Draft version October 15, 2018
Typeset using LaTeX **preprint** style in AASTeX62

Lab 1: Photon Counting & Detectors

Jessica Birky (A13002163)

ABSTRACT

This example manuscript is intended to serve as a tutorial and template for authors to use when writing their own AAS Journal articles. The manuscript includes a history of and documents the new features in the previous versions as well as the new features in version 6.2. This manuscript includes many figure and table examples to illustrate these new features. Information on features not explicitly mentioned in the article can be viewed in the manuscript comments or more extensive online documentation. Authors are welcome replace the text, tables, figures, and bibliography with their own and submit the resulting manuscript to the AAS Journals peer review system. The first lesson in the tutorial is to remind authors that the AAS Journals, the Astrophysical Journal (ApJ), the Astrophysical Journal Letters (ApJL), and Astronomical Journal (AJ), all have a 250 word limit for the abstract. If you exceed this length the Editorial office will ask you to shorten it.

1. INTRODUCTION

In order to understand the. Advancements in technology in the 20th century has now made it possible to observe electromagnetic radiation from the scales of $10^{-12} - 10^{3}$m (as short as gamma rays to as long as radio waves). Understanding how detectors work, the statistics of light collection, as well as the limitations of the instrument and sources of contaminations are a fundamental first steps before performing any scientific analysis with an instrument.

CCDs, the photoelectric effect

Types of noise we want to quantify

2. OBSERVATIONS

Nickel telescope, what are bias and flat frames

3. DATA REDUCTION & METHODS

3.1. *Theoretical Distribution of Light*

Before we can characterize our observations, it is important to first determine what we expect our data will look like. That is, we are measuring the number of photons hitting a detector over a fixed amount of time, what do we predict will be the number of photons that will hit each pixel in that tme interval? Let's assume that on average μ photons will hit a pixel over fixed time interval t. Now consider dividing that time interval into n increments: the probability that a n photons arrive to the detector is approximately Binomial. Now in the limit that $n \to \infty$, we arrive at the Poisson distribution:

$$P(x; \mu) = \frac{\mu^x}{x!} e^{-\mu} \tag{1}$$

which we interpret as the probability that x photons will arrive to a detector pixel in time interval t, given that μ photons arrive in that interval on average. The expected mean and variation for this distribution being $\langle x \rangle = \mu$ and $\langle \sigma^2 \rangle = \mu$.

In practice, the Poisson distribution is unstable to compute for large values of x because the term $x!$ becomes too large for the memory of the computer to handle. In order to try to reduce the memory load for large values of x, we tried approximating the $x!$ term using the Sterling approximation which requires less floating point operations than calculating factorials directly:

$$n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n \tag{2}$$

We found that the Sterling approximation is advantageous for approximating the Poisson at continuous values of x (whereas the exact method for computing can only take discrete values of x for $x!$), however it does not make the solution any more numerically stable for values of x larger than ~ 15 (Figure 1).

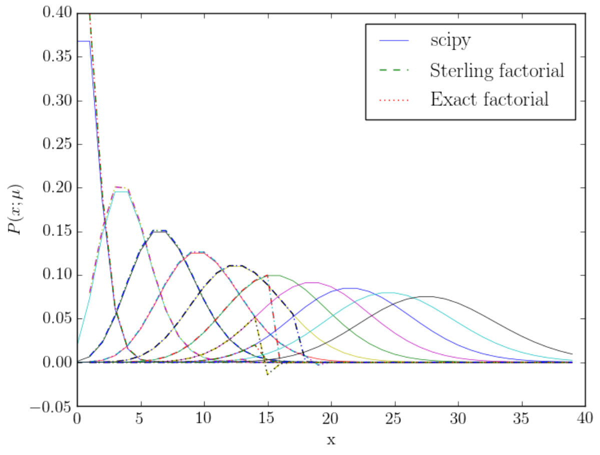

Figure 1. Comparison of different methods for computing the Poisson distribution.

