Krittika Summer Project
Study of SN2023ixf
Midterm Report

Jigyasa Chouhan

June, 2024

Contents

1 Charge Coupled Device(CCD) Detectors

A CCD or Charge Coupled Device is a highly sensitive photon detector. It uses semiconductors, mostly silicon based semiconductors, to detect light based on the photoelectric principle. The valence electrons get excited to the conduction band by absorbing photons in these detectors.

These detectors are placed in the focal plane of the telescope to record photons incident on it. Most astronomical objects are too faint that we can not afford to lose any electron so these detectors are made very sensitive using a 2D array of pixels for imaging.

Quantum Efficiency of a detector is the fraction of photons that produce photoelectrons. The Quantum Efficiency of CCD's reach up to high values like 90% for certain wavelengths. Once the pixel array of photoelectrons is ready, CCD uses charge transfer to read these. Each pixel consists of three electrodes. A large positive voltage applied to one will create a potential well which is filled with photo-electrons The voltage in an adjacent electrode is raised to the same level – allowing the electrons to flow. Decreasing the voltage of the original one completes the transfer.

The voltage of each charge packet is amplified and measured The measured voltage is digitized using analogue-to-digital converter (ADC), producing analogue-to-digital units (ADUs) (or counts) which are then stored. Gain is number of electrons/ADU – kept close to 1. Each pixel has a maximum charge carrying capacity – go beyond that and electrons spill into neighbouring pixels (called blooming). This way there is a continous reading of electrons and thus photons making CCD's a very efficient way of capturing photons from faint astronomical sources.

2 Image Reduction: Theory

A raw image is not very accurate and has some noise inevitably. There are a few techniques that can be used to improve any image. Signal-to-noise ratio (SNR) is a measure used in science and engineering that compares the level of a desired signal to the level of background noise. It is defined as the ratio of signal power to the noise power, often expressed in decibels. The following are some methods to remove noise and increase SNR.

2.1 Bias Subtraction

The raw images that a detector captures usually have counts even at unexposed regions. It is important to subtract these unnecessary counts from the image based on the sensitivity of each pixel. There is a series of zero exposure images taken. These values are called bias. Then a master bias is created by taking the median of all the bias captured. This master bias is then subtracted from the raw images taken by the detector.

2.2 Flat Fielding

We take into consideration the fact that each pixel of the 2D array has different sensitivity since all hardware are inherently unique and it's almost impossible to create exactly identical pixels. Vignetting, dust particles, and some other manufacturing artifacts may also result in varying sensitivities. So to normalize the raw image uniformly according to the sensitivity of each pixel. For this, we point the telescope to a region in sky which is known to be uniform. Typically twilight is used for this purpose. We take a series of flat field images in the same filter that is used for observation and subtract the bias from them. There may be varying levels of intensity between images, so we normalize the flats to have a median of 1. The median of the flats are again found to produce a master flat. Each bias subtracted raw image is then divided by the master flat to calibrate the sensitivity correctly. Finally, this gives us a usable science image.

3 Image Reduction: Code

We have the 30 raw images of the supernova SN2023ixf taken at different times along with some flats and biases. We are going to reduce these images using python to improve their SNR. Following is a generic description of how to go implement image reduction using python.
In the image reduction task, we used the following libraries and packages in python to implement bias subtraction and flat fielding:

- python 3
- astropy
- numpy
- matplotlib
- photutils
- pyregion
- SExtractor
- SWarp

To reduce a raw science image, we require a number of bias files and flat files. Images in astrophotography are stored as FITS files. Astropy aids us to deal with fits files in python. We will first have to define the directory structure to store the list of raw image files, bias files and flat files.

`from astropy.io import fits` this structure helps us to open fits file using
`exampleFile = sciList[0]`
`HDUList = fits.open(exampleFile)`
`HDUList.info()`
This will print some info about the opened fits file. In a similar way we can access all of the fits properties from `HDUList`.
This snippet of code will display the selected image for you:

```
## Get and display the image data

data = HDUList[1].data  # Get the data array (a simple numpy array) from the first extension.

mean, median, std = sigma_clipped_stats(data) # get some image statistics
plt.figure(figsize=(10,10))                     # set up the plot panel
plt.imshow(data, vmin = median - 2*std, vmax = median + 20*std, origin='lower')
plt.colorbar()
plt.show()
```

