

PHY 517 / AST 443:

Observational Techniques in Astronomy

Lecture 3:
CCDs /
FITS files

StarAlt

you have to pay attention to the format of your coordinates!

The screenshot shows the StarAlt software interface. Under the 'Mode' section, 'Staralt' is selected. In the 'Night' section, the date is set to 06 September 2017. The 'Observatory' section contains a dropdown menu with 'Roque de los Muchachos Observatory (La Palma, Spain)' highlighted, which is circled in red. Below the dropdown, instructions say 'Select one above or specify your own site with this format: Longitude(°East) Latitude(°N) Altitude(metres) UTC offset'. An example is given: 'Ex.: 289.2767 -30.2283 2725 -4'. A text input field is present below the example.

How are Mt. Stony Brook's coordinates correctly specified?

- A) 41 73
- B) 73 40
- C) -41 73
- D) 287 41

Coordinates: 40.914224°N 73.11623°W

[Stony Brook University](#)



Finding Charts

HW asked for chart orientation “as if you were looking at the sky with the naked eye”

WHAT NORTH-SOUTH ORIENTATION WOULD YOU LIKE? *

North Up North Down

WHAT EAST-WEST ORIENTATION WOULD YOU LIKE? *

East Left East Right

Which of these is correct?

- A) North Up, East Left
- B) North Up, East Right
- C) North Down, East Left
- D) North Down, East Right

Significant digits

Code output with way
too many digits:

$$99.123456789 \\ \pm 0.004556789$$

Round the uncertainty
to one (or two) digits:

$$0.00455679 \rightarrow 0.005$$

The location of this
digit tells you the
location of the last
significant digit:

$$99.123 \\ \pm 0.005$$

Voila:

$$99.123 \pm 0.005$$

Plagiarism

(a) Which of these are examples of plagiarism? (More than one answer may be correct.)

- i. Copying your lab-mate's introduction section of a lab report.
- ii. Taking somebody else's lab report, slightly modifying each sentence / paragraph, and submitting it as your own.
- iii. Copying your buddy's telescope proposal for a Keck telescope, and submitting it as your own.
- iv. Stating a "fact" from wikipedia without citing the original source.

(b) What happens when you get a "Q" grade? (Answer yes/no/maybe.)

maybe i. Immediate expulsion from school.

yes ii. You lose your scholarship.

maybe iii. Immediate "F" for the course grade.

yes iv. You have to take the "Q" course.

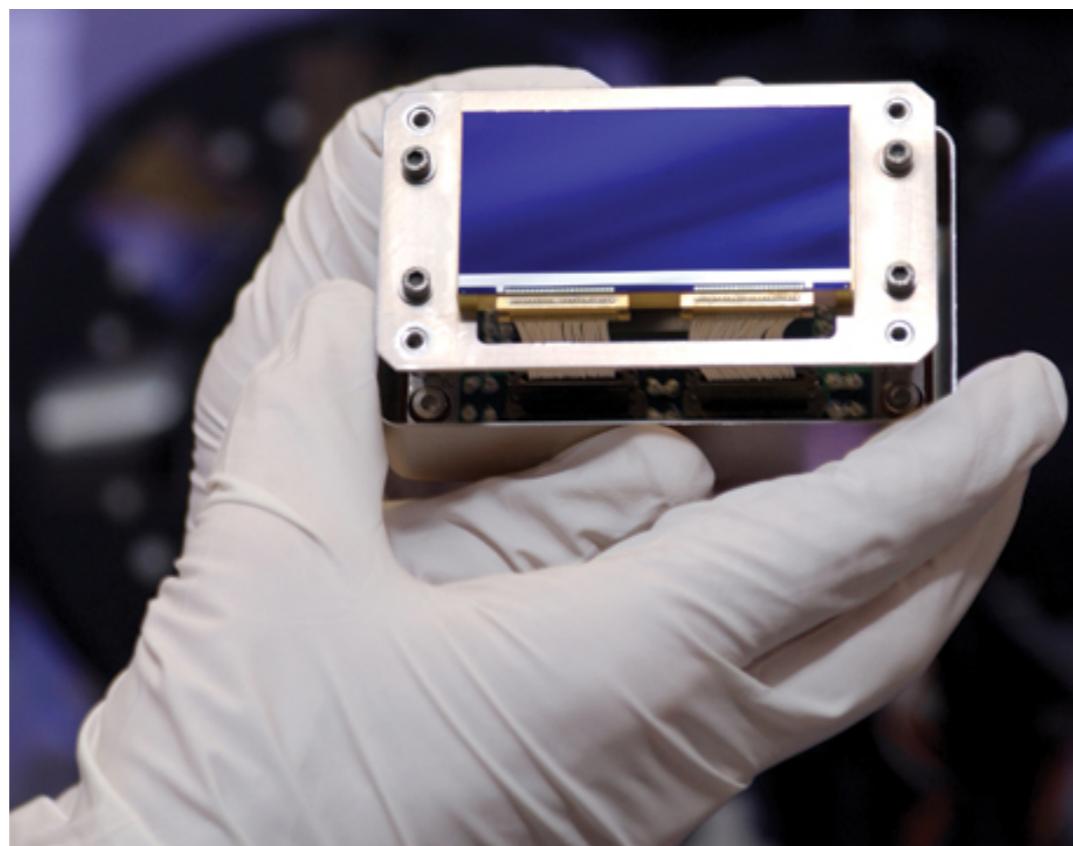
Note on Astro Computing cluster

- you will probably want to do at least some of the analysis steps on the cluster (*uhura* and *vulcan*)
- your data and scripts should go into `/astrolab/Fall_2021/username`
- only *uhura* and *vulcan* have all of the required software
 - *make sure to back up your data!!!*
- log out - do not block a machine by leaving it in screensaver mode; if you do, you have to write a letter of apology to Prof. Swesty
- change your password and keep it safe - else, see above

CCDs

CCDs

- CCD: “charge-coupled device”
- CCDs are the detectors of choice over much of the electromagnetic spectrum (X-rays to infrared)
- replaced photographic plates
- similar to detectors found in digital cameras



e2v

Figure 3. Kepler CCD in handling jig.

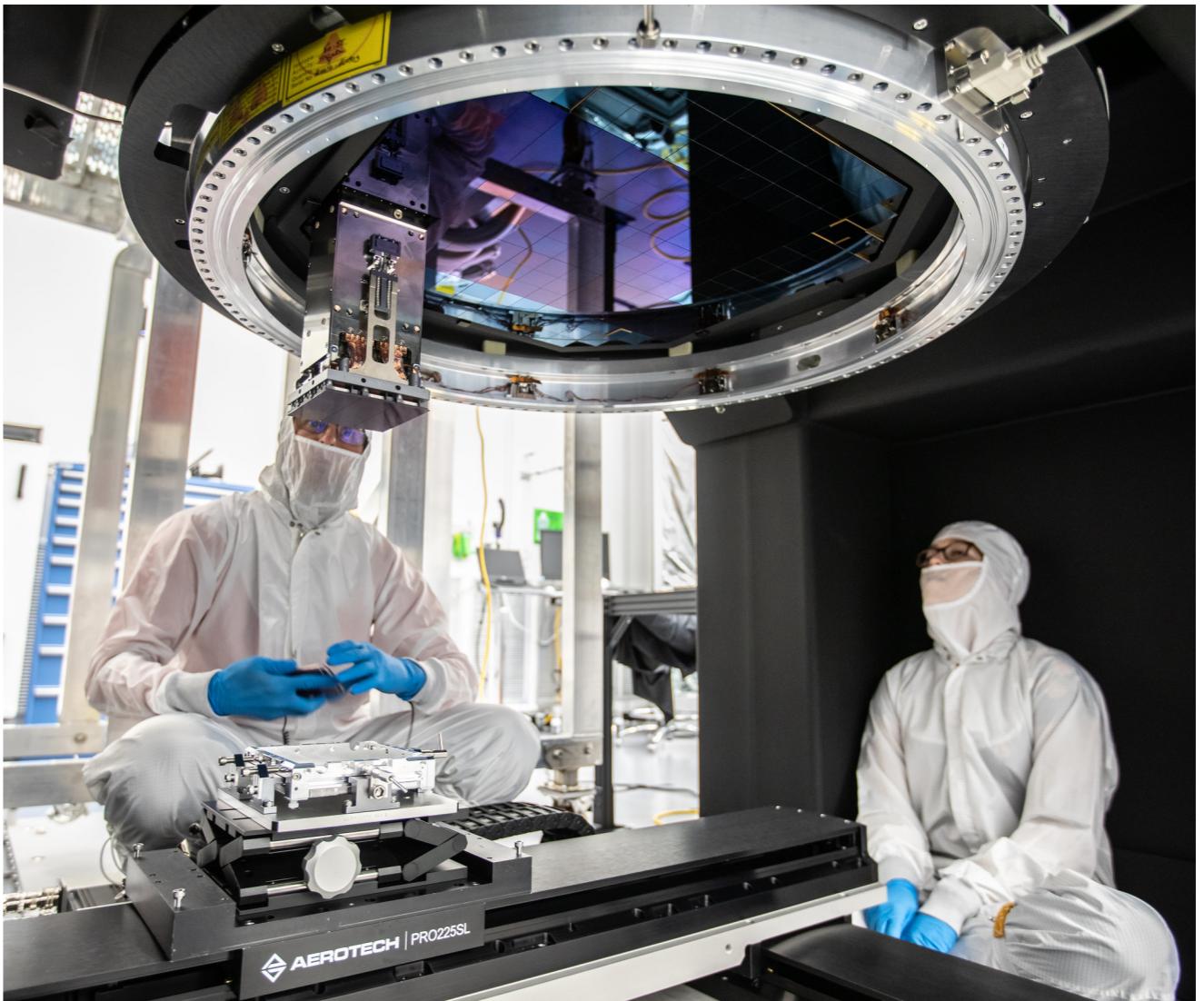
CCDs - Advantages

- (nearly) linear response
- high sensitivity
- low noise (especially when cooled)
- built-in digitization

$$N_{\text{electrons}} \propto N_{\text{photons}}$$

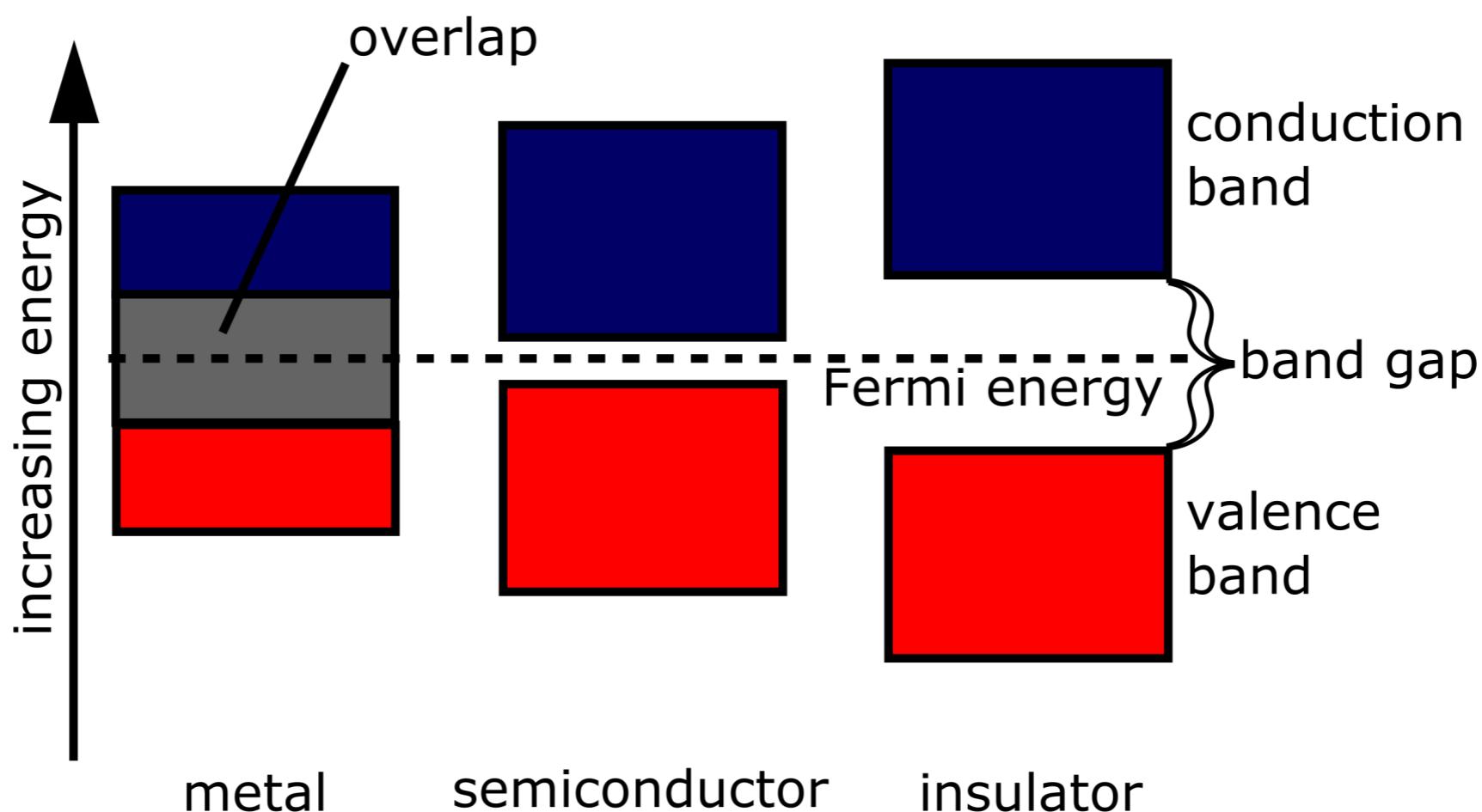
LSST Camera at SLAC

<https://www6.slac.stanford.edu/news/2020-09-08-sensors-world-largest-digital-camera-snap-first-3200-megapixel-images-slac.aspx>



