10 Image processing and analysis

Computers in astronomy:

- Control of telescopes and instrument
- Acquisition of digital data
- Image display, analysis, simulations…

Contents of this section:

1. Explanation of terminology
2. Standard data file format(FITS)
3. Introduction of some basic concepts of high(low)-pass filtering

10.1 Computers

Role of computers in astronomy:

- Acquisition of digital data from detector
- Image analysis
- Data storage

⇒**Growth of computer technology is enormous**

10.1.1 Data acquisition and data transfer

Data rate: number of digitized pixel values transferred per second to the host computer

Time to digitize a pixel value: 20-100 μ s ⇒ Data rate:800000bits/s
Data rate required to capitalize CCD: 100kb/s

Choice of ADC

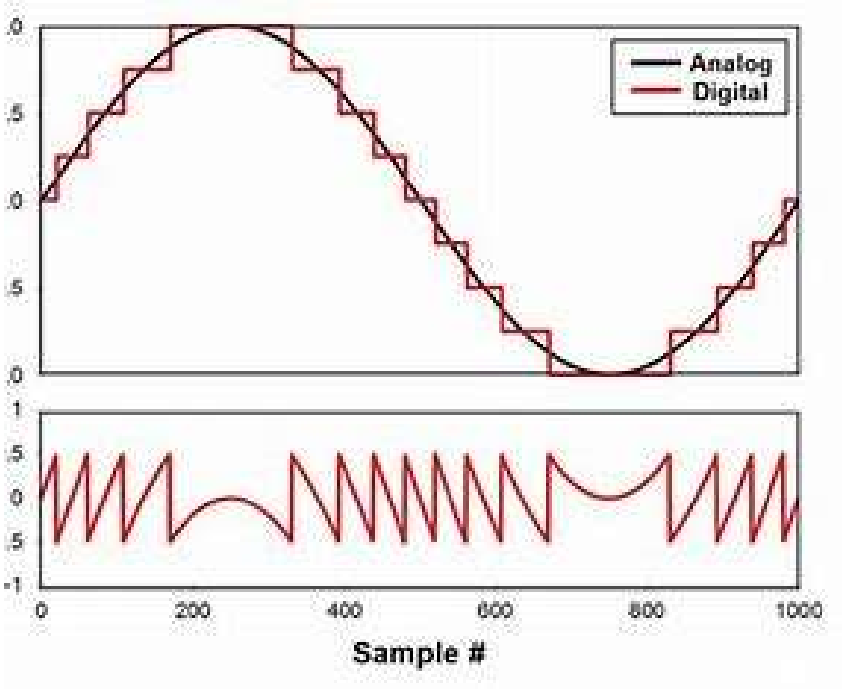
Data from detectors is affected by ADC
⇒Performance of ADC is important when selecting

Vital point:

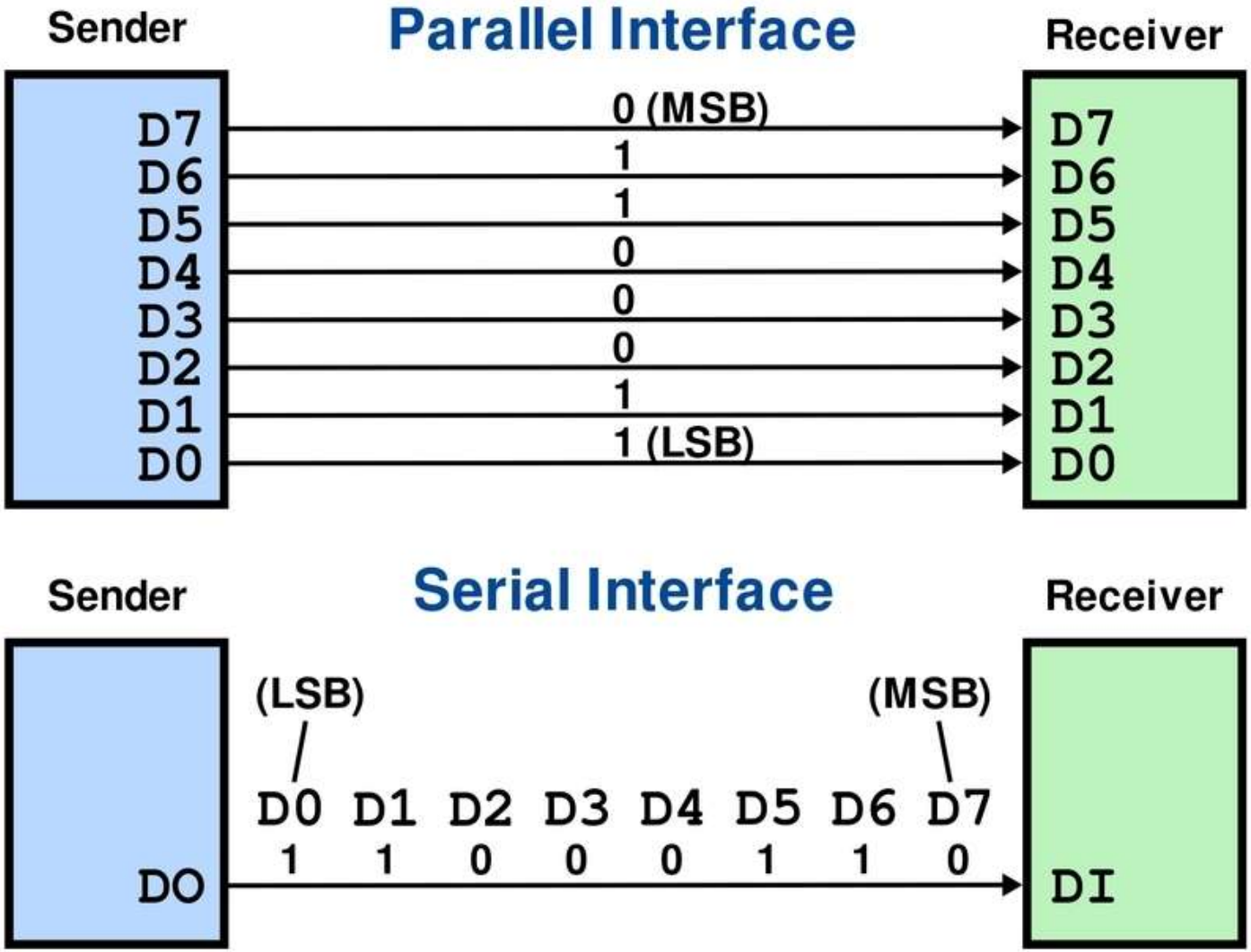
- Signal-to noise ratio
- **Digitized noise**

