

# On variable stars in globular cluster NGC 2419

Alankar Kotwal, Manikanta Reddy, Kumar Ayush  
Prof. Priya Hasan

14 December 2015

## 1 Introduction

Mankind has a specific interest in exoplanets driven by the assumption that life can only originate and exist on planets. The usual reasons that are brought forward to explain the lack of cluster planets are high stellar densities leading to high interaction rate and disruption of forming planetary systems, or even inhibition of planetary formation due to the truncation of protoplanetary disks. In globular clusters, low intrinsic metallicities could be the reason for absence of planets. However, metal-rich OCs NGC6791 ([Fe/H]=0.3) and NGC 6940 ([Fe/H]=0.3) were the subject of a number of transit surveys with null results, while a relatively metal-poor GC M4 ([Fe/H]  $\sim -1.2$ ) hosts a planet.

Thus, search is usually futile in metal poor old GCs but there is a chance of discovering a population of free floating planets (FFPs) caused by tidal disruption of planetary systems. [1]

NGC 2419 is one of the most distant and luminous globular clusters in our galaxy. With an absolute visual magnitude  $M_V \sim -9.5$  [4] and with [Fe/H]  $\sim -2.1$  dex [5], NGC 2419 is among the five brightest clusters in the Milky Way and belongs to the most metal-poor group which all, except AM-4 are located within  $R_{GC} \simeq 20$  kpc whereas NGC 2419 is put among the outer halo GCs with  $R_{GC} \simeq 90$  kpc [3]. It appears to be a very normal, low metallicity Galactic GC, the only exception being the HB “blue-hook”, a feature detected so far only in very few globulars and, most noteworthy, in those showing multiple main sequences and/or of likely extragalactic origin. [2]

A detailed catalogue of RR Lyrae stars could help get information about the origin of the cluster, and we have an opportunity to detect microlensing(ML) events which indicate existence of exoplanets in the GC.

## 2 Data

We have 93.85 minutes of data in the I band, and 138.86 minutes of data in the V band from the 2.01 m Himalayan Chandra Telescope. The exposure time for each image was 200 s and 350 s respectively. The CCD temperature was stable about 128 K with a standard deviation of less than 0.3 K.

## 3 Pre-processing

### 3.1 Sorting

We sort the images into three classes: Object, Flat and Bias. The Object and Flat images are further sorted based on the filter. (See Appendix A1 and A2)

### 3.2 Combine

Combining the bias and flat images into a master bias and master flat for each filter required the following steps:

1. Combine all bias images by average to get a master bias (Appendix B)
2. Subtract the master bias from each flat image
3. Combine all flat images by a weighted average with the inverse of exposure time as the weight.
4. Normalize the result to obtain master flat (Appendix B)

Steps 2-4 are executed separately for each filter. Note that we have no rational explanation for Step 3 as the counts do not scale proportional to the exposure times. An ad-hoc hypothesis that flats with higher exposure time are less relevant is not satisfactory. (See Appendix A3)

### 3.3 Align

We need the images to be aligned in order to fix the co-ordinates of a star and perform time series photometry over the set of images. The most commonly used techniques use the World Coordinate System(WCS) data specified in the header of the image.

In the case that WCS data is missing, this process of image registration is simplified using Discrete Fourier Transforms(DFT). The translational shift in the image domain corresponds to a phase shift in the Fourier domain. Inaccuracies in this calculation however limit the significance of sub-pixel resolution in the alignment process. (See Appendix A4)

The alignment process was made difficult due to dark borders in the flat image which adds noise at borders in the flat corrected image.(Appendix B) This noise adds a phase shift in the fourier domain. To evade this, we trimmed the images before alignment.

## 4 Aperture Photometry

### 4.1 Introduction

Aperture photometry is a technique that makes no assumption about the actual shape of the source PSF but simply collects and sums up the observed counts within a specified aperture centered on the source. The aperture used may be circular (usually the case for point sources), square, or any shape deemed useful.

