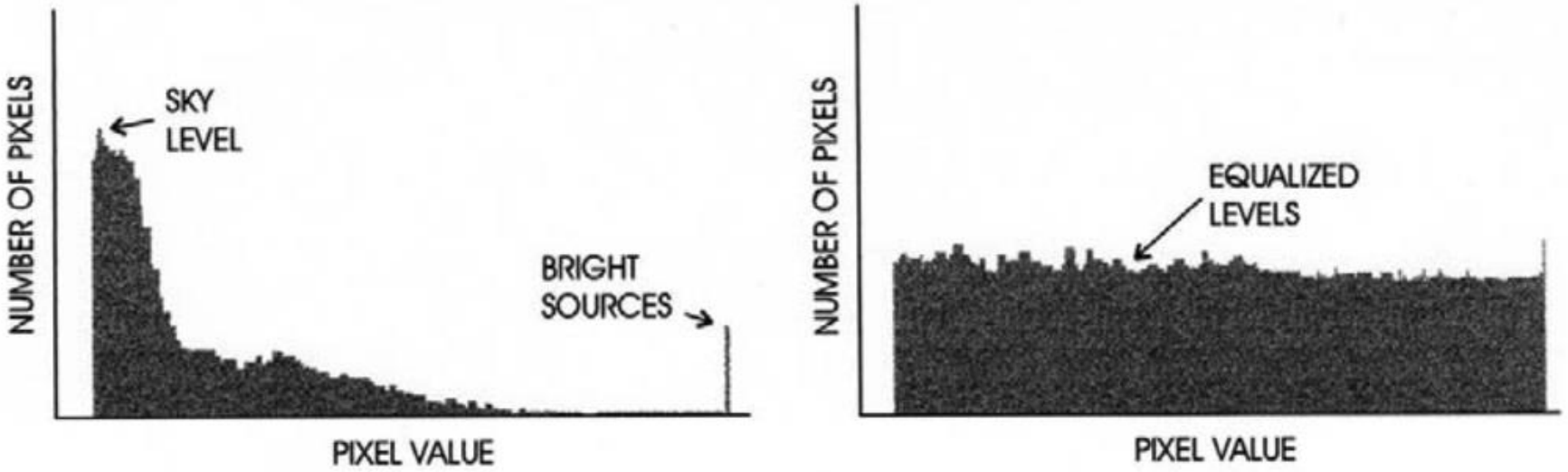


Figure 10.4. (Left): histogram of the distribution of signal values in the image. In a sparse field most of the pixels record the sky value. (Right): a display transformation that results in equalization of the histogram bringing up faint objects.