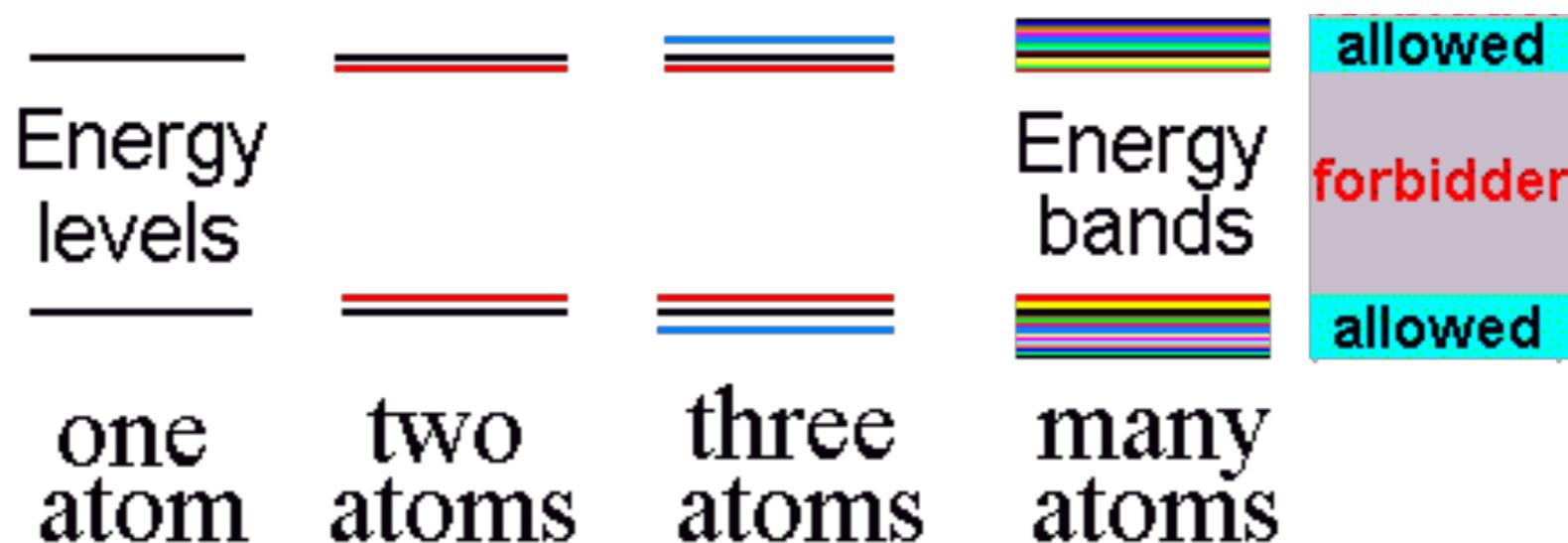
Semi-Conductors

- CCDs are made of semi-conducting silicon wafers
- key feature: small energy gap between “valence band” (energy levels of outermost bound electrons) and “conduction band” (energy levels of free electrons)



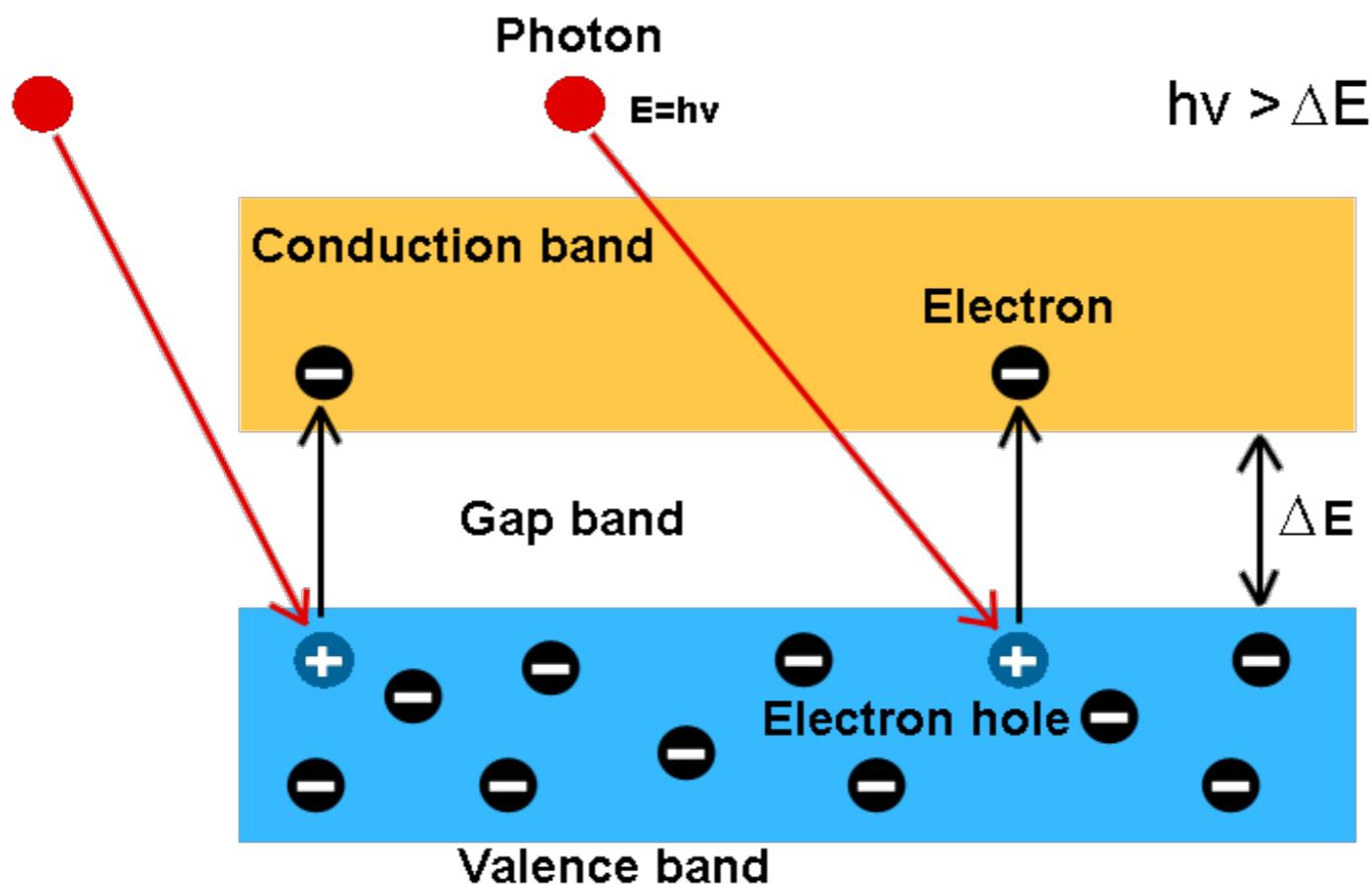
Why energy bands?

- single atom: discrete energy levels
- two atoms:
 - outer electron orbitals overlap
 - Fermi exclusion principle still holds → energy levels split to accommodate electrons
- many atoms: energy bands



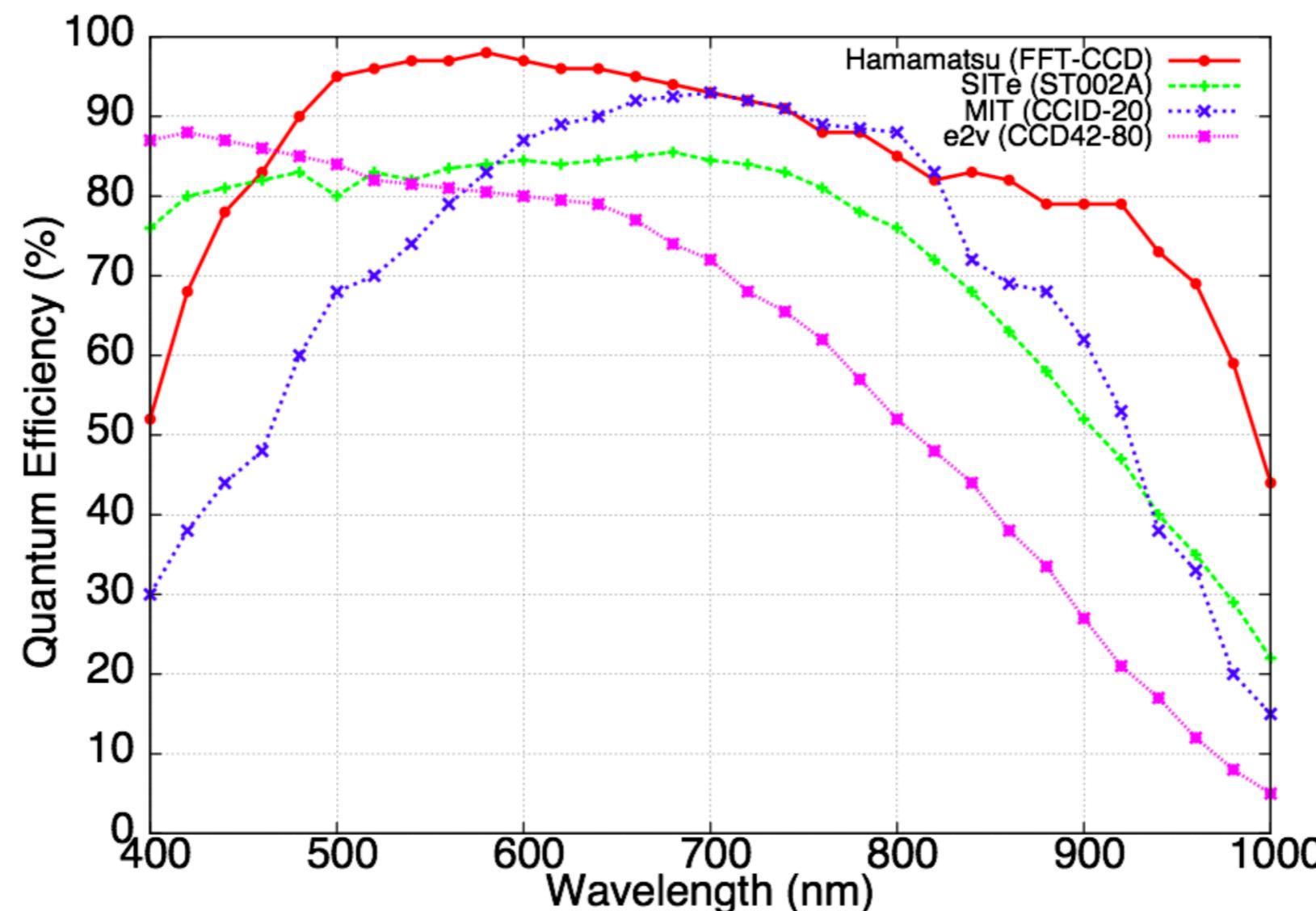
Photoelectric effect

- light is quantized, “photons” $E = h\nu$
- when a photon is absorbed, the energy is transferred to an electron → “jumps” into conduction band



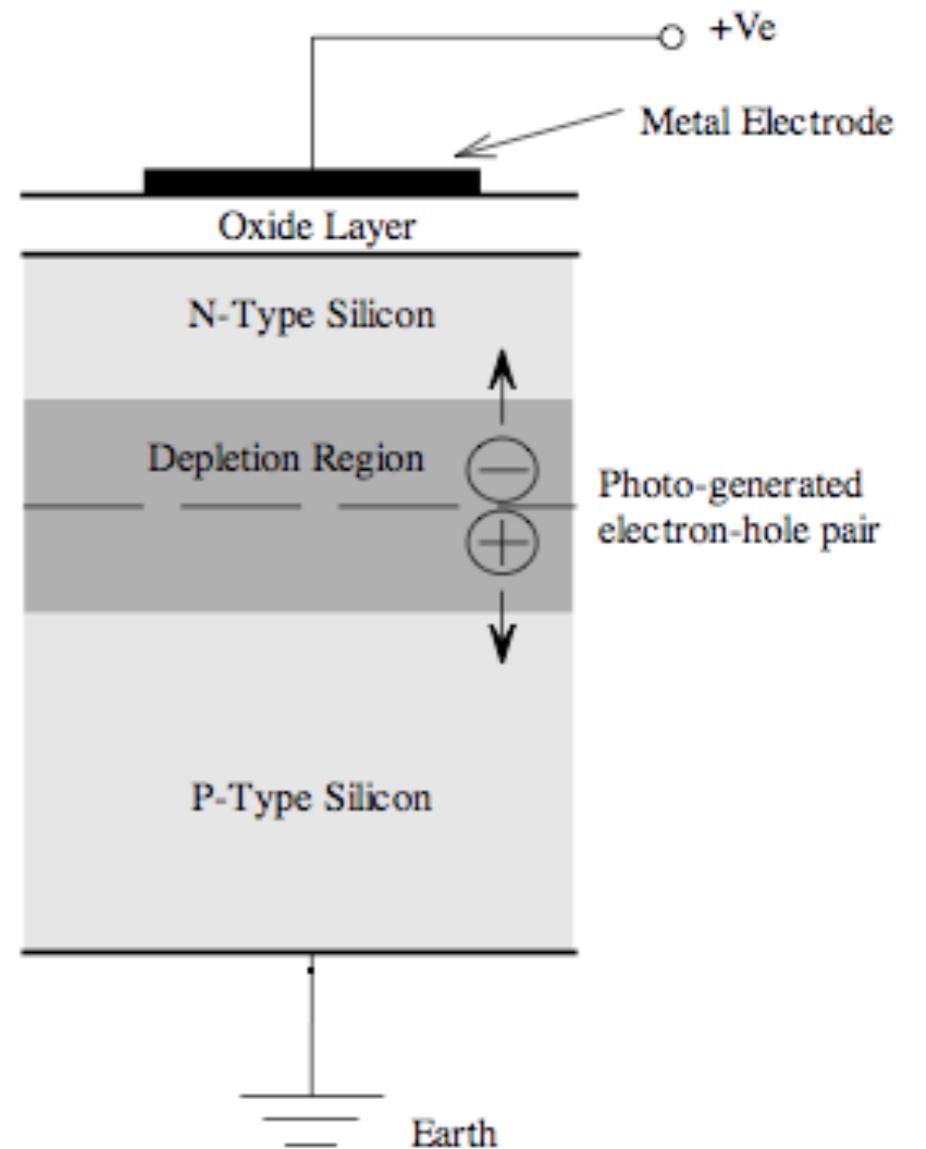
CCD Quantum Efficiency (QE)