The basic application of aperture photometry starts with an estimate of the center of the PSF and then inscribes a circular software aperture of radius  $r$  about that center. The radius  $r$  may simply be taken as three times the fullwidth at half-maximum : the radius of a PSF that would contain 100% of the flux from the object.

There is a well behaved relation between the radius of the aperture of extraction of a point source and the resultant S/N obtained for such a measurement. An optimum radius aperture, that is, one that provides the optimum or best S/N for the measurement, can be determined for any given PSF and generally has a radius of near FWHM. This optimum radius is a weak function of the source brightness, becoming smaller in size for fainter sources. [6]

### 4.2 Analysis

We use the Aperture Photometry Tool [7] to give us the co-ordinates of stars and the FWHM for a PSF which is the sum of a third degree polynomial and a gaussian. The identification and selection of stars is done manually. This is partially an irrational decision. The high density of stars at the center of the GC make automated detection of stars difficult and all the saturated pixels at the center are discarded without second thought. The human eye in this case, can help identify at least a few stars with good confidence. For stars further from the center, automation could give better results but we put more faith in our own eyes.

The aperture radius and co-ordinates thus obtained are used to sum up counts and generate a time series. The counts are normalized by exposure times and converted to a log scale

$$m = -2.5 \log n + 10$$

This time series had to be processed further to correct for atmospheric extinction. This required us to calculate the zenith angle for the image and correct the magnitude by  $\Delta m \log \zeta$ , where  $\zeta$  is the zenith angle. The  $\Delta m$  depends on the atmospheric conditions during the observation and has to be calibrated from some stars confidently known to not be variables. (See Appendix A5)

For plots, refer Appendix B

## 5 Discussion

All plots without correction for atmospheric extinction have similar features, which persist after the correction. We also show magnitude versus  $\sec \zeta$  which is expected to be a straight line.

We present a selection of 5 random stars in the I band and note the similar features you observe for them, indicating a bias in the images. Our first hypothesis that the discrepancy lies due to atmospheric extinction does not completely account for the effect.

Although, note that the only major discrepancy lies in the first four data points which appear to be underexposed as compared to other images on visual examination. If we ignore those points, the photometry procedure as prescribed seems fit to be applied to further data sets and thus the variability of stars in NGC 2419 can be studied and catalogued.

## A Code

### A.1 Sort on Type

```
#!/usr/bin/python

#import required packages
from astropy.io import fits
from os import listdir
from os.path import isfile, join
from shutil import copyfile

#all images
images = listdir('./data')

for image in images:
    try:
        #open header
        hdu = fits.open('./data/'+image, ignore_missing_end=True)

        #sort accordingly
        if hdu[0].header['object']=='flat':
            copyfile("./data/"+image, "./data_flat/"+image)
        elif hdu[0].header['object']=='bias':
            copyfile("./data/"+image, "./data_bias/"+image)
        elif hdu[0].header['object']=='NGC2419':
            copyfile("./data/"+image, "./data_obj/"+image)
        else:
            print "Unknown file type"
    #if error
    except IOError:
        print "Corrupt file or can't read", image
```

## A.2 Sort on Filter

```
#import required packages
from astropy.io import fits
from os import listdir
from os.path import isfile, join
from os import rename
import pdb

obj_dir = 'data_obj/'
flat_dir = 'data_flat/'

obj_names = listdir(obj_dir)
flat_names = listdir(flat_dir)

obj_names.remove('i')
obj_names.remove('v')
flat_names.remove('i')
flat_names.remove('v')

obj_images = [fits.open(obj_dir+i, ignore_missing_end=True)[0].header['FILTER']]
for i in obj_names]
flat_images = [fits.open(flat_dir+i, ignore_missing_end=True)[0].header['FILTER']]
for i in flat_names]

#pdb.set_trace()

for i in range(len(obj_names)):
    #pdb.set_trace()
    if 'I' in obj_images[i]:
        rename(obj_dir+obj_names[i], obj_dir+'i/'+obj_names[i])
    elif 'V' in obj_images[i]:
        rename(obj_dir+obj_names[i], obj_dir+'v/'+obj_names[i])
    else:
        print 'Unknown band!'

for i in range(len(flat_names)):
    #pdb.set_trace()
    if 'I' in flat_images[i]:
        rename(flat_dir+flat_names[i], flat_dir+'i/'+flat_names[i])
    elif 'V' in flat_images[i]:
        rename(flat_dir+flat_names[i], flat_dir+'v/'+flat_names[i])
    else:
        print 'Unknown band!'
```