10.3 PRINCIPLES OF IMAGE ANALYSIS AND PROCESSING
10.3.1 Displaying images

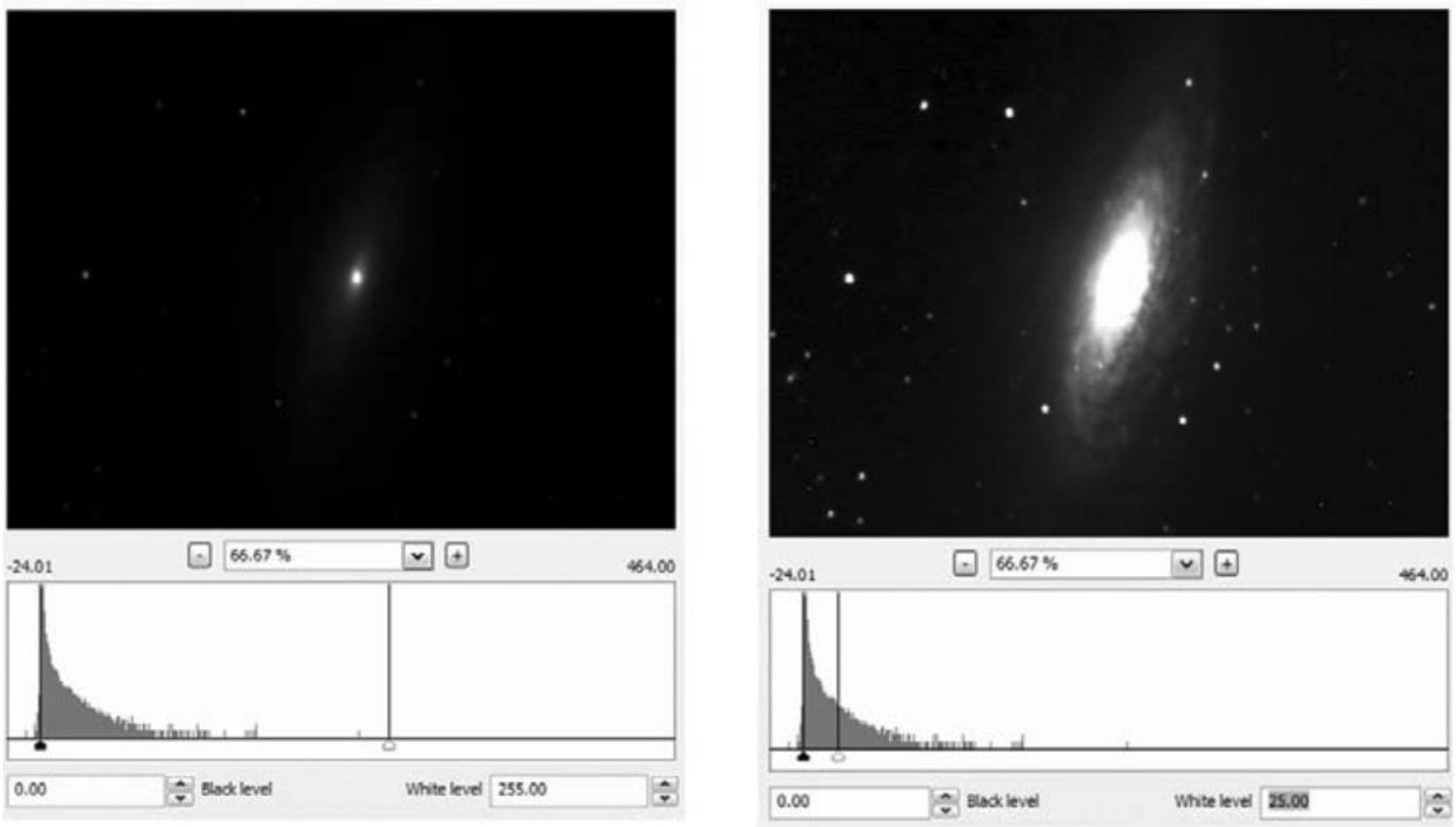


Figure 10.5. Two different linear stretches on the same image using FITS Liberator.

White level 255 → 25

10.3 PRINCIPLES OF IMAGE ANALYSIS AND PROCESSING
10.3.1 Displaying images

Color Representation in Images

- Gray Scale: Uses 256 shades of gray.
- False-Color: Maps intensity levels to colors for enhanced visualization. Useful for non-visible wavelengths (infrared, X-ray).
- True Color: Images adjusted to balance colors naturally, but often different from what the human eye would perceive.

Filter combination

- spectrally augmented color: Uses filters (e.g., replace red with infrared) to enhance red/blue contrast.
- substitute-filter color: Isolates specific features with narrow-band filters (e.g., highlighting methane absorption inJupiter’s clouds).
- emission highlighting (e.g., the use of narrow-band filters to enhance nebular emission light, such as pink H-alpha and green oxygen, against starlight)
- color translate images made at invisible wavelengths (e.g., representing X-rays in blue)
- enhanced color: Increases saturation to improve contrast (e.g., Voyager images of planets)
- two-filter color: Creates a balanced color image from two-wavelength data, interpolating the third.

10.3 PRINCIPLES OF IMAGE ANALYSIS AND PROCESSING
10.3.1 Displaying images

Examples of Color Enhancements in Astronomy

- GALEX Image: UV tail of Mira shown in blue.
- Infrared Images: Near-infrared wavelengths (J, H, K) mapped to visible colors (blue, green, red).
- Composite Images: Combining X-ray, radio, and visible light for multi-wavelength visualizations (e.g., Cen A galaxy).