3.2. *Data Manipulation*

Procedure to combine frames and subtract bias, point to code

3.3. *Types of Detector Noise*

How to calculate the readnoise, other noise, error propagation

4. DATA ANALYSIS & MODELING

4.1. *Bias Frames*

4.2. *Combined Images & Histograms*

The figures in 3 and 4 show the combined, bias-subtracted histograms and images for each of the eight different exposure times (3, 6, 12, 24, 48, 96, 192, 384, and 768 seconds). The left side of each panel shows the histogram of the number of detector counts in analog to digital units (ADU),

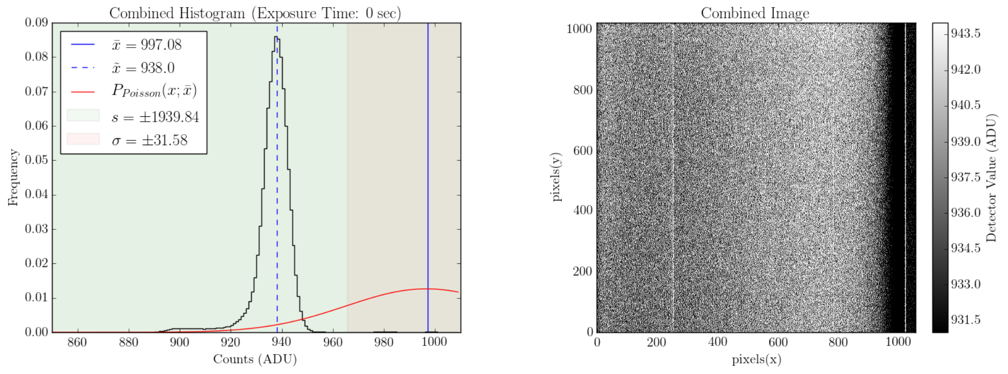

Figure 2. Combined bias frames.

normalized such that the sum of all of the bins is one (plotted in black). The solid blue vertical line marks the mean of the data, and the dashed blue line marks the median, indicating which distrubtions are skewed by outlying points (where the mean is far from the median): in the combined bias frame the mean is skewed higher than the center of the distribution, and for all of the other exposures the mean is skewed lower than the center. The two shades of green mark ± 1 and ± 2 standard deviations above and below the mean. For all distributions, the $\pm 1\sigma$ lines lie far outside of the bulk of the distribution around the median, which indicates again that there are many outlying the part of the distribution shown. Finally, the distribution in red shows the expected Poisson distribution, as a function of the mean of the data ($\bar{x}$), with a one standard deviation range also shaded in red (where the expected standard deviation for the Poisson is $\sigma = \sqrt{\bar{x}}$).

The right side of each panel shows the 2D images. In order to visualize the detector counts per pixel, we must map the number of counts to an RGB color value, which we use the matplolib 'gray' colormap for. To emphasize the features in the data, we choose the min an max values of the colormap to be the 10th and 90th percentile counts of the data.

4.3. *Quatifying Detector Noise*

Calculate readnoise n stuff

5. DISCUSSION

5.1. *Poisson Comparisons*

Looking at Figures 3 and 4, we find that the detector count distribution is close to a Poisson distribution for low exposure times, but for larger exposure times it increasingly diverges with a much larger standard deviations than expected (as summarized by Figure 5). We also see that low exposure times (3, 6, and 12 sec) have a single peak in the distribution around the mean, however at 48 sec the distribution becomes bimodal, and for high exposure times (96, 192, 384, and 768 sec) the distribution develops three separate peaks. Comparing these histograms to the corresponding images at the right, we can see the regions on the detector which correspond to the different peaks: the dark edges make up roughly the bottom third of the distribution, and the bright center making up the higher thrid of the distribution. If the distribution was Poissonian for each pixel, we would expect the signal to vary uniformly across the detector, but instead there is obvious contamination at different spatial regions.