### A.3 Combine

In the following code, we first create a master bias, subtract it from all the flats and create master flats as described in the text

```
from astropy.io import fits
from astropy import units as u
from ccdproc import combiner
import ccdproc
from os import listdir
import numpy as np
import pdb

print "Combining..."

# All images: biases
images = [ccdproc.CCDData.read('data_bias/'+i, unit='adu') for i in
listdir('./data_bias')]

c_images = combiner.Combiner(images)
master_bias = c_images.average_combine()

master_bias.write('data_bias/master_bias.fits')

# All images: flats i
images = [ccdproc.CCDData.read('data_flat/i/'+i, unit="adu") for i in
listdir('./data_flat/i/')]
sub_images = [ccdproc.subtract_bias(i, master_bias) for i in images]
c_images = combiner.Combiner(sub_images)
c_images.scaling = [1/float(fits.open('data_flat/i/'+i)[0].header["EXPTIME"])]
for i in listdir("./data_flat/i")]
master_flat_i = c_images.average_combine()
master_flat_i.data = master_flat_i.data/np.ma.average(master_flat_i.data)
master_flat_i.write('data_flat/i/master_flat.fits')

# All images: flats v
images = [ccdproc.CCDData.read('data_flat/v/'+i, unit="adu") for i in
listdir('./data_flat/v/')]
sub_images = [ccdproc.subtract_bias(i, master_bias) for i in images]
c_images = combiner.Combiner(sub_images)
c_images.scaling = [1/float(fits.open('data_flat/v/'+i)[0].header["EXPTIME"])]
for i in listdir("./data_flat/v")]
master_flat_v = c_images.average_combine()
master_flat_v.data = master_flat_v.data/np.ma.average(master_flat_v.data)
master_flat_v.write('data_flat/v/master_flat.fits')

# c_images = combiner.Combiner(images)
# c_images.scaling = lambda arr: 1/np.ma.average(arr)
# master_flat_v = c_images.average_combine()

# master_flat_v.write('data_flat/v/master_flat.fits')
```

## A.4 Align

```
import os

import scipy as sp
import scipy.misc
import imreg_dft as ird
from astropy.io import fits
print "Libraries imported..."

names_i = os.listdir('./data_corr_trimmed/i')
names_v = os.listdir('./data_corr_trimmed/v')
print "Directories enlisted..."

print "Entering I alignments..."
im_ref = fits.open('data_corr_trimmed/i/'+names_i[0])[0].data
for i in range(len(names_i)):
    im_test = fits.open('data_corr_trimmed/i/'+names_i[i])[0].data
    result = ird.similarity(im_ref, im_test, numiter=2)
    print i, names_i[i], "tvec:", result['tvec'], " success:", result['success']
    out_hdu = fits.PrimaryHDU(result['timg'])
    out_hdu.writeto('data_align/i/'+names_i[i])
print "I alignments done..."

print "Entering V alignments..."
im_ref = fits.open('data_corr_trimmed/v/'+names_v[8])[0].data
for i in range(len(names_v)):
    im_test = fits.open('data_corr_trimmed/v/'+names_v[i])[0].data
    result = ird.similarity(im_ref, im_test, numiter=2)
    print i, names_v[i], "tvec:", result['tvec'], " success:", result['success']
    out_hdu = fits.PrimaryHDU(result['timg'])
    out_hdu.writeto('data_align/v/'+names_v[i])
print "V alignments done..."
```