- fraction of photons that are absorbed
- depends on wavelength
- different technologies lead to red vs. blue optimized CCDs



One pixel

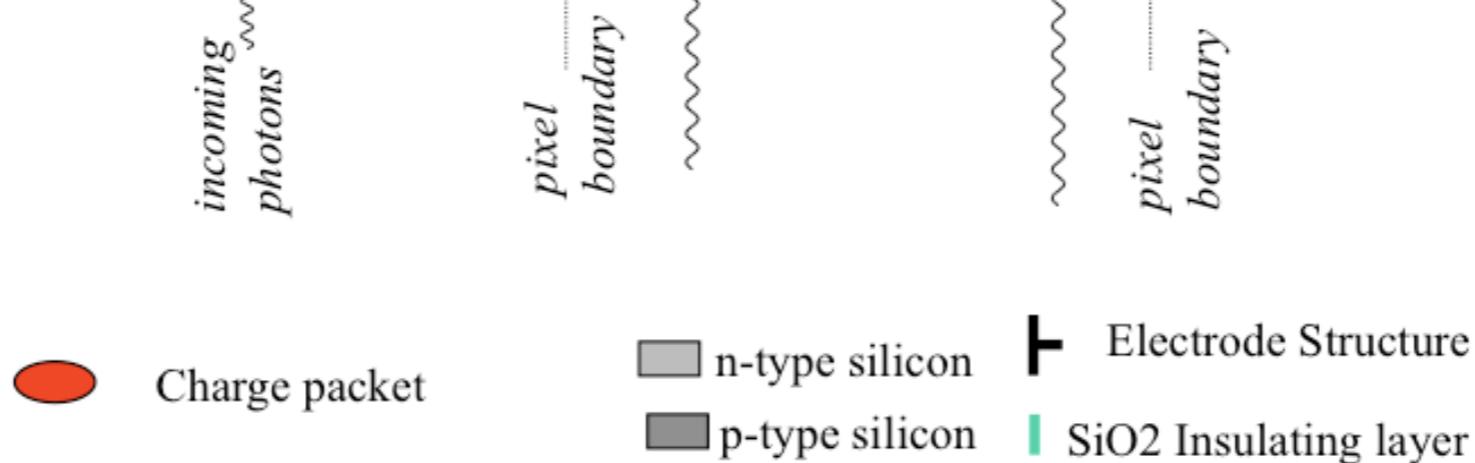
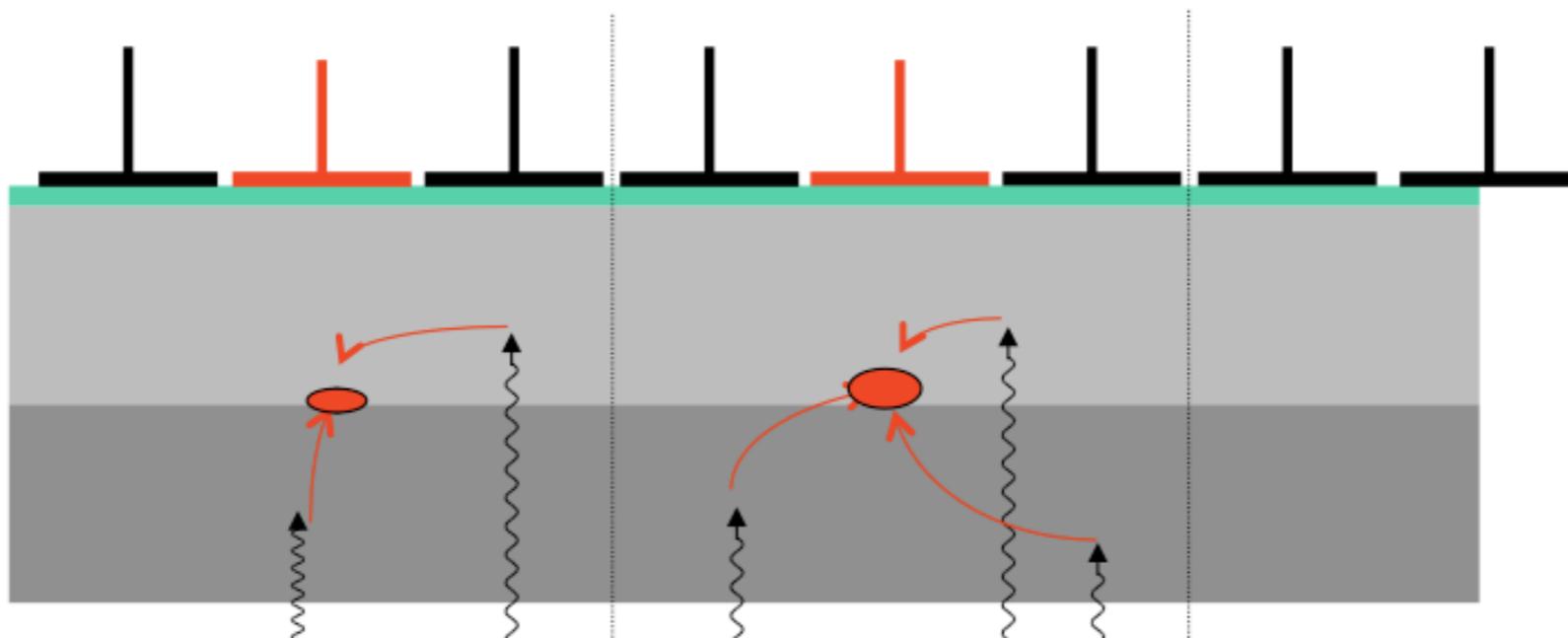
- apply an electric field to keep electrons / holes separated



Microsoft

Many pixels

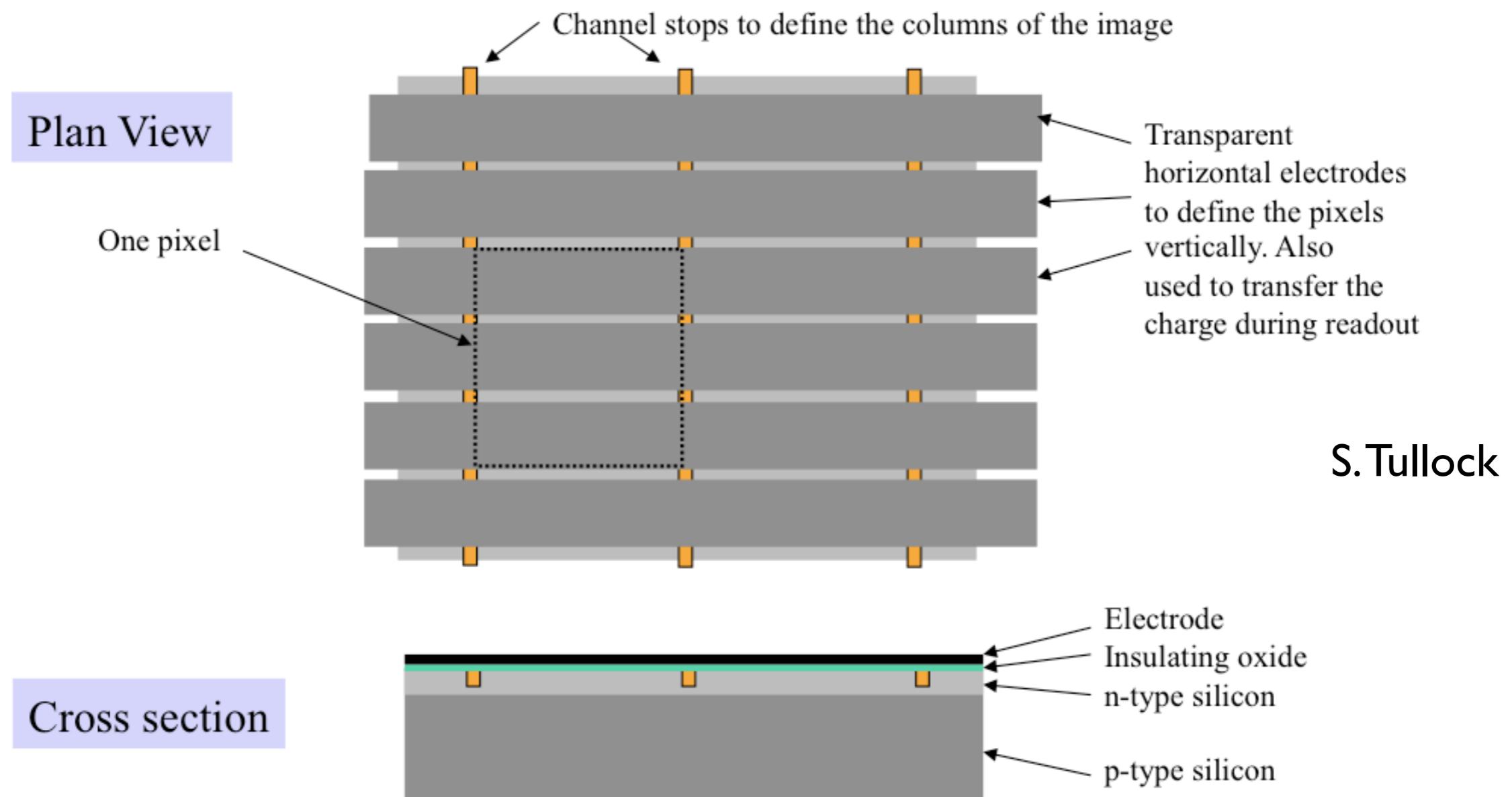
- pixels are defined by the electric field generated by the applied electrodes



S.Tullock

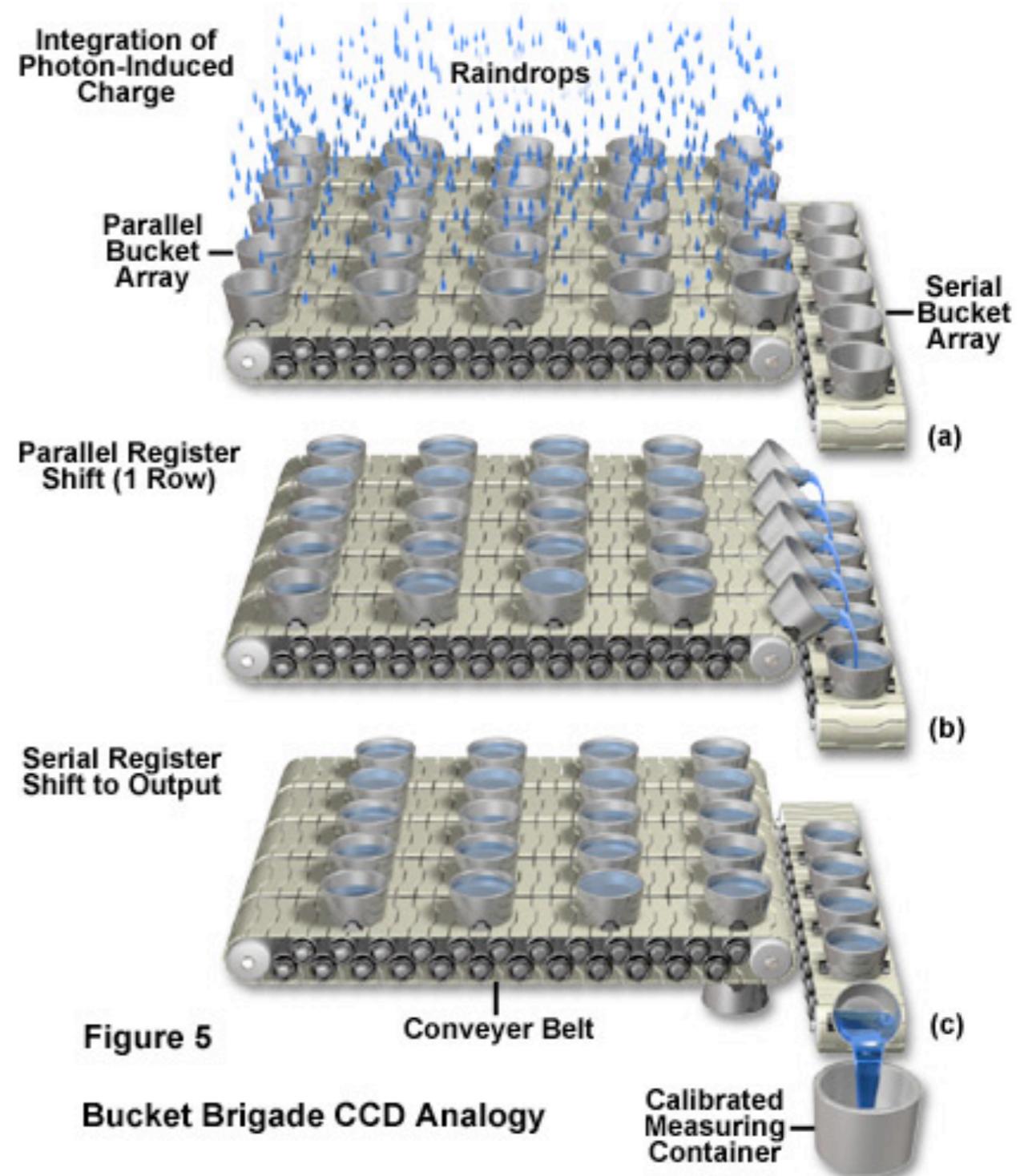
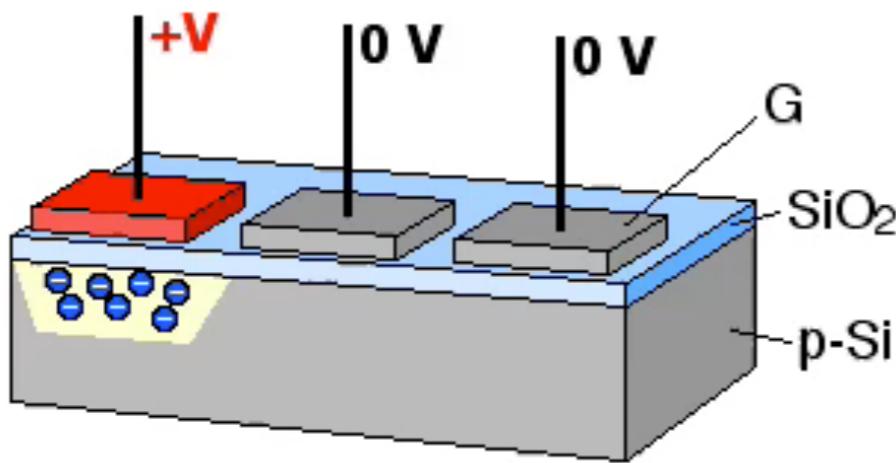
Many pixels