  Digitization process:
  Signals will be rounded up or down by half of “steps”

Appearance of low-noise CCD
⇒Effect of digitized noise becomes strong, relatively



Connection between ADC and computer



Parallel interface:

- High data rate
- Transferred by multi-way cable
- System is complex

Serial interface:

- Low data rate
- Transfer signal with a single line
- System is simple

Example of serial interface

RS-232-C, RS-432-C

- Represent logic “1” or “0” by positive or negative voltage(RS232-C), 0 or positive voltage(RS432-C)
- Classified as a legacy interface

LAN, Ethernet

- Use 2 cables to transfer (one for sending, the other for receiving)

Optical fiber

- Realize low-noise, high-speed, and high capacity transformation

10.1.2 Data file formats

Photon image and charge image:

- Transferred by CCD
- Location of the charge image=location of arriving photons
- Amount of charge∝number of photons

Flow of treatment of image

① Readout process(charge⇒voltage)
② Digitalization by ADC
③ Stored as 2D-array in computer(digitalized image, master data)
④ Copy and analyze observation data (master data is usually locked not to be changed)

FITS data file format

FITS file consists of 3 parts, ①header, ②data and ③tailer

①header

- Each lines has keyword(max 8 words), values and comments
- Header must contain following keywords, “SIMPLE”, “BITPIX”, “NAXIS”, “NAXISn”(n=1, 2, ･･･), and “END”
- Users can write original keywords

②data

- Contains continuous sequences according to the NAXIS parameter

③tailer

- Represents the end of FITS file by null(00)

SIMPLE	Has the value in byte 30 of either T (true) of F (false). It is simply a statement of whether or not the file conforms to the FITS standard.
BITPIX	An integer describing the number of bits in the data values. Options are 8, 16, 32 for 8-bit, 16-bit, and 32-bit unsigned integers. Floating-point data can be represented (e.g., −32 and −64 for 32-bit and 64-bit, respectively).
NAXIS	The dimension of the data array. If value is zero, no data follows. Value of 1 for 1-D data such as intensity values in a spectrum. For image data, NAXIS = 2 (e.g. rows and columns of CCD), and NAXIS = 3 would be used for a data cube of spatial coordinates vs. velocity. The maximum value of NAXIS is 999.
NAXIS1, NAXIS2, NAXIS*n*	Each specify number of elements along that axis, with the convention that NAXIS1 is the axis whose index changes most rapidly and NAXIS*n* is the axis whose index changes the slowest. For example, in a CCD image the number of columns would go in NAXIS1 and the number of rows in NAXIS2.