## A.5 Photometry

```
from astropy.io import fits
from photutils import CircularAperture
from photutils import aperture_photometry
from math import log10
from astropy import units as u
from astropy.time import Time
from astropy.coordinates import SkyCoord, EarthLocation, AltAz
import pdb

slope_i = 0
slope_v = 0

ngc = SkyCoord(["07h38m08.51s +38d52m54.9s"])
myObs = EarthLocation(lat=32.779444*u.deg, lon=78.964167*u.deg, height=4500*u.m)

#I Band
#These will be the list of 'list' of coordinates and radius of stars of interest
nameFile = open('dates_i.csv', 'r').readlines()
timeDict = {}

for i in nameFile:
    entries = i.rstrip().split(',')
    timeDict[entries[0]] = (float(entries[2]), entries[1])

coordFile = open('coords_I.csv', 'r').readlines()
coordinates = []

for i in coordFile:
    entries = i.rstrip().split(',')
    thisCoord = [float(entries[0]), float(entries[1]), float(entries[2])]
    coordinates.append(thisCoord)

for i in range(len(coordinates)):

    outfile = open('photos/i/'+str(i)+'.csv', 'w+')

    for key in sorted(timeDict.keys(), key = str.lower):

        aperture = CircularAperture(
            [(coordinates[i][0], coordinates[i][1]), r=coordinates[i][2]])
        phot_table = aperture_photometry(
            fits.open('data_final/i/'+key)[0].data, aperture)
        thisMag = -2.5*log10(
            float(phot_table['aperture_sum']))
            /float(fits.open('data_obj/i/'+key)[0].header['EXPTIME']))
            +10

        time = Time(timeDict[key][1])
        altaz = ngc.transform_to(
            AltAz(obstime = time, location = myObs))

        secZenAngle = altaz.secz

        finalMag = thisMag - float(secZenAngle)*slope_i
        thisLine = str(timeDict[key][1]) + ','
        + str(timeDict[key][0]) + ',' + str(finalMag) + '\n'
```



```

        outfile.write(thisLine)

outfile.close()

# V Band
#These will be the list of 'list' of coordinates and radius of stars of interest
nameFile = open('dates_v.csv', 'r').readlines()
timeDict = {}

for i in nameFile:
    entries = i.rstrip().split(',')
    timeDict[entries[0]] = (float(entries[2]), entries[1])

coordFile = open('coords_V.csv', 'r').readlines()
coordinates = []

for i in coordFile:
    entries = i.rstrip().split(',')
    thisCoord = [float(entries[0]), float(entries[1]), float(entries[2])]
    coordinates.append(thisCoord)

for i in range(len(coordinates)):

    outfile = open('photos/v/'+str(i)+'.csv', 'w+')

    for key in sorted(timeDict.keys(), key = str.lower):

        aperture = CircularAperture(
            [(coordinates[i][0], coordinates[i][1])], r=coordinates[i][2])
        phot_table = aperture_photometry(
            fits.open('data_final/v/'+key)[0].data, aperture)
        thisMag = -2.5*log10(
            float(phot_table['aperture_sum']))
            /float(fits.open('data_obj/v/'+key)[0].header['EXPTIME'])+10

        time = Time(timeDict[key][1])
        altaz = ngc.transform_to(
            AltAz(obstime = time, location = myObs))

        secZenAngle = altaz.secz

        finalMag = thisMag - float(secZenAngle)*slope_v
        thisLine = str(timeDict[key][1]) + ','
        + str(timeDict[key][0]) + ',' + str(finalMag) + '\n'
        outfile.write(thisLine)

outfile.close()