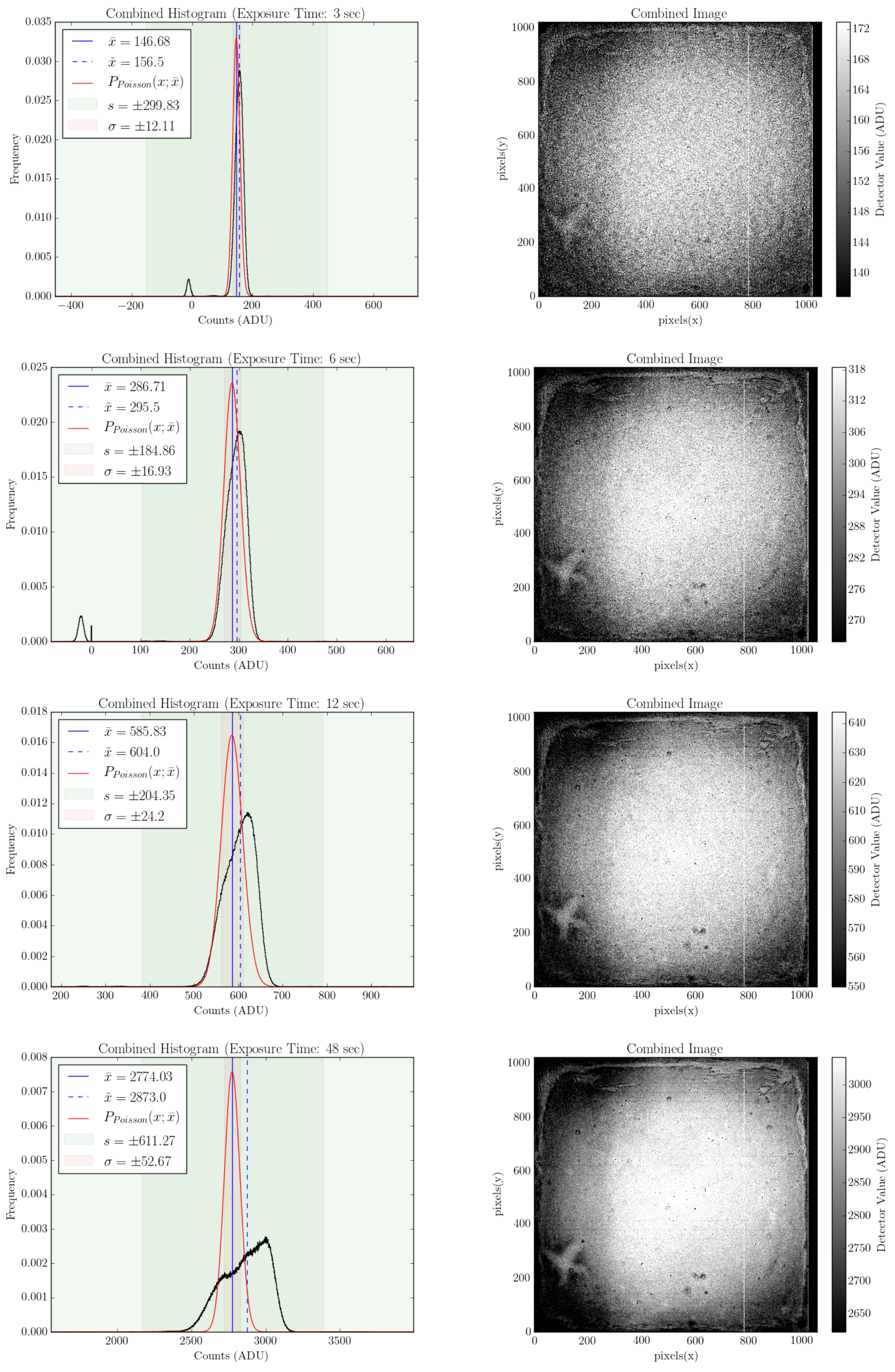

Figure 3. Combined, bias-subtracted, flat frame histograms and images for varying exposure times (0, 3, 6, and 12 sec).