- ... and by insulator strips between columns



Reading out CCDs

- “rainbuckets on conveyor belts” analogy
- 1 conveyor belt = 1 CCD column
- in practice: modulate the electric fields to move pixel charges



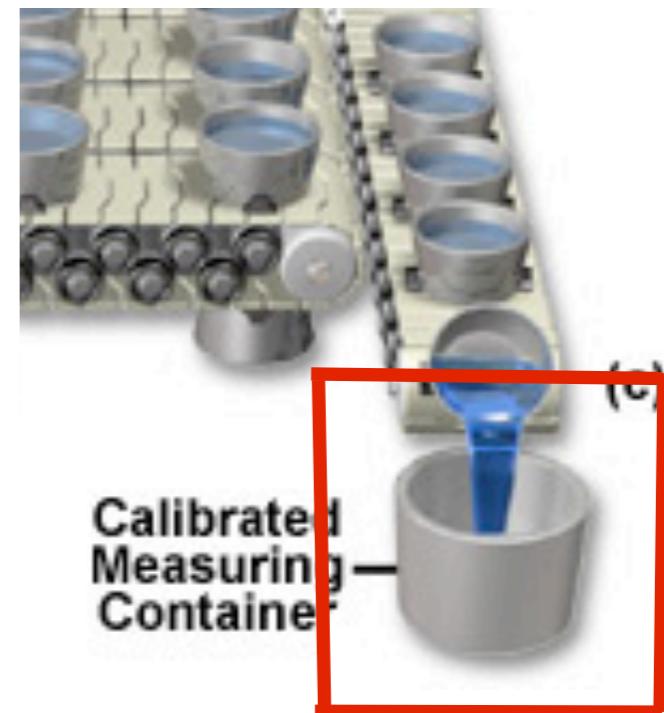
Bucket Brigade CCD Analogy

Cold Spring Harbor Protocols

Assembling the Image

- each charge collection is passed to an amplifier and analog-to-digital converter (ADC)
- final output: “counts” or ADUs (analog-to-digital units) → *integer value*
- can apply rescaling: “gain”

Cold Spring Harbor Protocols



$$\text{gain } G = \frac{N_{\text{electrons}}}{N_{\text{counts}}}$$

Full Well Capacity

- each pixel can only hold a limited charge → *full well capacity*, of the order of 100 000 e⁻
- ADCs have a maximum output value, e.g. 16-bit = 2^{16} = 65536 counts
- gain should be chosen roughly so that ADC maximum ~ full well
- typically, gain ~ 2-4

Measured number of counts

$$N_{\text{electrons}} \propto N_{\text{photons}}$$

- we want to measure N_{photons} (or, flux)
- our CCD returns N_{counts}
- what else contributes to $N_{\text{counts}}?$

Read-out noise

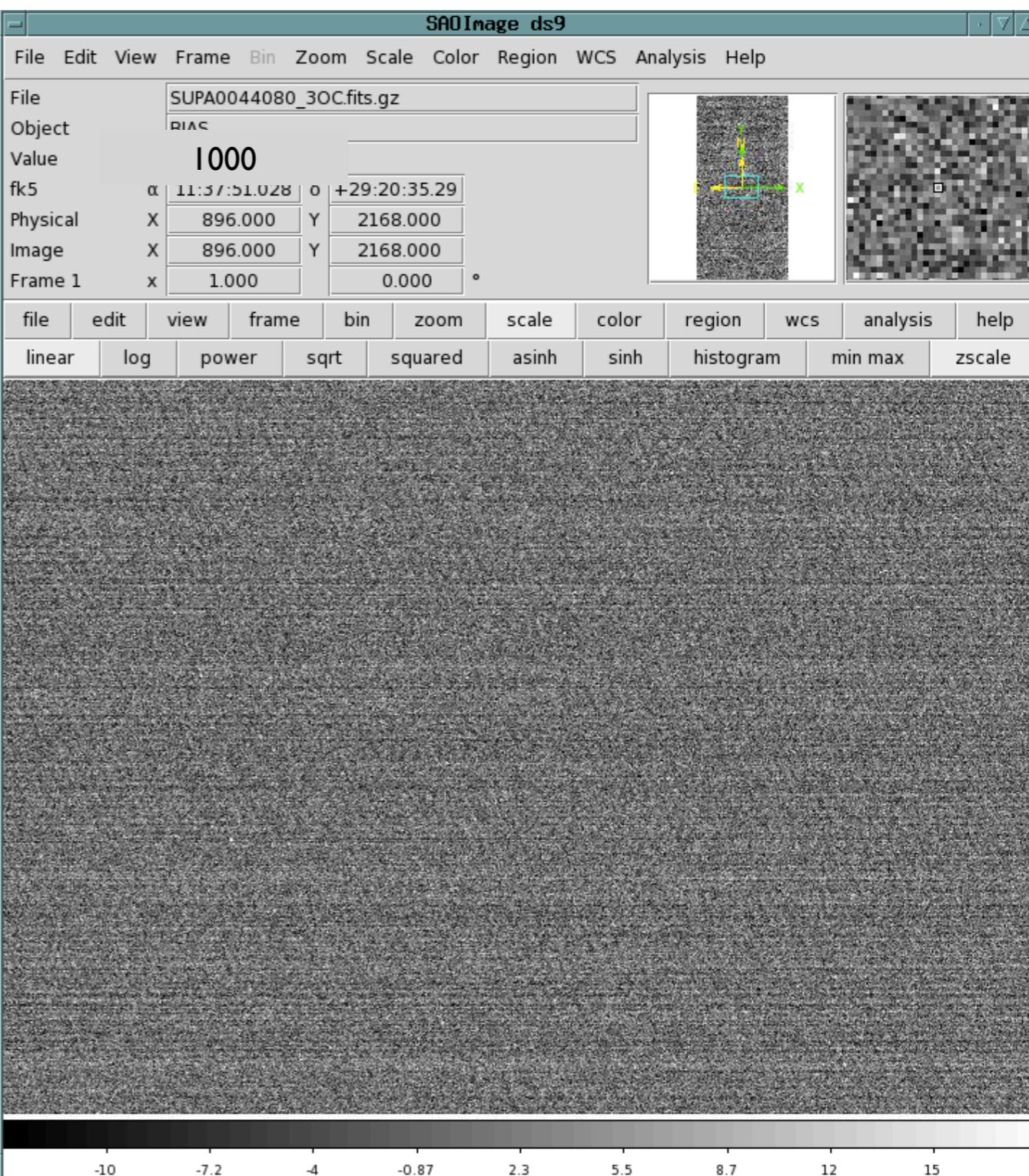
- **read-out noise:** noise produced by various electronics during read-out, e.g. the amplifiers
- the slower the read-out, the lower the read-out noise

Bias level

- **bias level**: an electronically induced offset which ensures that the ADC always gets a positive input
- the bias needs to be **subtracted** so that the counts are proportional to the signal
- note: the bias level is not a “counting process”, i.e. the standard deviation of the bias values (i.e. the read-out noise) is NOT $\text{sqrt}(N_{\text{cts}}[\text{bias}])$

Bias images

- images with 0s exposure time
- single bias frame: pixel values scatter around the bias level, width of this distribution is the read-noise
- master bias frame (median of many bias frames): read-noise is averaged out, remaining structure is due to electronics



Overscan region

- problem: the bias level may not be stable
- images on large astronomical cameras come with an *overscan* region
- each row is clocked out more often than there are physical pixels
- can be used as an in-situ estimate of the bias level
- use the extra pixels to estimate the bias level of each row; subtract it from entire row
- the overscan is subtracted from all images (including bias frames)

Dark current

- the energy gap in the semi-conductor is small → thermal noise leads to extra charge accumulation
- proportional to the exposure time
- cooling the CCDs significantly mitigates dark current
- professional astronomical CCDs cooled to -100°C → almost no dark current

Dark frames

- **dark frame:** images taken with closed shutter
- similar to bias frames; need to be subtracted
- (subtracting non-bias-corrected darks subtracts the bias, too)
- our cameras have no overscan and substantial dark current → we will use dark frames instead of bias frames
- *Q: What temperature and exposure time do the dark frames have to have?*

Flat-field

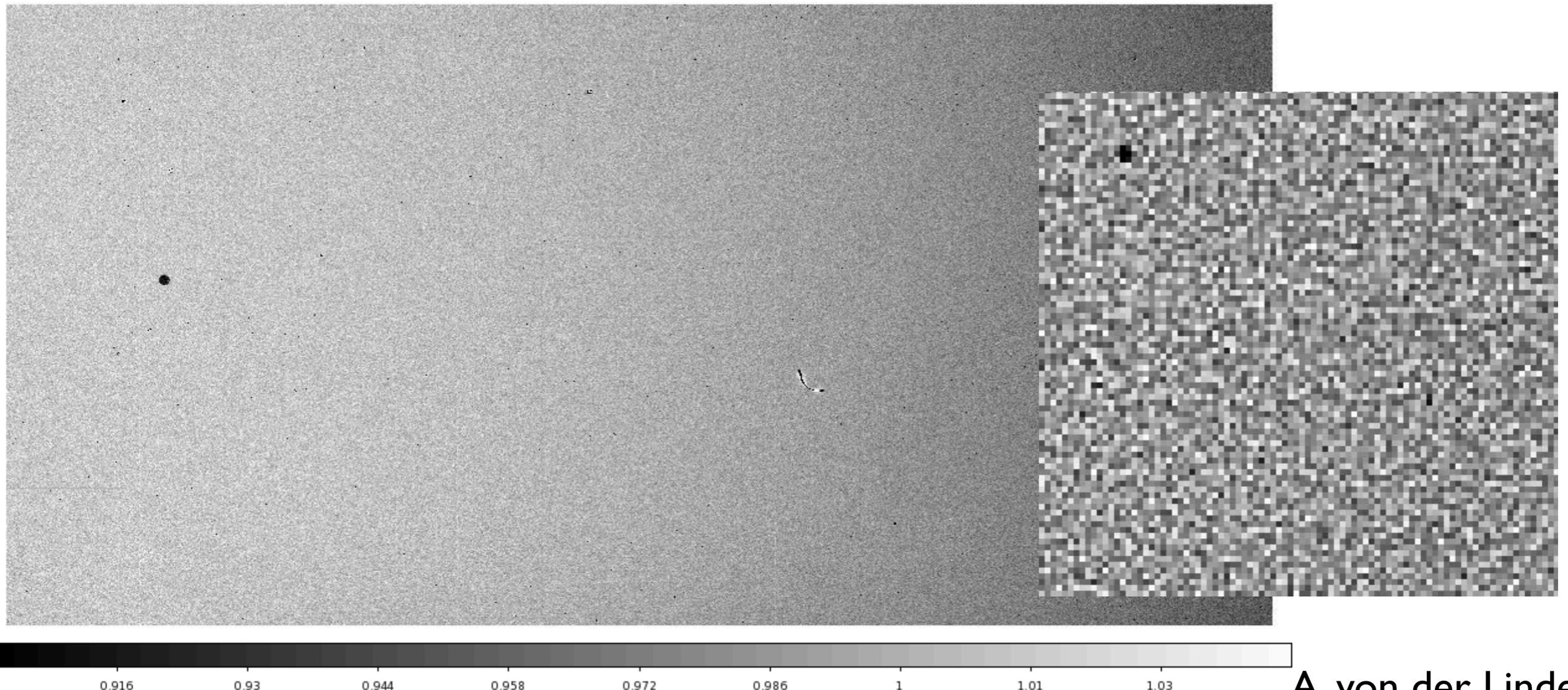
- the pixels in a CCD do *not* have uniform sensitivity
- due to variations in silicon crystal, electric field, pixel size, illumination (vignetting)

$$N_{\text{electrons}} = A_{ij} N_{\text{photons}}$$

- A_{ij} different for each pixel
- need to correct for differences for meaningful measurements

Flat-field

- flat-field: take an image of a spatially uniform source of light (e.g. the twilight sky, or a screen in the dome)
- input signal (N_{photons}) is the same for each pixel; variations in N_{counts} are due to different sensitivities