Once we know these basic things, we proceed to reducing the image using bias and flats. First of all, we need to find the median of the bias files. In the above mentioned way, we open all the bias fits files and store them in numpy arrays. Using the np.median function using the appropriate axis, we create a master bias array. This master bias array is then subtracted from the raw images as well as flats for further development.

```
## Median-combine the bias frames to make a master bias

# Create a 3D array to store all the bias files together.
ny = 2501
nx = 2148
numBiasFiles = len(biasList)
biasImages = np.zeros((ny, nx, numBiasFiles))

# Add the files to the array
for i in range(numBiasFiles):
        HDUList = fits.open(biasList[i])     # Open the file
        biasImages[:,:,i] = HDUList[1].data  # Load the data into the appropriate layer
        HDUList.close()                      # Close the file

# Create the master bias frame by doing a median combination for each pixel over all layers
masterBias = np.median(biasImages, axis=2)
```

We can make sure the master bias is correct by displaying it on the screen.

Next we will have to create a master flat in the same way as we did with bias. But before taking the median of the flats, we need to subtract the bias as well from the flat images. After having bias subtracted flat images, we shall also divide each pixel of a flat file with its normalising factor to bring all pixel values between 0 and 1.

```
## Load and normalize the flat fields in preparation for making a master flat field.

# Set up the 3D array
numFlatFiles = len(flatList)
flatImages = np.zeros((ny, nx, numFlatFiles))

# Load the files into the array, with bias subtraction and normalization
for i in range(numFlatFiles):
        # Load the data from the fits file
        HDUList = fits.open(flatList[i])
        data = HDUList[1].data * 1. # Convert to floating point
        HDUList.close()

        # Bias-subtract, normalize, and add to the array layer
        data -= masterBias
        normfactor = np.median(data[500:1600,500:1600])
        print(normfactor)
        flatImages[:,:,i] = data / normfactor
```

After getting the master flat, we shall also flag the unexposed pixels in the image to remove redundant data. So we assign the value 'NaN' to the pixels that are unexposed.

Finally, we will now process the science frames. In the science frames, we just subtract the master bias from all the values and divide it by the master flat as shown.

```python
## Bias subtract and flat-field all science frames and write out pre-processed files.

numSciFiles = len(sciList)
print('Found %d science files'%numSciFiles)

for i in range(numSciFiles):
        # Read in the FITS data.
        HDUList = fits.open(sciList[i])
        primaryHeader = HDUList[0].header
        imageData = HDUList[1].data
        HDUList.close()

        # Correct for the bias and flats here
        procData = (imageData - masterBias) / masterFlatFixed

        # Prepare the output FITS structure in simple format
        procHDU = fits.PrimaryHDU(procData)  # Create a new HDU with the processed image data
        procHDU.header = primaryHeader       # Copy over the header from the raw file
        procHDU.header.add_history('Bias corrected and flat-fielded') # Add a note to the header

        # Write the reduced frame to disk
        print(sciList[i],'->',procList[i])
        procHDU.writeto(procList[i], overwrite=True)
```

Now we have the reduced usable science images in a folder named processing. Now we will proceed to use these images to study the objects in the image.

4 Astrometry

Astrometry is a branch of astronomy that involves precise measurements of the positions and movements of stars and other celestial bodies. So to perform astrometry on our image basically just means to analyse which part of the sky has been captured in the image and align all the images taken according to their respective offsets to precisely study any one object that we want to focus on, in our case, SN2023ixf.
The website "astrometry.net" is an online facility to easily perform the astrometry by just uploading our reduced science images on the website and download the results then and there. This gives us data about the exact location of where the telescope is pointing i.e. Right Ascension ans Declination of the spot in the sky. The website uses ready made database to find this out using the background of stars visible in the image. It will also help us to find the offset of each image taken on various days to help us focus on the object we are actually studying.

5 PSF photometry: Theory

Photometry is a system of language, mathematical formulations, and instrumental methodologies used to describe and measure the propagation of light through space and materials. In consequence, the radiation so studied is confined to the visible (VIS) portion of the spectrum. Only light is visible radiation. The point spread function (PSF) photometry is used to describe the response of a focused optical imaging system to a point source or point object. It is able to account for crowding but is computationally expensive.