5.2. *Sources of Noise & Systematic Effects*

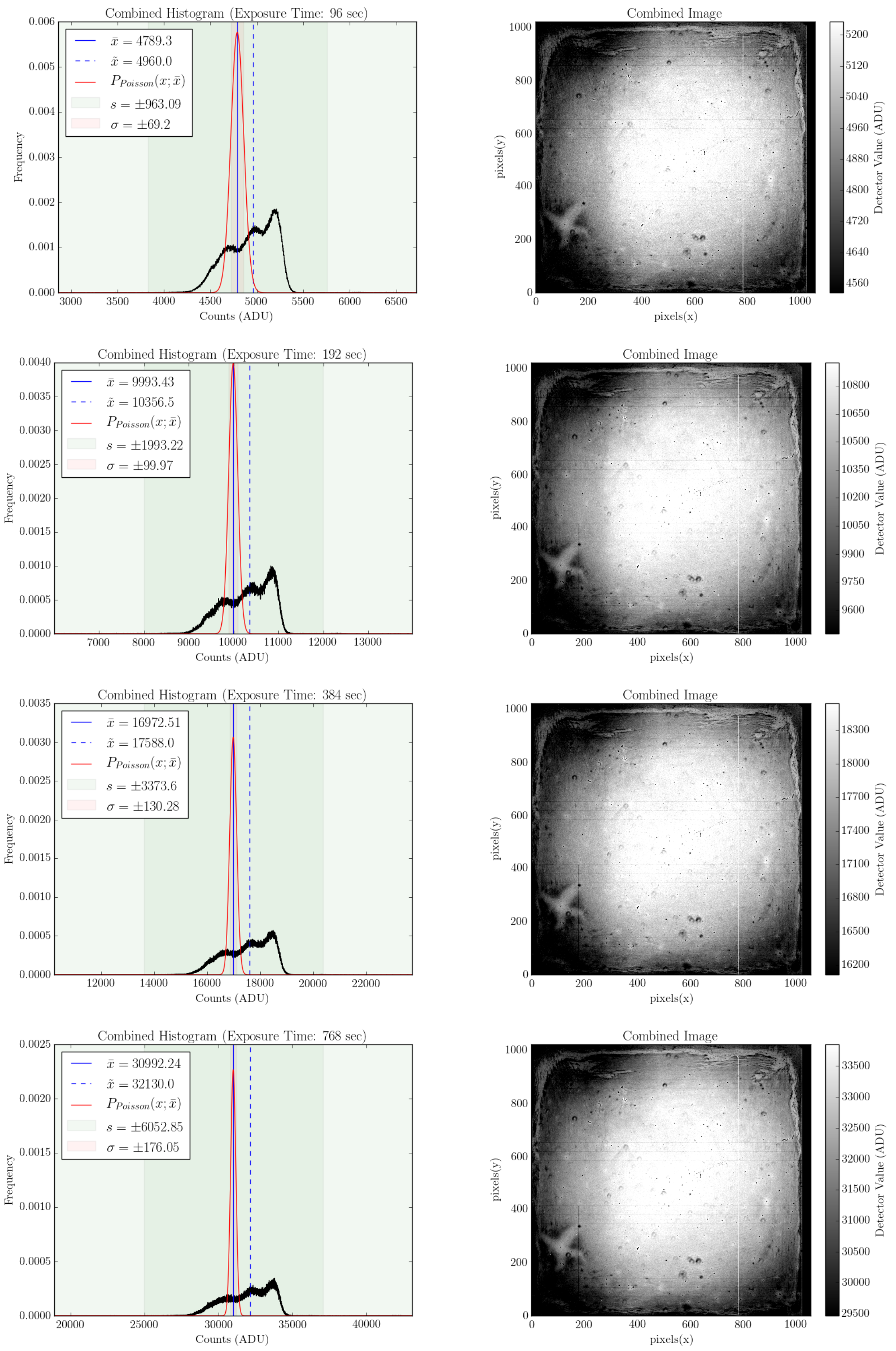

Figure 4. Combined, bias-subtracted, flat frame histograms and images for varying exposure times (96, 192, 384, and 768 sec).