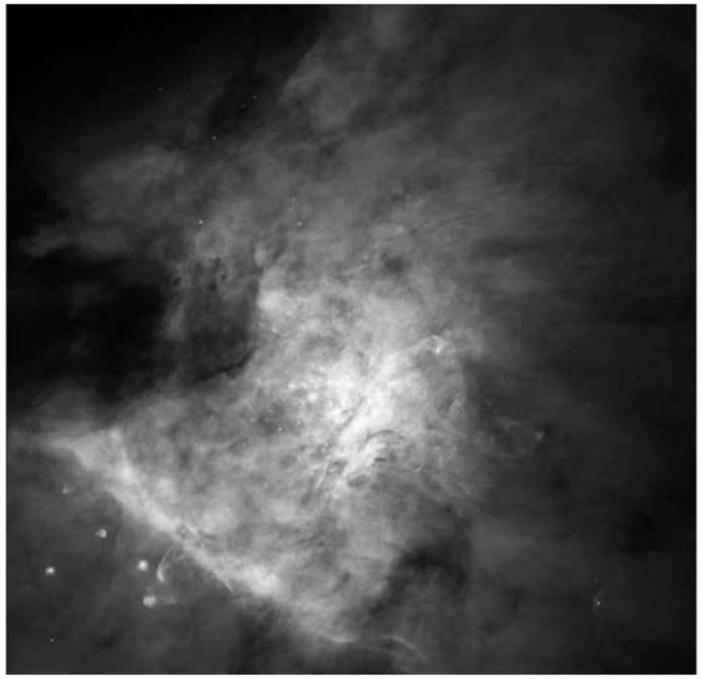


Figure 10.6. The Orion Nebula in certain narrow-band filters rendered here in gray but in false color in Plate 12 demonstrates the effectiveness of adding color for emission highlighting.

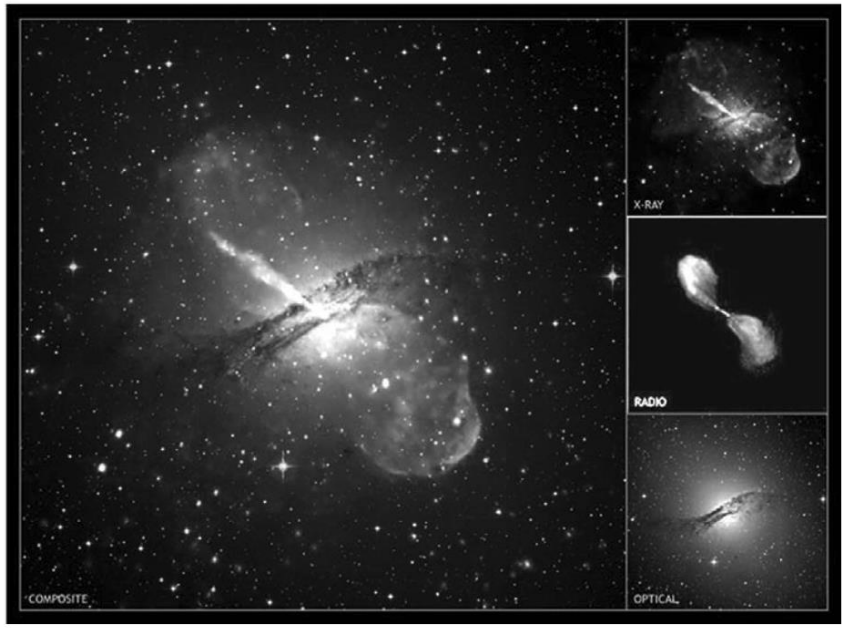


Figure 10.7. The active galaxy Centaurus A in X-ray, radio, and visible light combined into a composite false-color composite is shown here in grayscale. The color version is shown in Plate 4 and the book cover. Credit: see color section.