5.1 Introduction to PSF

Point Spread Function (PSF): The PSF describes how a point source of light (such as a star) appears on the detector due to the combined effects of telescope optics, the atmosphere (in ground-based observations), and the detector itself. Ideally, it would be a delta function, but in practice, it is a broader distribution. PSF Characteristics: The PSF shape can vary across the field of view due to optical aberrations and atmospheric distortions. It typically has a core and wings, with the core being the most concentrated part of the light distribution.

5.2 Importance of PSF Photometry

Crowded Fields: In star clusters or dense regions of the sky, stars can overlap significantly. PSF photometry enables accurate measurement of individual star brightness in these crowded fields. Precision: By modeling the PSF, astronomers can achieve more precise photometric measurements than with aperture photometry, especially when dealing with faint or closely spaced objects.

5.3 Steps in PSF Photometry

- PSF Estimation
  PSF Model Creation: The first step is to create a PSF model, typically using bright, isolated stars in the image. The PSF can be modeled using analytical functions (e.g., Gaussian, Moffat) or empirical data.
  Spatial Variation: If the PSF varies across the field, a spatially varying PSF model can be constructed to account for changes in the PSF shape over the image.

- PSF Fitting
  Initial Source Detection: Stars are detected using algorithms such as thresholding or wavelet transforms. Initial positions and rough estimates of their brightness are determined.
  PSF Fitting Algorithm: The PSF model is fitted to each detected star by adjusting the model parameters (position, flux, etc.) to minimize the difference between the model and the observed data. Common fitting methods include least squares and maximum likelihood estimation.

- Crowded Field Deconvolution
  Simultaneous Fitting: In crowded fields, multiple stars are fitted simultaneously to account for overlapping PSFs. Iterative methods are often used to refine the fits.
  Residual Analysis: After fitting, the residual image (original image minus fitted stars) is analyzed to detect any remaining stars or assess the quality of the fit.

- Flux Measurement
  Flux Calculation: The total flux of each star is derived from the PSF fit parameters. This flux is proportional to the star's brightness.
  Error Estimation: Uncertainties in the measurements are estimated, considering noise, fitting errors, and residuals.

6 PSF photometry: Code

Photutils provides a modular set of tools to perform PSF photometry for different science cases. The tools are implemented as classes that perform various subtasks of PSF photometry. High-level classes are also provided to connect these pieces together.

The PSF photometry python notebook attached shows how the code uses modules like astropy and photoutils to perform the photometry on our reduced images of the sky and gives us the magnitude of the supernova in different images on different days.

7 Aperture photometry

Aperture photometry is a fundamental technique used in the analysis of astronomical images, particularly for measuring the brightness of point-like sources such as stars. This method involves defining a circular aperture around the object of interest and summing the pixel values within this aperture to estimate the total flux from the object.

7.1 Aperture photometry tool

The Aperture Photometry Tool (APT) is a widely-used software application designed for perform- ing aperture photometry on astronomical images. APT offers a user-friendly interface and a range of features that make it a valuable tool for astronomers.

- User Interface: APT provides an intuitive graphical user interface (GUI) that allows users to visually select apertures and background regions directly on the image. This makes the process of performing photometry straightforward and accessible, even for those with limited programming experience.

- Customization: Users can customize the size and shape of the apertures, as well as the background regions, to suit their specific needs. APT also supports batch processing, en- abling photometry to be performed on multiple images simultaneously.
- Output: The tool generates detailed reports that include the measured fluxes, uncertainties, and other relevant parameters. These reports can be easily exported for further analysis or integration into larger datasets.
- Integration with Other Software: APT can be used in conjunction with other astro- nomical software, making it a flexible tool that can be integrated into various workflows.

7.2 Photoutils: SkyCircularAperture

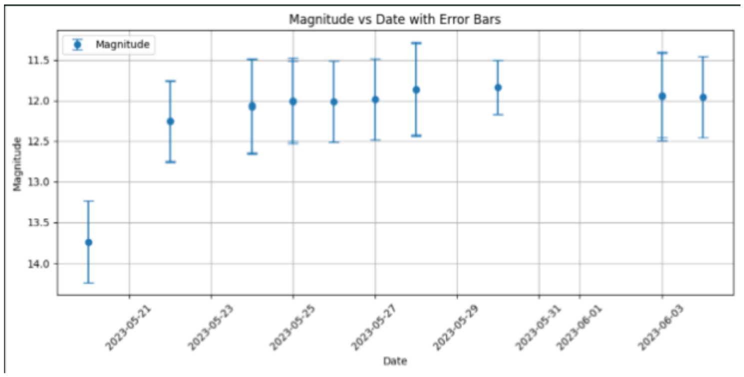
Photutils is a Python library that provides tools for performing photometry and other image analysis tasks in astronomy. One of its key features is the SkyCircularAperture class, which is used for defining and applying circular apertures in celestial coordinates.