Flat-field

- flat-field is a *multiplicative* correction (unlike bias / dark)
- in practice: take a series of flat-field images
- correct each flat image by the appropriate dark frame
- average the flat-field images (reduces counting noise)
→ master flat-field
- ... *and then what?*

Types of flat fields

dome flats:

- ✓ easy
- ✓ constant conditions
 - not entirely uniform
 - different spectrum than astronomical objects

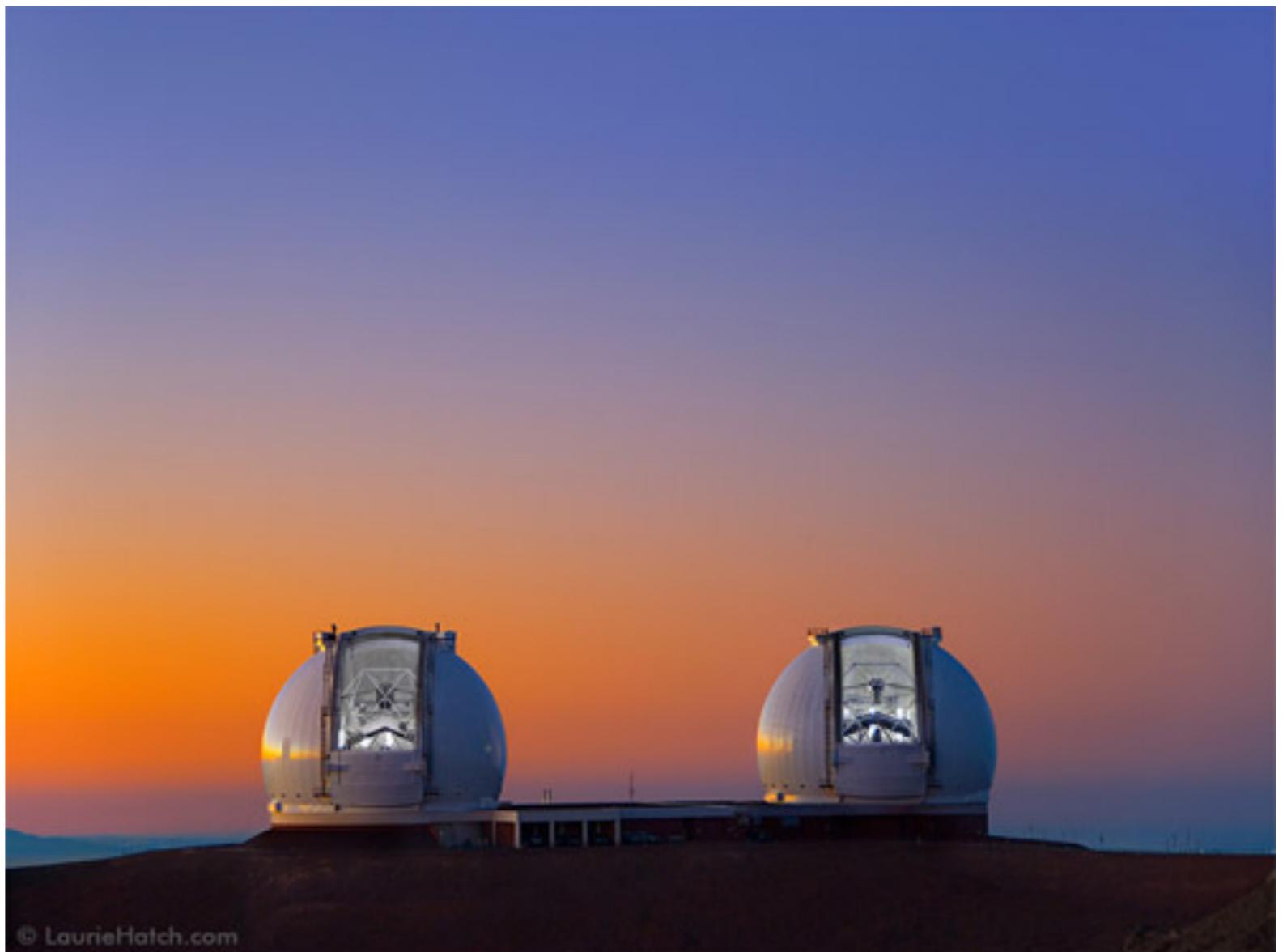


A. von der Linden

Types of flat fields

twilight flats:

- ✓ same “source”
- ✓ almost uniform
- variable
- difficult

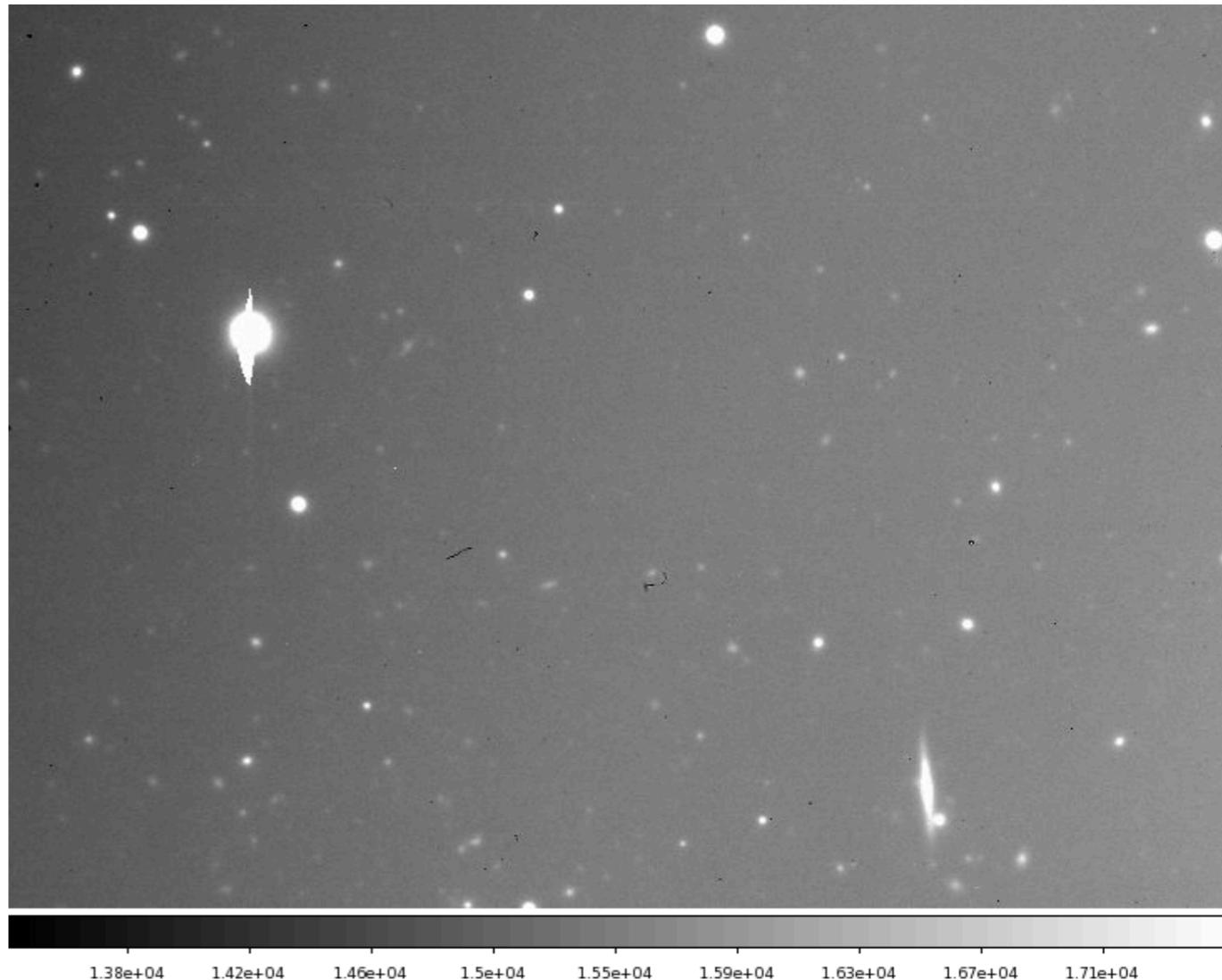


L. Hatch

Types of flat fields

night-sky flats: if observations of several different targets are taken in one night, can average these images into flat-fields assembled from the sky background (best to mask out detected objects)

- ✓ most similar to data
- ✓ uniform
- need “empty” fields
- need a lot of images