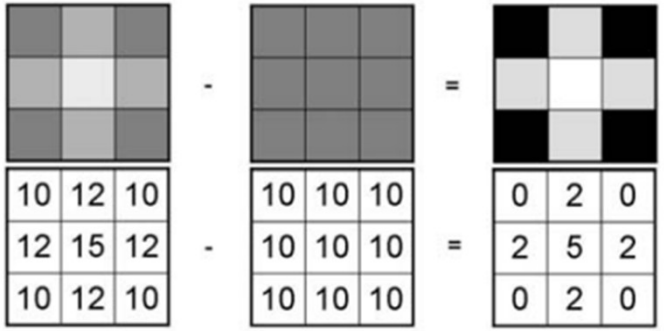
10.2 Data reduction and analysis systems

Data reduction = image processing (for most astronomers)

Before data analysis, we have to execute subtraction of dark and flat fielding

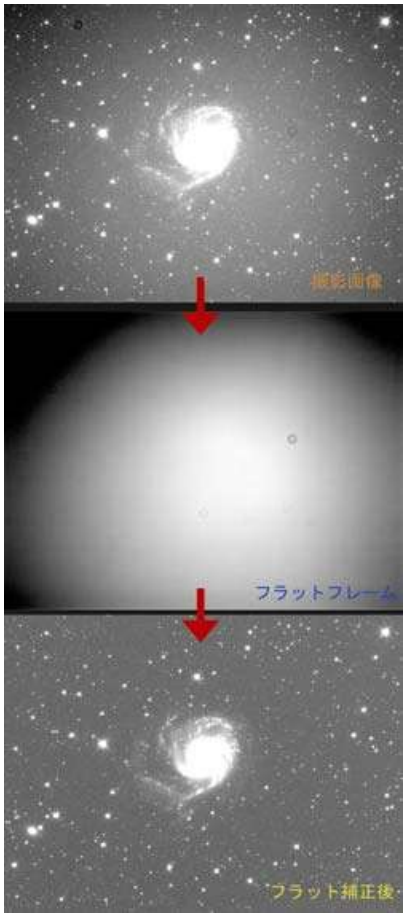
Subtraction of dark

Remove counts from dark current



Flat fielding

Correct the efficiency of each pixel



Example of image processing software in astronomy

- AIPS(for radio interferometric data)
- IRAF(general-purpose software system for astronomical data)
- STSDAS(contains data reduction pipeline for HST)
- ESO(for construction of data processing)
- STARLINK(help users use computers to analyze their observation)
- IDL(useful for modelling and simulations)

10.2.1 The IRAF package

IRAF consists of several packages, like

- **system(system utilities)**
- **language(command language)**
- **dataio(data input and output)**
- **images(general image processing)**
- **noao(astronomy package)**
- **plot(general graphics utilities)**

We have to call packages to available commands, except system and language packages

system	*language*	*dataio*	*images*	*noao*	*plot*
allocate deallocate devstatus directory help lprint	bye ehistory eparam logout lparam print	rfits	hedit imarith imcopy imdelete imheader imhistogram imrename imstatistics *tv* display	*digiphot* *apphot* qphot *imred* *ccdred* combine *proto* fixpix	contour

Example of IRAF task usage
①data analysis with dataio package

```
cl >dataio
da>devstatus mycomputer!mta   [This checks whether or not the tape/disk drive is
                               already allocated.]
da>allocate mycomputer!mta
da>help rfits                 [Hit spacebar to advance a page; "q" to quit.]
da>epar rfits                 [epar or eparam is shorthand for edit parameter
                               settings associated with this task.]
```

②display image on screen

SAOimage DS9: software for display FITS image

```
cl >images
im>tv
tv >display imagename      [Where imagename might be raw012(.imh); try dev$pix
                           for a built-in image of M51.]
```

DS9 includes useful tasks, like multi image buffer, scale map, and algorithm.

③Primary treatment(dark subtraction and flat fielding) with IRAF

```
cl>images
```

[Enter the major package. Read the help file and edit the epar file. The key parameters are the operand 1 (the first file), op (+, −, * , /), operand 2 (the second file), result (the output file).]

```
im>imarith file1–file2 outputfile
```

[You can also type the filenames directly on the command line.]

Before executing commands, we have to check output filename and text of command

④Compile user script

#: comment out
Variable: must be declared by type at the beginning of the body of the script

“while” and “if-then” statements are avaliable

```
# IRAF SCRIPT addimages.cl
# script to add two images and divide by a constant
# first declare the task name and the required parameters
  procedure addimages (image1, image2, image3)

# next list all the parameters, and give the prompt for them that will appear in the
# param file

  string image1          {prompt="first image"}
  string image2          {prompt="second image"}
  string image3          {prompt="output image"}
  real number=1000       {prompt="number to divide by}
  bool divbool           {prompt="divide by number?"}
# notice that the optional parameter is given a default value
# start the main program
  begin

# add image1,image2 to make image3
# notice that the required parameters are entered in parenthesis and separated by
# commas
  imarith (image1, "+", image2, image3)

# check to see if division should be done
# notice that the commands inside {} are only execute if divbool is "yes"
  if (divbool==yes) {
      imarith (image3, "/", number, image3)
        }
# end program
  end
```

⑤aperture photometry

qphot task: simple aperture photometry
DAOPHOT: fitting PSF for “crowded” fields

⑥Sextractor package

Classify galaxies from stars in deep images