- SkyCircularAperture: This class allows for the creation of circular apertures based on the sky coordinates (right ascension and declination) of the target objects. It is particularly useful when working with images that have World Coordinate System (WCS) information, enabling accurate placement of apertures on celestial objects.
- Application: SkyCircularAperture can be used in a variety of photometric tasks, includ- ing the measurement of fluxes from stars and other point sources. By defining apertures in sky coordinates, it ensures that the measurements are consistent across different images and observation sessions.
- Integration with Photutils: Photutils provides additional functionality, such as back- ground subtraction and error estimation, which can be easily combined with SkyCircularAperture to perform comprehensive photometric analysis.

Aperture photometry, whether conducted using tools like APT or libraries like Photutils, remains a cornerstone technique in astronomical data analysis. It provides a robust method for measuring the brightness of celestial objects, particularly in cases where the sources are well- isolated and point-like.

8 Photometry Results

SExtractor is a widely-used algorithm for source extraction and photometry. It works by detecting and cataloging sources in astronomical images, and then measuring their fluxes. This was not that usefull as we were getting high error bars and was not working good for i,r band . An image illustration for photometry using sextractor is shown here for g band. Ultimately we prefered psf fit over it.



8.1 U band Photometry

8.1.1 Challenges in U-band Photometry

U-band photometry, focusing on the ultraviolet (UV) spectrum, is particularly challenging due to atmospheric absorption and scattering. The Earth's atmosphere significantly reduces the amount of UV light reaching ground-based telescopes, leading to darker images and lower signal-to-noise ratios. These atmospheric effects make it difficult to detect and accurately measure celestial objects in the U-band. Figure 2 illustrates the difficulty in star detection due to the faintness of U-band observations.

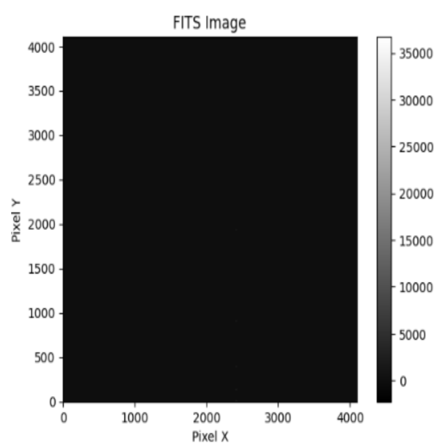


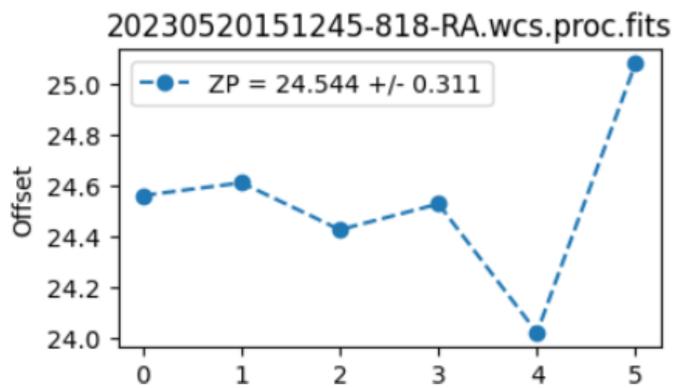
Figure 1: U band

8.1.2 Initialization and PSF Data Extraction

- SExtractor is used for the initial detection of stars in U-band images.
- This involves scanning the image for point sources, using pixel intensity thresholds to identify potential stars.
- Once stars are detected, their X, Y coordinates are extracted for further analysis.
- PSF (Point Spread Function) fitting is then applied to these detected stars to calculate their instrumental magnitudes.
- The PSF model is used to fit the detected stars, which allows for an accurate measurement of their brightness before calibration.
- The same algorithm is employed as in g, i, and r bands, with modifications to account for the lower signal levels in U-band data.