```

## B Images and Plots

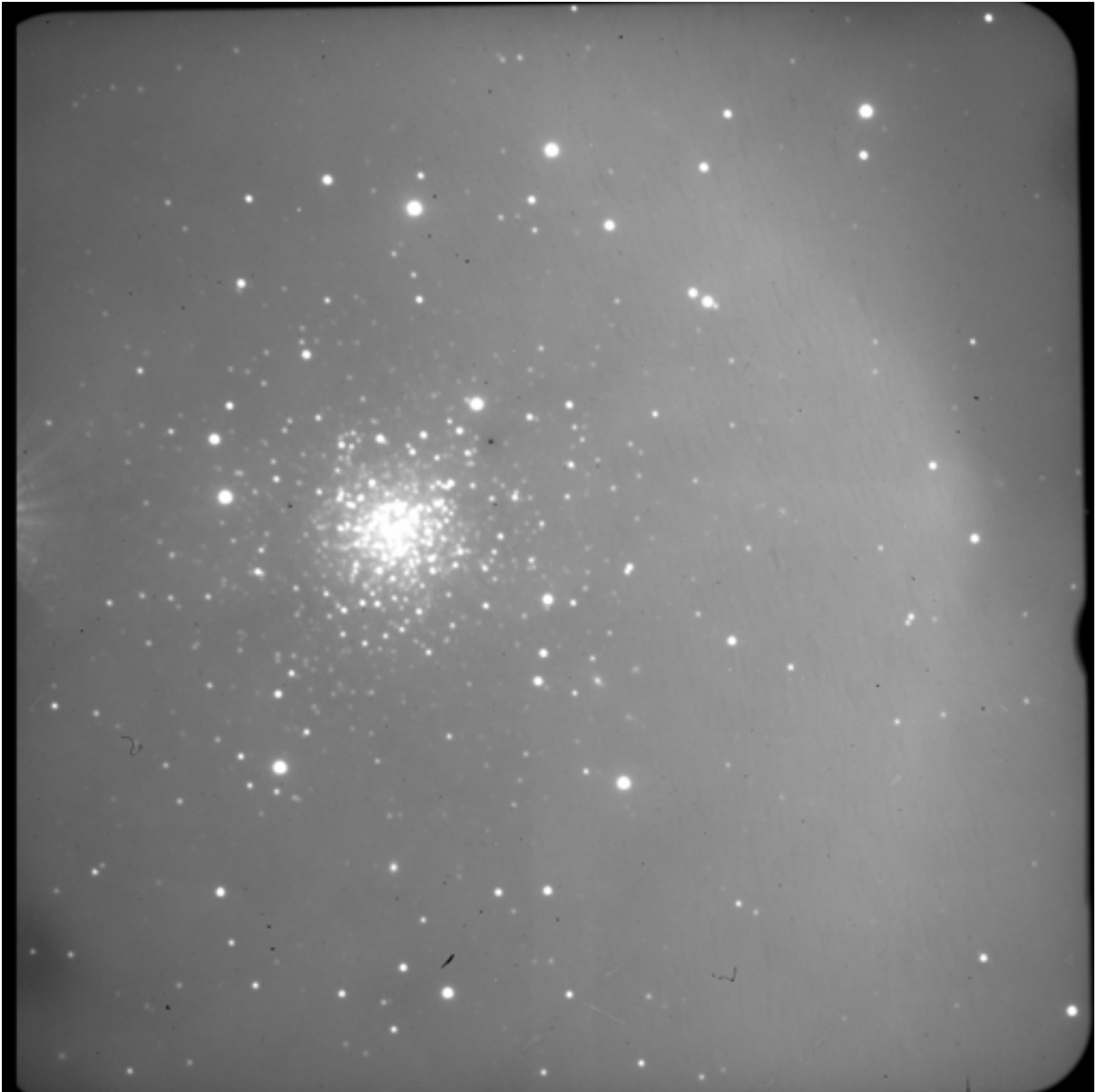


Figure 1: I-band Raw Image