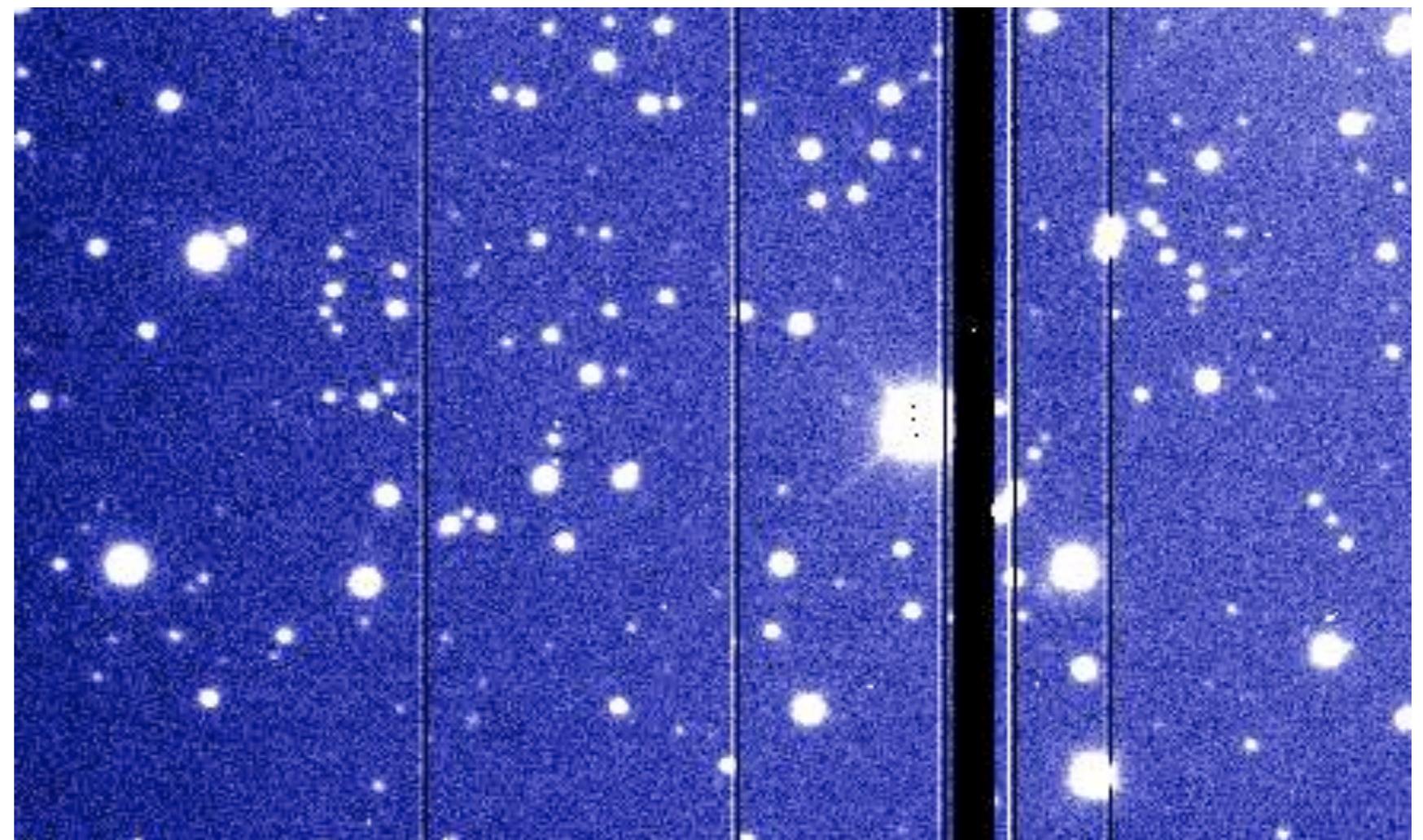


Artifacts

dead pixels / columns / rows: no (or little) response

hot pixels / columns / rows: very high noise

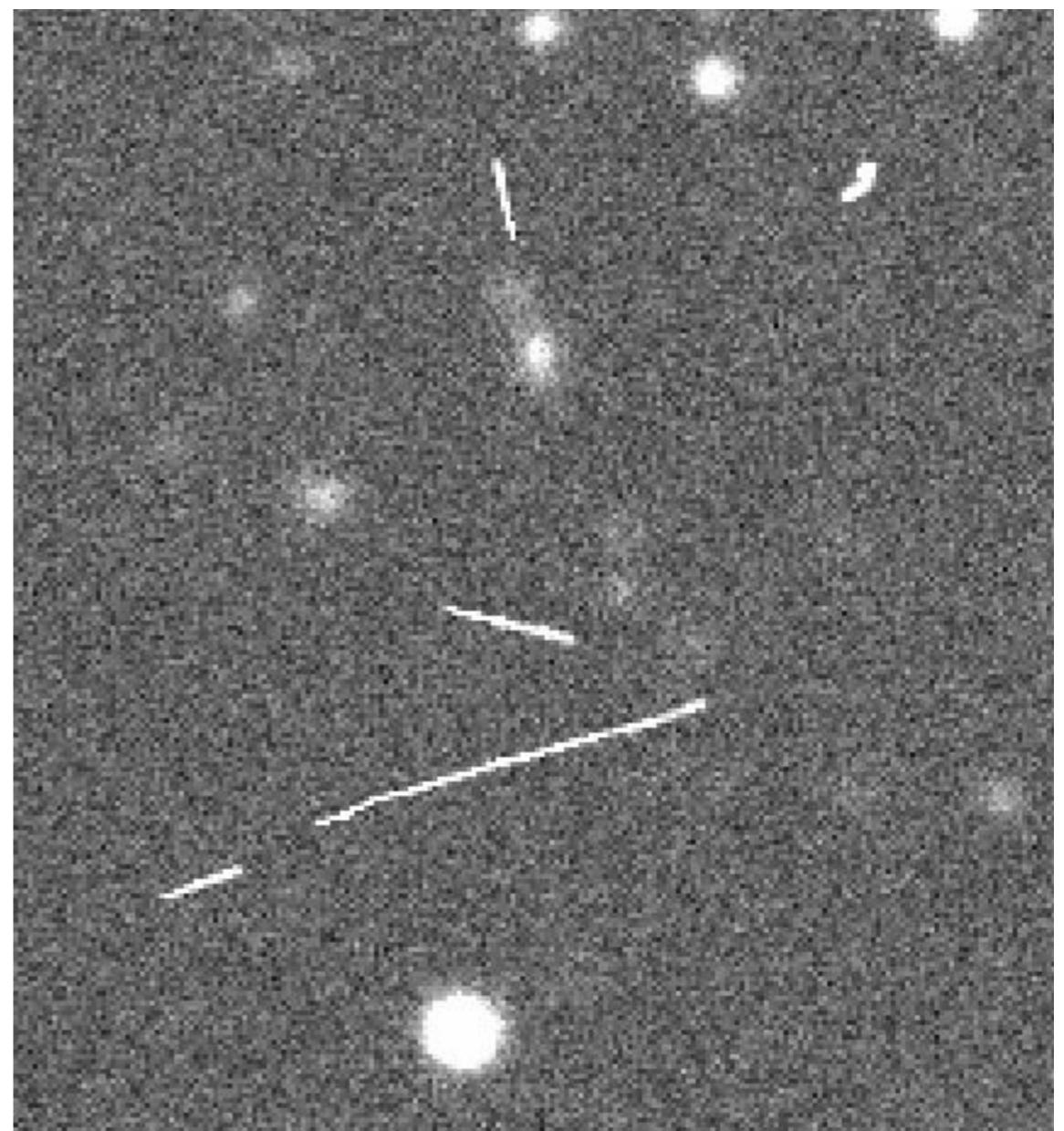
*signal is not
recoverable;
pixels need to
be masked in all
exposures*



Artifacts

cosmic rays: charged
particles hit the CCD

*need to be masked -
single exposure*

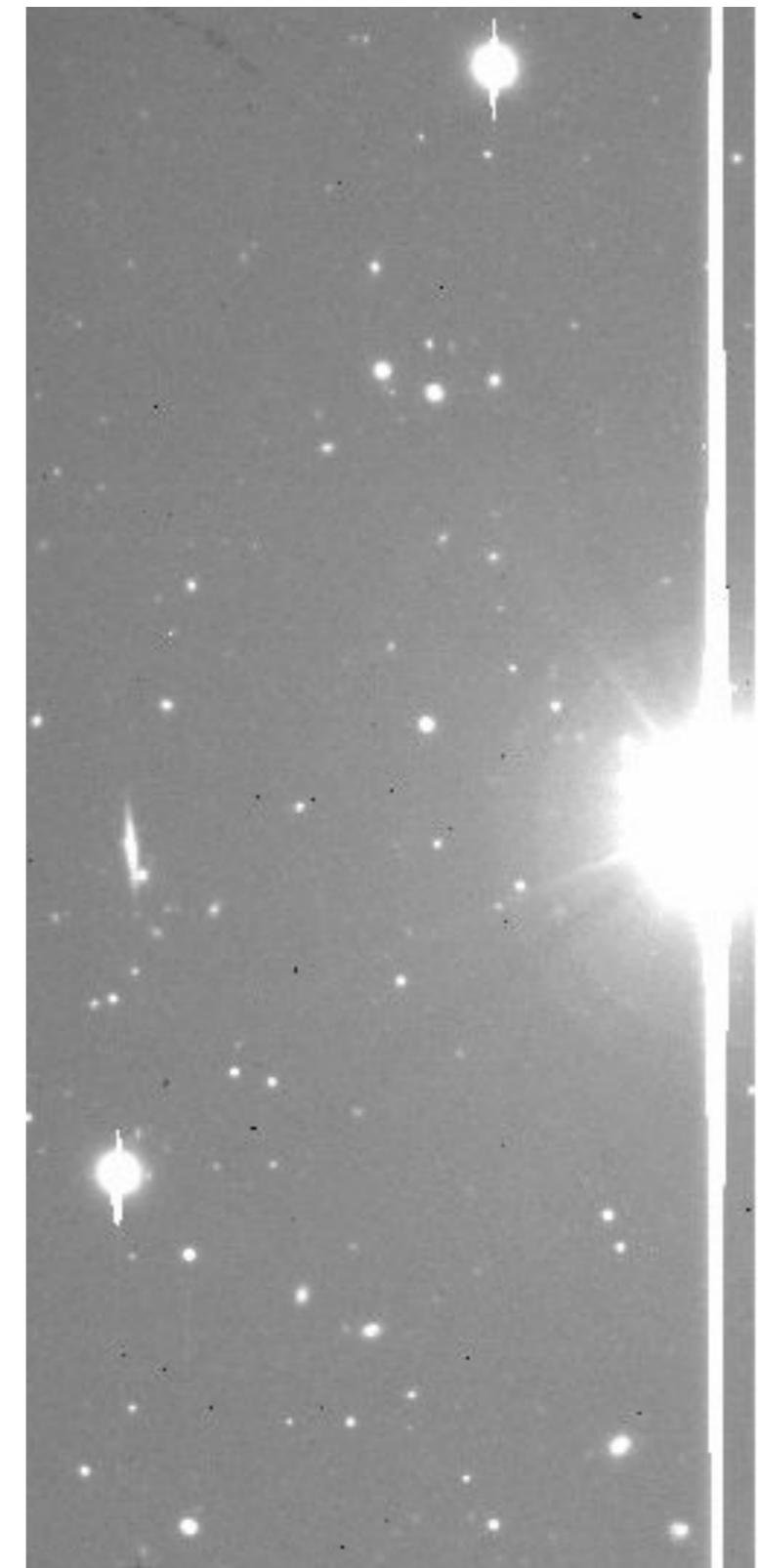


A. von der Linden

Artifacts

saturation spikes: when full well capacity is reached, electrons spill over into neighboring pixels

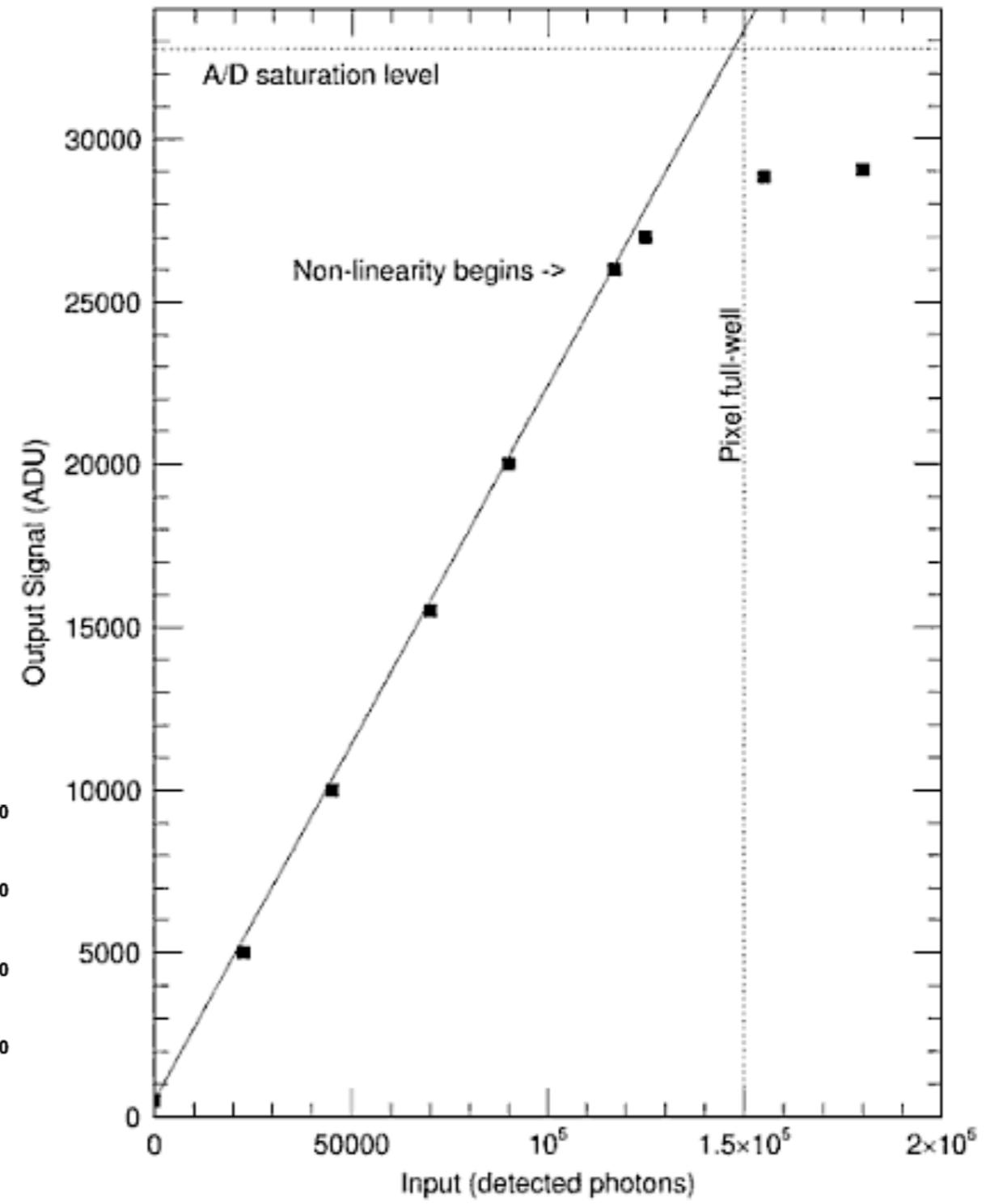
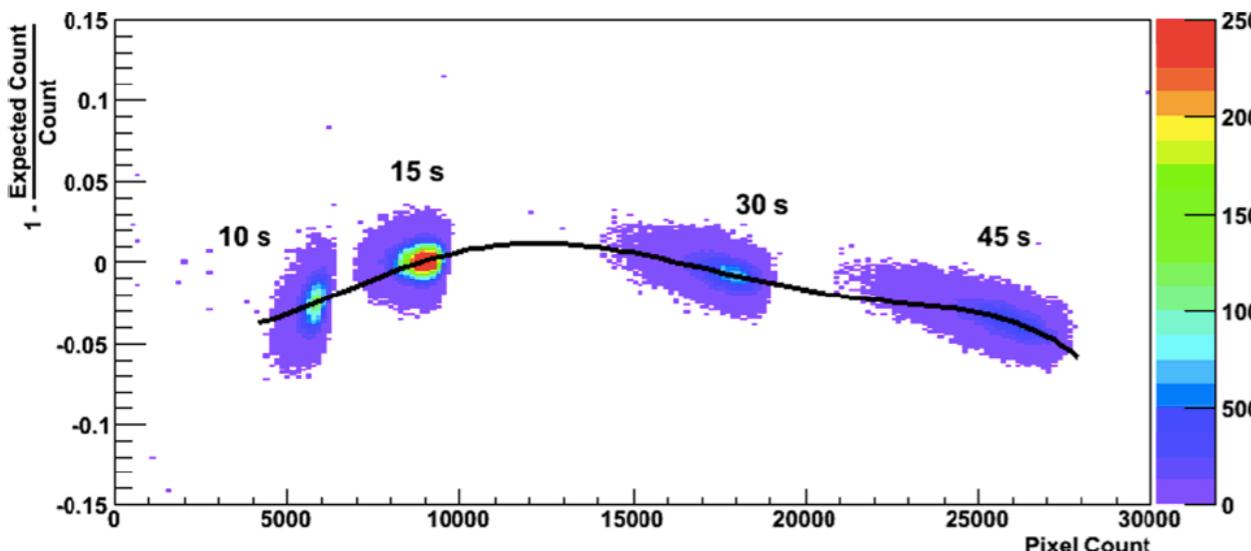
*need to be masked -
single exposure*