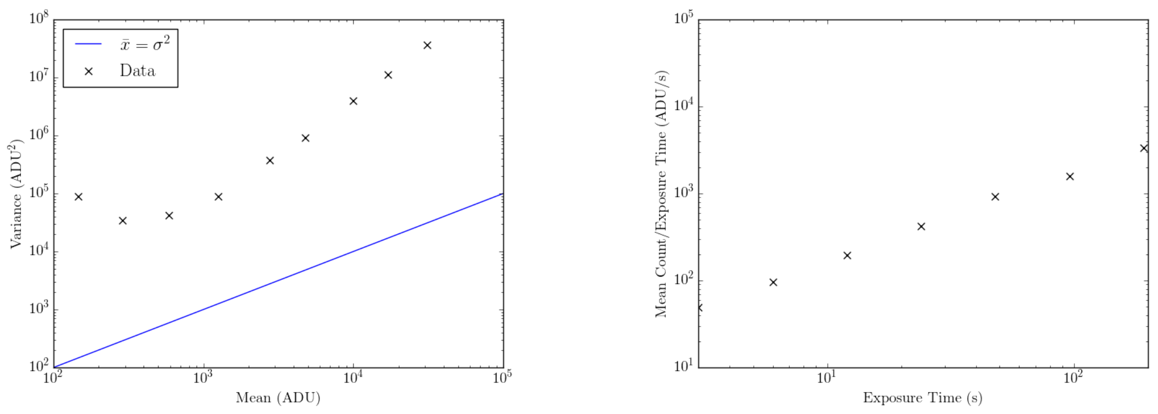

Figure 5. Mean vs. variance for all bias-subtracted exposures.

Variations in the quantum efficiency of each pixel. Dust particles on the lens scatter and absorb light unevenly at different points on the detector.

Checked for systematics due to temperature variation that could maybe be responsible for dark current: found temperature to vary only within 4C (Figure 6). Checked for systematic differences in different datasets.

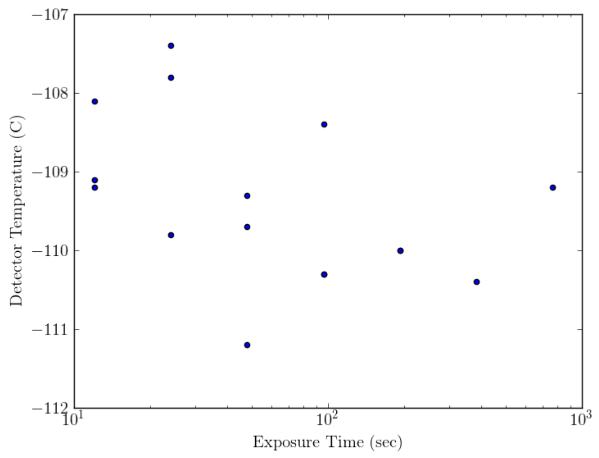

Figure 6. Temperature of the CCD detector for different exposure frames does not vary more than about 4C.

6. CONCLUSION

7. AUTHOR CONTRIBUTIONS

This project was done in collaboration with Julian Beas-Gonzalez and Russell Van-Linge (Group E), using data collected by Group B. The code written is my own (except for a few snippets for making histograms and reading fits files from the class notes), and can be found on github: https://github.com/jbirky/photonCounting.

8. APPENDIX

8.1. *Statistics*

Mean and variance for a set of data points $x = x_1, ..., x_N$:

$$\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i \tag{3}$$

$$s^2 = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2 \tag{4}$$

8.2. *Code*

Statistical Code

```python
def mean(array):
    return np.sum(array)/len(array)

def stdev(array):
    return np.sqrt(sum((array - mean(array))**2)/(len(array) - 1))

def factorial(n, **kwargs):

    method = kwargs.get('method', 'sterling')

    if method == 'exact':
        nfact = 1
        for i in range(n):
            nfact *= (i+1)
    elif method == 'sterling':
        nfact = math.sqrt(2*math.pi) * n**(n+.5) / math.e**n

    return nfact

def poisson_approx(xarray, mu, **kwargs):

    method = kwargs.get('method', 'sterling')

    pdist = np.array([mu**x/factorial(x, method=method) * math.e**(-mu) \
                for x in xarray])

    return pdist
```