8.1.3 Calibration and Finding Constant Stars

- Calibration is essential to convert instrumental magnitudes into accurate photometric measurements.
- **Offset Plots:**
  - Offset plots are used to identify constant stars, which are stars that show consistent brightness over time.
  - Each star's brightness is compared to its catalog value, and stars with significant deviations (outliers) are excluded from the calibration process.
  - The median and standard deviation of the remaining stars' offsets are calculated to determine the photometric zero point and estimate measurement errors.
- **Relative Photometry:**
  - Involves comparing the brightness of stars over time to identify those with stable magnitudes.
  - Stable stars are ideal for calibration as their consistent brightness helps in accurately determining the zero point.
  - The figure shows a plot of magnitude corrections (magcor) over time for selected stars, with consistent magcor values indicating stable stars.



9 Final Light Curve and modeling

The culmination of these efforts is the generation of the final light curve, which provides a com- prehensive view of the brightness variations of the observed stars over time. This light curve is crucial for subsequent modeling and analysis to understand the properties and behavior of the celestial objects studied.

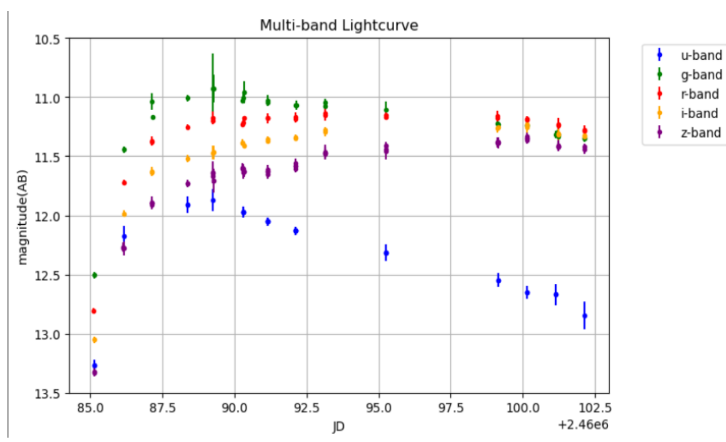


Figure 2: Light Curve

The final light curve, compiled from the g, i, r, and u band data, is attached to this report. It represents the culmination of the photometric analysis and provides the basis for further modeling and interpretation of the astrophysical phenomena under study.

9.1 Modeling the Supernova Progenitor

,

After going through the intricate processes of image reduction, photometry across different bands, and all associated complex procedures, one might wonder about the purpose of these steps. Why is it essential to perform these detailed analyses? The answer lies in the ultimate goal: to derive critical parameters of the progenitor of the supernova (SN). The observed light curves, particularly in the U-band, serve as a fundamental tool for this purpose. However, as previously mentioned, U-band photometry is significantly influenced by atmospheric conditions, necessitating corrections for the black body spectrum to ensure accurate analysis.

To achieve these corrections and proceed with accurate modeling, we employed the SuperBol package. This tool is specifically designed for analyzing supernova light curves and estimating key parameters crucial for understanding the supernova's properties.
SuperBol allows us to estimate parameters such as:

- Bolometric luminosity
- Temperature evolution
- Radius expansion

By applying SuperBol, we derived the black body corrected luminosity, which provides a more accurate representation of the supernova's energy output. This step is crucial for obtaining reliable estimates of the supernova's characteristics, free from the distortions caused by atmospheric effects.
After obtaining the corrected magnitudes from SuperBol, the next step was fitting a curve to this data. We utilized a specific luminosity model described in Section 3.1 of a research paper, which incorporates the decay of radioactive Nickel-56 (56Ni) and Cobalt-56 (56Co). These isotopes are critical for powering the supernova's light curve. The fitted curve, which models the light curve behavior over time, looks like this (see Figure ??). This curve is instrumental in understanding the temporal evolution of the supernova's brightness and provides insight into the underlying physical processes.
From the fitted curve, we can estimate the mass of 56Ni by analyzing the model's parameters. The luminosity function of the supernova is directly related to the characteristic time—a period of rapid brightening followed by a slow dimming. The slope of the plot is indicative of the mass (M), while the intercept corresponds to the characteristic time (tc).
Our estimated value for the mass of Ni is This estimation is based on the assumption that after 90 days, the light curve is primarily influenced by the decay of 56Ni, which dominates the energy output at this stage of the supernova's evolution.