Artifacts

non-linearity: even before saturation level is reached, response becomes non-linear

can be measured from dome-flats



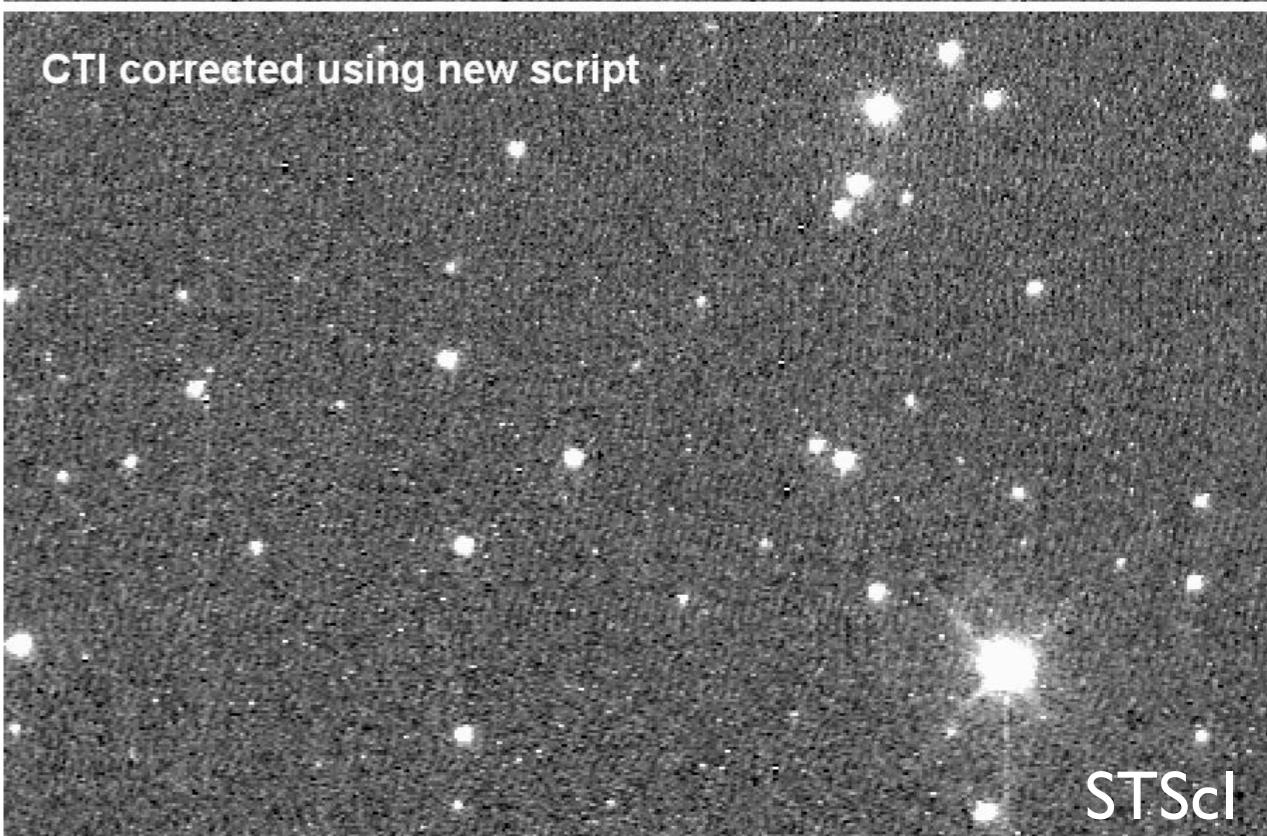
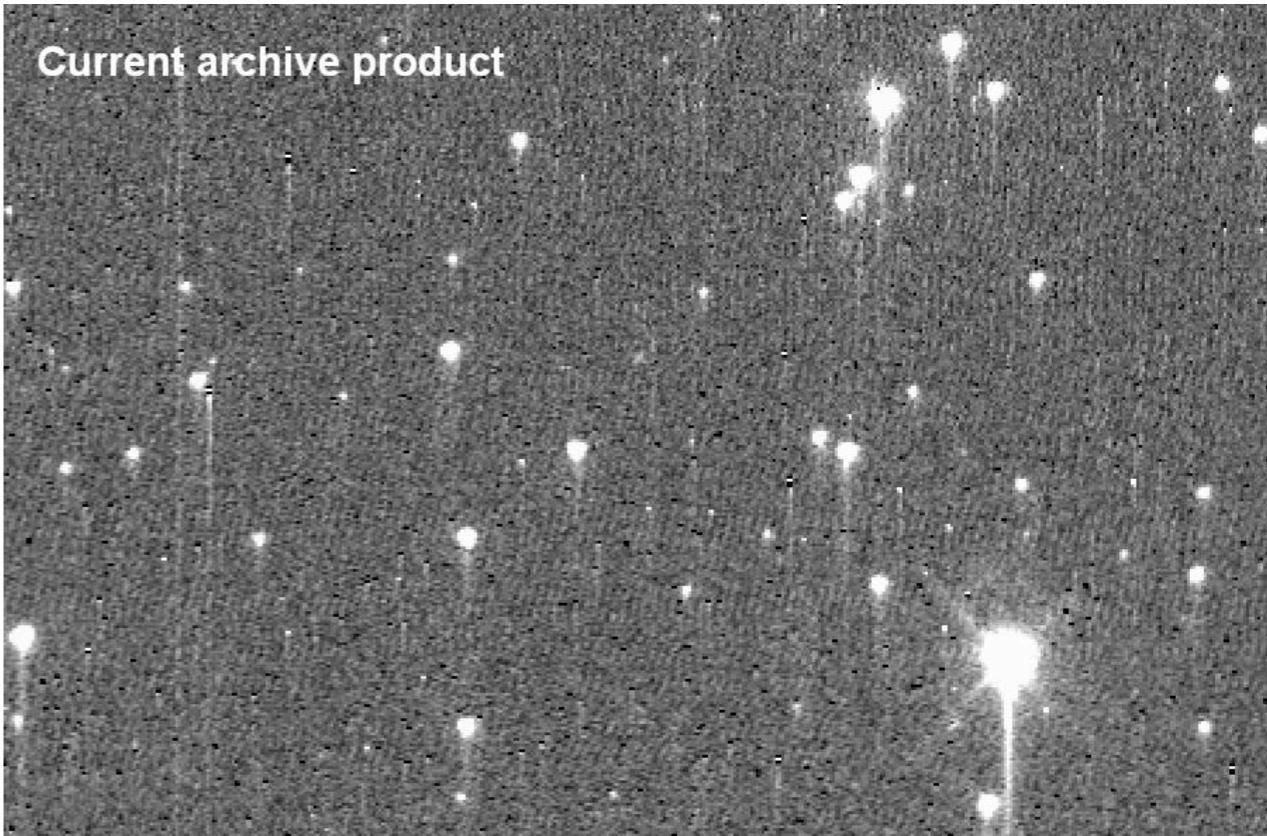
Artifacts

Charge Transfer Inefficiency (CTI): not all electrons are transferred from one pixel to the next during read-out

Charge Transfer Efficiency (CTE): fraction of photons that is transferred

CTI is a significant problem for Hubble's cameras because of radiation damage

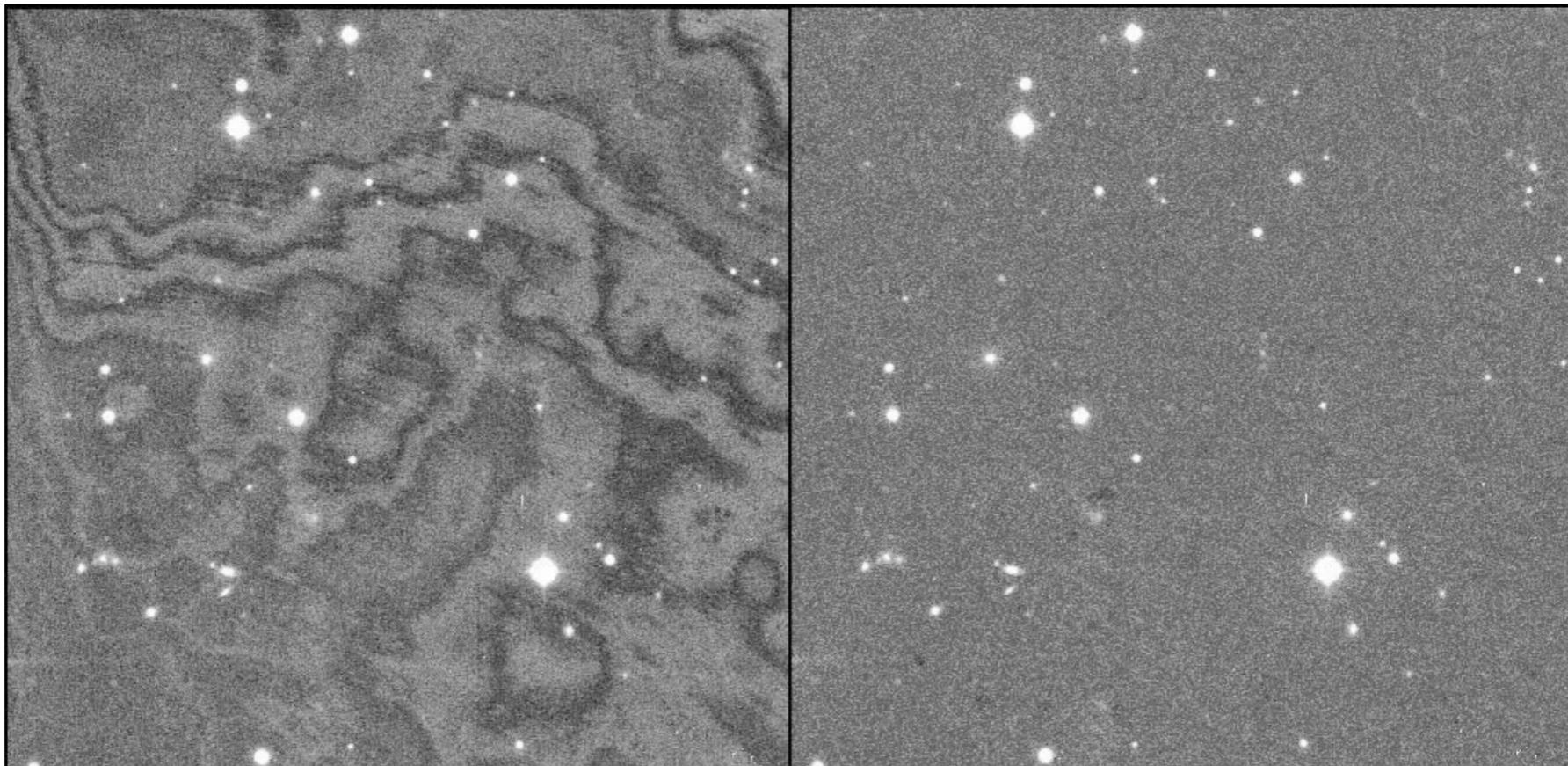
correction based on re-distributing charge



Artifacts

fringing: some light is reflected within the CCDs → leads to interference with incident light

fringing increases with wavelength, and decreases with thickness of CCDs



M. Schirmer

needs to be modeled; e.g. by subtracting a heavily smoothed image

FITS files

So you took all those images, now what?

Images

FITS: Flexible Image Transport System

- open standard for astronomical images
- (at least) two parts:
 - image (binary format, integer or float)
 - ASCII *header*
- can have multiple extensions (images)

Header Keywords

Mandatory Structure:

Table 5.1: Mandatory
keywords for primary header.

- 1 `SIMPLE`
- 2 `BITPIX`
- 3 `NAXIS`
- 4 `NAXISn, n = 1, ..., NAXIS`

(other keywords)

last `END`

Conforms to standard:
`T(rue) / F(alse)`

Bits per pixel:
16: integer
32: float

Number of axes:
2d image: `NAXIS=2`

Image dimensions

End of header

Example

Images from our CCD camera:

```
====> file m13.00000077.FIT (main) <====  
SIMPLE = T/CCDSOFT-SOFTWARE BISQUE 3  
BITPIX = 16  
NAXIS = 2  
NAXIS1 = 1024  
NAXIS2 = 1024  
BSCALE = +1.00000000000E+000  
BZERO = +3.27680000000E+004  
BIAS = 100  
FOCALLEN= +3.55600000000E+003  
OPTDREF = +0.00000000000E+000
```

How big is the image (pixels by pixels)?

Example

Images from our CCD camera:

```
HFTHREH = +0.00000000000E+000
APTDIA = +3.560000000000E+002
TELESCOP= 'Meade LX200'
UBSERVER= 'T. Cohen, B. Schultz, X. Liu, B. Baserdem'
DATE-OBS= '2016-08-30T03:11:28.477'
TIME-OBS= '03:11:28.477'
SWCREATE= 'CCDSoft Version 5.00.210'
SET-TEMP= -5.00000000000E+000
COLORCCD= 0
DISPCOLR= 1
IMAGETYP= 'Light Frame'
CCDFPFT = 1

UBSERVER= SBIGLEXI version 1.0
FILTER = 'Visual'
EXPTIME = +1.00000000000E+001
EXPOSURE= +1.00000000000E+001
LW_JPNL= 200
CCD-TEMP= -5.232156845990E+000
TEMPERAT= -5.232156845990E+000
INSTRUME= 'SBIG STL-1001 3 CCD Camera'
EGAIN = +2.060000000000E+000
F-GAIN = +2.060000000000F+000
ICK = 342
TE = 447
```

Specifying coordinates

The astrometric information in FITS images (also referred to as the WCS) is stored in the header using a standard set of keywords. The reference location is defined by the following keywords:

Specifying coordinates

The astrometric information in FITS images (also referred to as the WCS) is stored in the header using a standard set of keywords. The reference location is defined by the following keywords:

- CRVAL1: defines the right (α) ascension of the reference pixel
- CRVAL2: defines the declination (δ) of the reference pixel
- CRPIX1: the x location of the reference pixel
- CRPIX2: the y location of the reference pixel

Specifying coordinates

The astrometric information in FITS images (also referred to as the WCS) is stored in the header using a standard set of keywords. The reference location is defined by the following keywords:

- CRVAL1: defines the right (α) ascension of the reference pixel
- CRVAL2: defines the declination (δ) of the reference pixel
- CRPIX1: the x location of the reference pixel
- CRPIX2: the y location of the reference pixel

The plate scale and rotation of the image is contained in the CD MATRIX (CD?_? keywords).