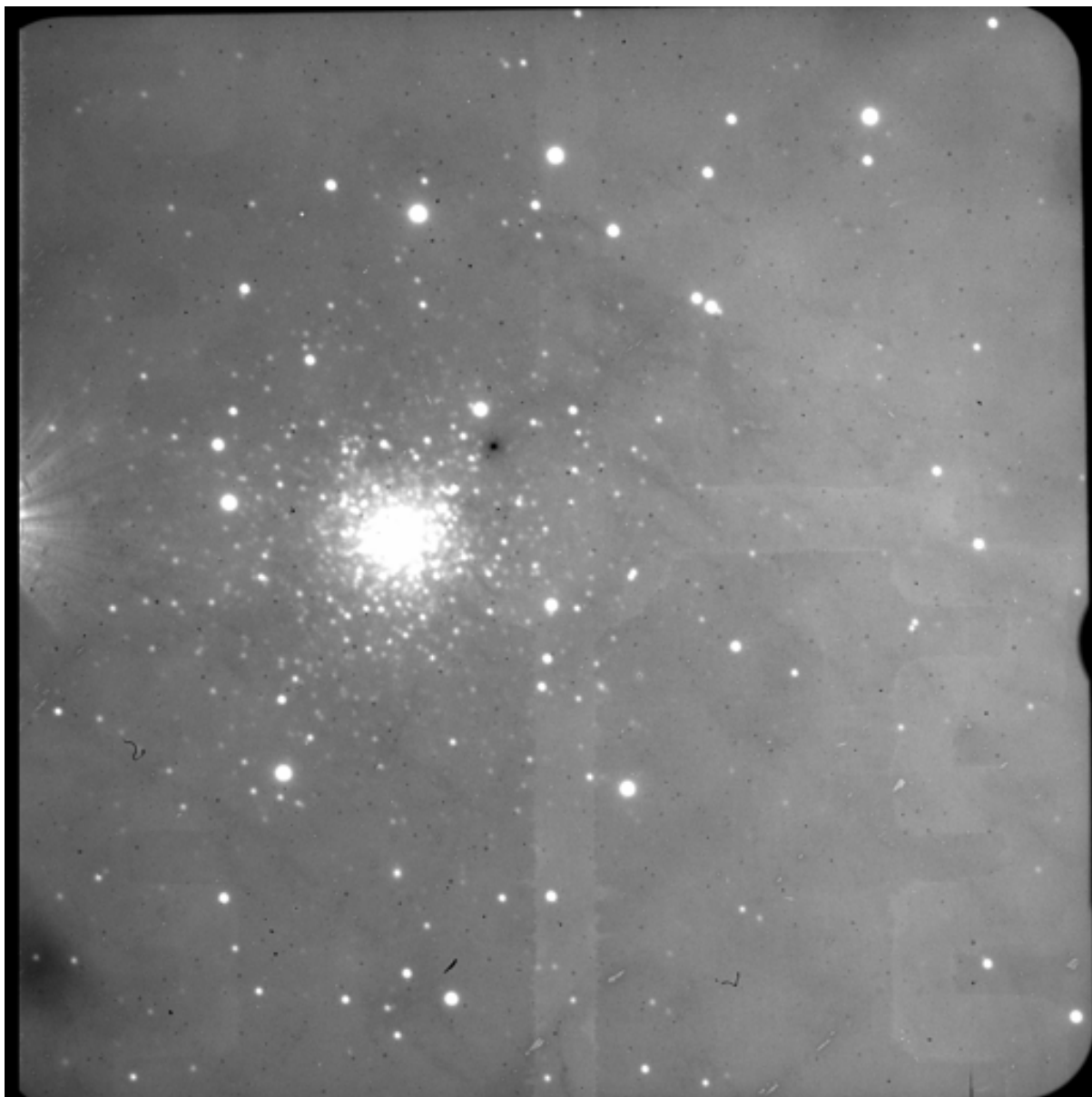


Figure 2: V-band Raw Image