Data Processing code

```python
def readData(folder):
    """
    Read all frames from a given directory into one matrix.
    input:  directory to folder containing frames
    output: array3d (dimension: xpixel x ypixel x # frames)
            array2D (dimension: 1D flattened img x # frames)
    """

    files = os.listdir(folder)

    array3D, array2D = [], []
    for ff in files:
        arr = fits.getdata(folder + ff)
        array3D.append(arr)
        array2D.append(arr.flatten())

    return np.array(array3D)

def combineFrame(data_array):
    """
    Combine image frames and return 1D array.
    input:  data array (dimension: # 1D detector counts x # frames)
    output: 1D combined array of detector counts (ADU)
    """

    return np.median(data_array, axis=0)
```

Histogram/Image code

```
def plotAll(array2D, **kwargs):

    arr = array2D.flatten()

    avg = mean(arr)
    std = stdev(arr)
    med = median(arr)
    Npix = len(arr)

    sigma = kwargs.get('sigma', 2)
    low = int(np.round((avg-sigma*std)))
    high = int(np.round((avg+sigma*std)))
    rng = kwargs.get('rng', [low, high])
    exp = kwargs.get('exp')

    fig, (ax1, ax2) = plt.subplots(1,2, figsize=[18,6])

    # Histogram
    #===========
    hr = np.arange(rng[0], rng[1])
    hist = []
    for i in hr:
        counts = len(np.where(arr==i)[0])
        hist.append(counts)
    #normalize histogram
    hist = np.array(hist)/sum(hist)
    ax1.step(hr, hist, color='k')

    #mean and median lines
    ax1.axvline(avg, color='b', label=r'$\bar{x}=%s$'%(np.round(avg,2)))
    ax1.axvline(med, color='b', label=r'$\tilde{x}=%s$'%(np.round(med,2)), linestyle='dashed')

    #sigma levels
    if kwargs.get('show_level', True) == True:
        for i in np.arange(1,sigma+1):
            if i == 1:
                ax1.axvspan(avg-i*std, avg+i*std, facecolor='g', alpha=0.05, label=r'$s=\pm␣%s
            else:
                ax1.axvspan(avg-i*std, avg+i*std, facecolor='g', alpha=0.05)


    #poisson distribution
    xarray = np.arange(rng[0], rng[1], 1)
    pdist = poisson.pmf(xarray, avg)
    ax1.plot(xarray, pdist, color='r', label=r'$P_{Poisson}(x;\bar{x})$')
    std_expected = math.sqrt(avg)
    ax1.axvspan(avg - std_expected, avg + std_expected, facecolor='r', alpha=0.05, \
                label=r'$\sigma=\pm␣%s$'%(np.round(std_expected,2)))

    ax1.legend(loc='upper␣left')
    ax1.set_xlabel('Counts␣(ADU)')
    ax1.set_ylabel('Frequency')

    if 'exp' in kwargs:
        ax1.set_title('Combined␣Histogram␣(Exposure␣Time:␣%s␣sec)'%(exp))
    ax1.set_xlim(rng)

    # Image
    #===========
    hrng = kwargs.get('hrng', [np.percentile(arr, 10), np.percentile(arr, 90)])
    pl = ax2.imshow(array2D, origin='lower', interpolation='nearest', cmap='gray', vmin=hrng[0
    fig.colorbar(pl, ax=ax2, fraction=0.046, pad=0.04).set_label('Detector␣Value␣(ADU)')

    ax2.set_xlabel('pixels(x)')
    ax2.set_ylabel('pixels(y)')
    ax2.set_title('Combined␣Image')

    if 'save_dir' in kwargs:
        save_dir = kwargs.get('save_dir')
        plt.savefig(save_dir + 'exposure%s.png'%(exp))
    plt.show()
```