- CD1_1 is the partial of first axis coordinate w.r.t. x
- CD1_2 is the partial of first axis coordinate w.r.t. y
- CD2_1 is the partial of second axis coordinate w.r.t. x
- CD2_2 is the partial of second axis coordinate w.r.t. y

$$\begin{pmatrix} CD1_1 & CD1_2 \\ CD2_1 & CD2_2 \end{pmatrix} = scale * \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$$

Specifying coordinates

$$\begin{pmatrix} CD1_1 & CD1_2 \\ CD2_1 & CD2_2 \end{pmatrix} = scale * \begin{pmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{pmatrix}$$

Thus, to go from image coordinates (x,y) to sky coordinates (α, δ):

$$\begin{pmatrix} \alpha - CRVAL1 \\ \delta - CRVAL2 \end{pmatrix} = \begin{pmatrix} CD1_1 & CD1_2 \\ CD2_1 & CD2_2 \end{pmatrix} \begin{pmatrix} x - CRPIX1 \\ y - CRPIX2 \end{pmatrix}$$

Specifying coordinates

After astrometric calibration:

```
CTYPE1  = 'RA---TAN-SIP' / TAN (gnomic) projection + SIP distortions
CTYPE2  = 'DEC--TAN-SIP' / TAN (gnomic) projection + SIP distortions
EQUINOX =           2000.0 / Equatorial coordinates definition (yr)
LONPOLE =           180.0 / no comment
LATPOLE =            0.0 / no comment
CRVAL1  =      250.418630769 / RA of reference point
CRVAL2  =      36.5118440685 / DEC of reference point
CRPIX1  =      351.470682144 / X reference pixel
CRPIX2  =      386.277894974 / Y reference pixel
CUNIT1  = 'deg'      / X pixel scale units
CUNIT2  = 'deg'      / Y pixel scale units
CD1_1   = -3.33986320359E-05 / Transformation matrix
CD1_2   =  0.000411933007076 / no comment
CD2_1   = -0.000411849476697 / no comment
CD2_2   = -3.33825508905E-05 / no comment
```

Our camera does not know where the telescope is pointing
- astrometric calibration is important e.g. for exoplanet lab

Viewing FITS images

best done with specialized software

e.g. ds9 (by
Smithsonian
Observatory)

<http://ds9.si.edu>

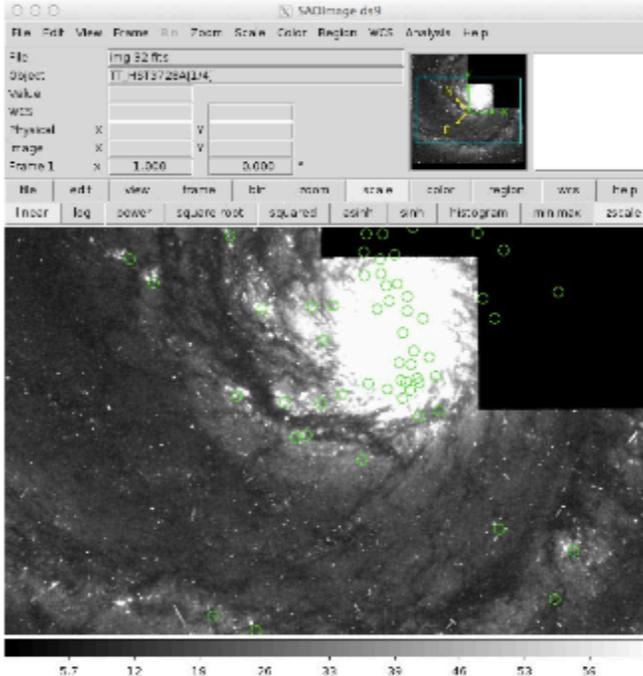
SAOImage DS9

Home | What's New | Download | Documentation | Gallery

[Tweet](#) [Email](#)

SAOImage DS9 Version 7.4

DS9 version 7.4 is now available on the [Download](#) page. New to version 7.4 is image blocking and reordering of data cube axes. Please see the [What's New](#) page for more details. *News Flash-- Version 7.5b4 is now available*



SAOImage DS9 development has been made possible by funding from the Chandra X-ray Science Center (CXC) and the High Energy Astrophysics Science Archive Center (HEASARC). Additional funding was provided by the JWST Mission office at Space Telescope Science Institute to improve capabilities for 3-D data visualization.

[Home](#) | [What's New](#) | [Download](#) | [Documentation](#) | [Gallery](#)

[SAOImage DS9](#) [@SAOImageDS9](#)

SAOImage DS9 version 7.5b4 is now available for download at ds9.si.edu/site/Beta.html

15 Jul

[SAOImage DS9](#) [@SAOImageDS9](#)

SAOImage DS9 version 7.5b3 is now available for download at ds9.si.edu/site/Beta.html. New support for Simple Image Access protocol

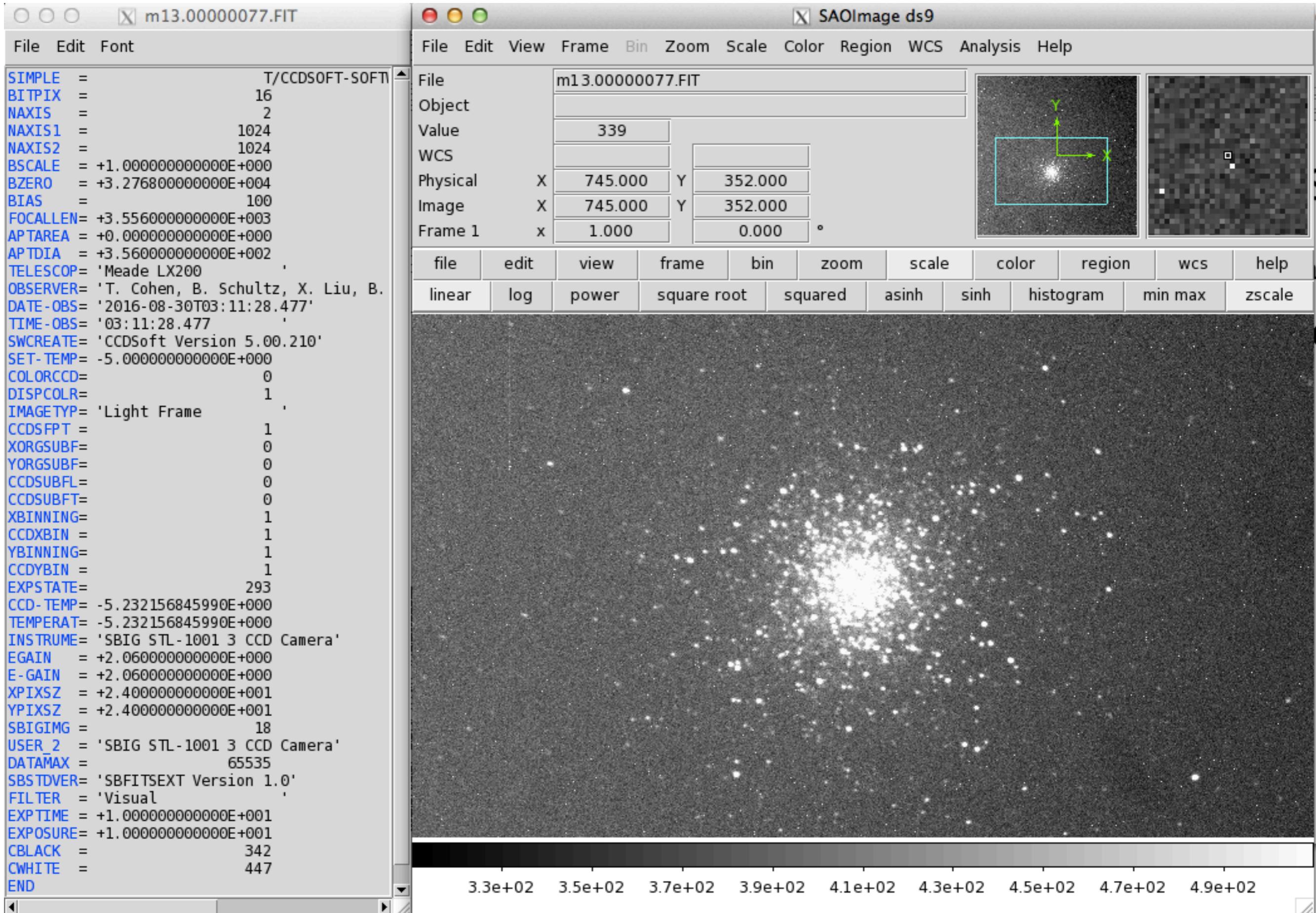
14 Jul

[SAOImage DS9 Retweeted](#) [Eric Mandel](#) [@astrosoftware](#)

[Embed](#) [View on Twitter](#)



SMITHSONIAN ASTROPHYSICAL OBSERVATORY | 60 GARDEN STREET | CAMBRIDGE, MA 02138



Viewing FITS headers

- ds9 (File -> Display Header)
- python (see tutorial), R, C, matlab, ...
- command-line tools: dfits and fitsort

```
[anja@ki-ls08 test_data]$ dfits 00000001.BIAS.FIT | more
====> file 00000001.BIAS.FIT (main) <====
SIMPLE = T/CCDSOFT-SOFTWARE BISQUE 3
BITPIX = 16
NAXIS = 2
NAXIS1 = 1024
NAXIS2 = 1024
PCRSIZE = +1.00000000000E+000
```

```
[anja@ki-ls08 test_data]$ dfits *.DARK.FIT | fitsort EXPTIME CCD-TEMP
FILE          EXPTIME          CCD-TEMP
00000011.DARK.FIT +1.00000000000E+001 -5.232156845990E+000
00000012.DARK.FIT +2.00000000000E+001 -5.232156845990E+000
00000013.DARK.FIT +4.00000000000E+001 -4.817803680962E+000
00000014.DARK.FIT +8.00000000000E+001 -4.817803680962E+000
00000015 DARK FIT +1.00000000000E+002 -4.817803680962E+000
```

Viewing FITS headers

- if dfits and fitsort not available: use “fold” command

```
[anja@ki-1s08 test_data]$ fold 00000001.BIAS.FIT | more
SIMPLE = T/CCDSOFT-SOFTWARE BISQUE 3
BITPIX = 16
NAXIS = 2
NAXIS1 = 1024
NAXIS2 = 1024
-----
```

FITS image manipulation / math

- python: FITS file handling part of astropy (see tutorial)
- R: FITSio package
- matlab: fitsread, etc.
- C: cfitsio library

command line:

- FTOOLS, e.g.:

```
ftpixcalc out.fits "(a-b)" a=img1.fits b=img2.fits
```

Python Tutorials

← → ⌂ 🔒 github.com/anjavdl/PHY517_AST443/wiki/Python



There are a number of ways of working with python, in particular:

1. interactively on the command line (shell mode)
2. with python programs / scripts
3. through jupyter notebooks
4. through [Google Colab](#) notebooks

Shell mode

In shell mode, you simply type the command `python` or `python3` into a terminal, which will print some details about the version of python you have installed, the date, and the commands for how to get help, credits, and licensing information. The terminal will show `>>>`, which means you can now type in python commands and the interpreter will execute those commands. Note that on the Computing Lab machines, `python` invokes python2.7, so you need to call `python3` to get python3.

Python programs / script

The second way to use python is in script mode. You write a text file containing all of the python commands you want python to execute, in the order that you want them done. After you save this file, which is called a script, you type `python` into a terminal with your script filename as the first argument, and the python interpreter reads the script and executes it line by line. An example is the [rdj2aa.py](#) script that you can use to convert a list of RA, Dec, JD to azimuth, altitude, and UT date. In this case it takes two additional filenames for the input and output files:

```
python rdj2aa.py in.txt out.txt
```

Jupyter notebooks

[Jupyter notebooks](#) are a great way to keep code organized and documented. Jupyter can be used with several kernels, including python. Please see [here](#) for further instructions specific to this class.

Labs and Write-Ups

- [Guidelines](#)
- [How to write a decent lab report](#)
- [Observing Equipment](#)
- [Observing Calendar](#)
- [Lab 1: CCDs](#)
- [Lab 2: Exoplanet transit](#)
- [Lab 3: Diffuse Nebula Spectroscopy](#)
- [Lab 4: Your own proposal](#)
- [Discontinued: Radio Interferometry](#)
- [Weather](#)
- [End-of-night report](#)

Computing

- [Computing Resources](#)
- [Astro Software Overview](#)
- [Bash](#)
- [awk and sed](#)
- [LaTeX](#)
- [Python](#)
- [jupyter](#)
- [GitHub](#)
- [ds9](#)
- [SExtractor](#)
- [Topcat](#)
- [Astrometry.net](#)
- [dfits and fitsort](#)
- [Image arithmetic \(+ftools\)](#)

Python Tutorials

Jupyter notebooks

[Jupyter notebooks](#) are a great way to keep code organized and documented. Jupyter can be used with several kernels, including python. Please see [here](#) for further instructions specific to this class.

Google Colab

[Google Colab](#) lets you run python code (in notebook format) on Google's computers. This means that there is no need for you to install python on your own computer. It also makes it easy to share your notebook with others through Google Drive and/or github, and thus to work collaboratively on code. Our experience shows that if you need a particular python package for your project, it is much easier to ``install'' it on Google Colab than your own computer or the Computing Lab.

Python tutorials for this class

[General python tutorial](#)

Data file for python tutorial: [test_data.txt](#)

[Python notebook](#) on reading FITS files, numpy, plotting histograms.

Fits files: [00000025.BIAS.FIT](#) , [00000026.BIAS.FIT](#)

Older tutorials in python 2.7: [Python primer](#), [notebook](#)

Installing python on your laptop

Python is available for all common operating systems. The easiest way to install it is through [anaconda](#). Anaconda includes all the packages needed for this lab, in particular numpy and astropy. Anaconda does take up significant disk space, so instead you could install miniconda along with the packages you need.

Other python resources

Excellent tutorials and resources are available on the webpage for [Prof. Zingale's python class](#).

Python Tutorials

tutorials are in jupyter notebook format

on your laptop:

- if you installed python through anaconda on your laptop, jupyter is included
- python is not installed on your laptop? try google colab

on the machines in the computing lab:

- you can run `python3` in a shell / as a script
- uhura / vulcan have jupyter, but only with `python2.7`