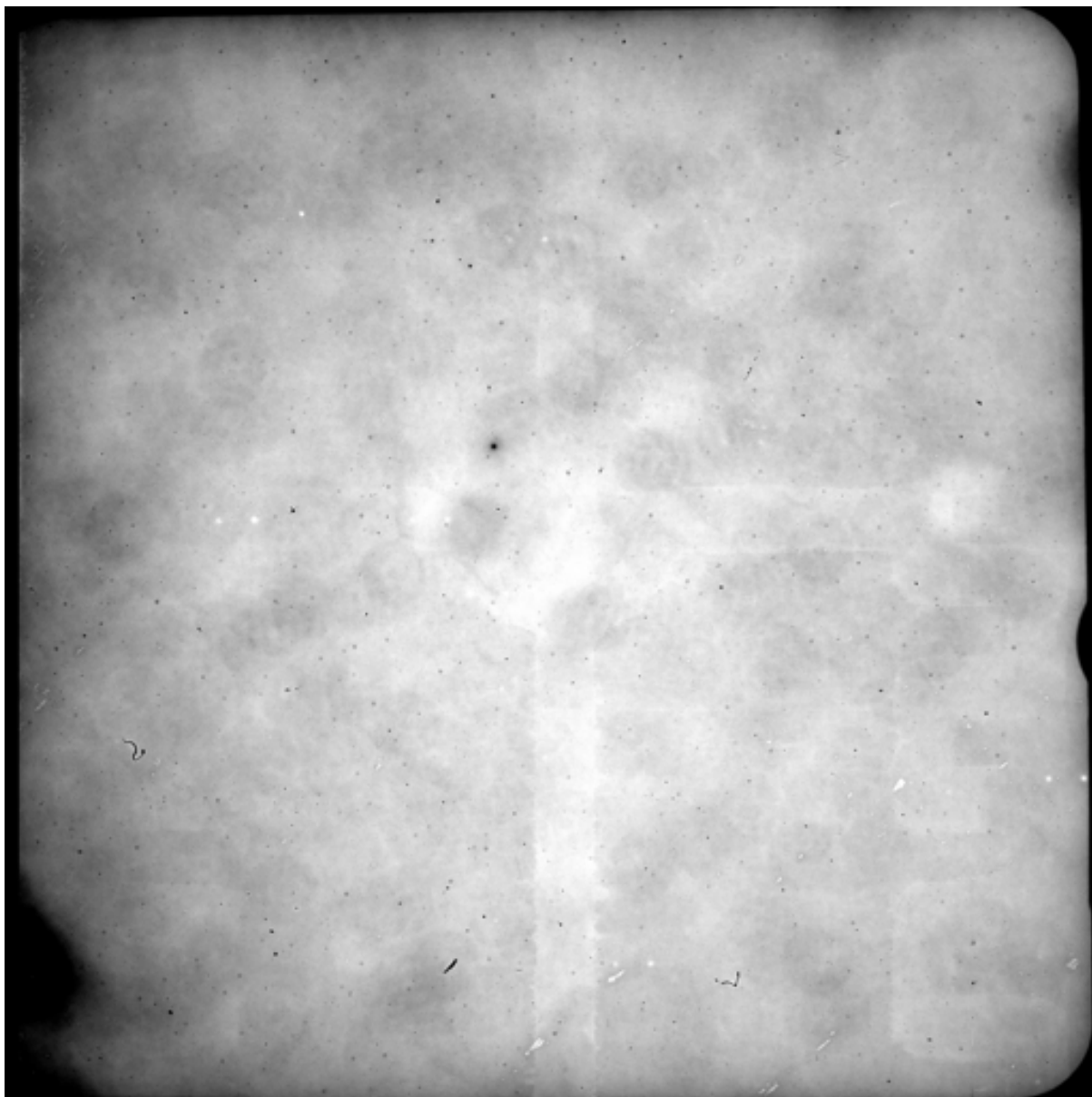


Figure 3: I-band Flat Image



Figure 4: V-band Flat Image

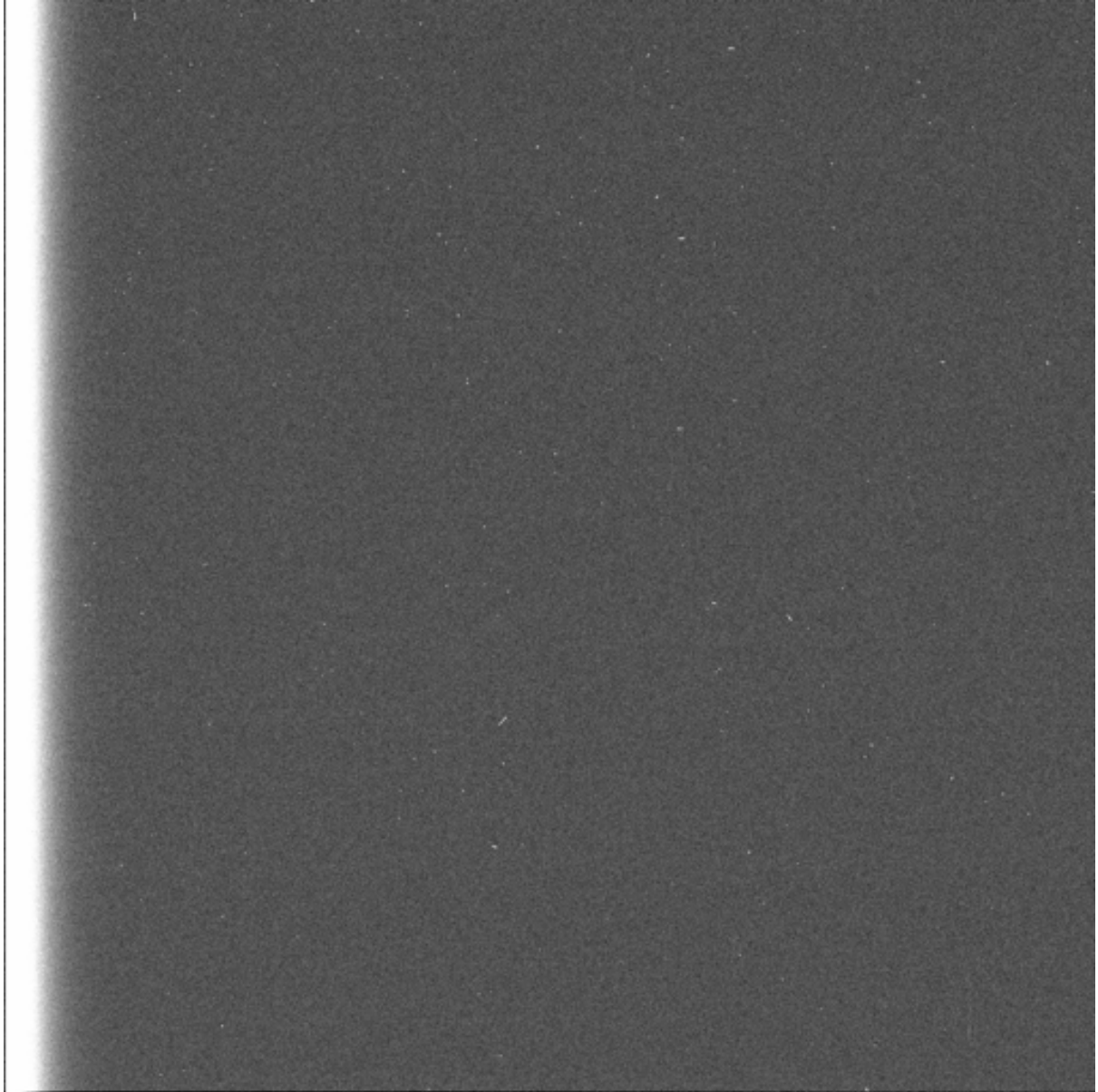


Figure 5: Master Bias Image



Figure 6: I-band Final Image

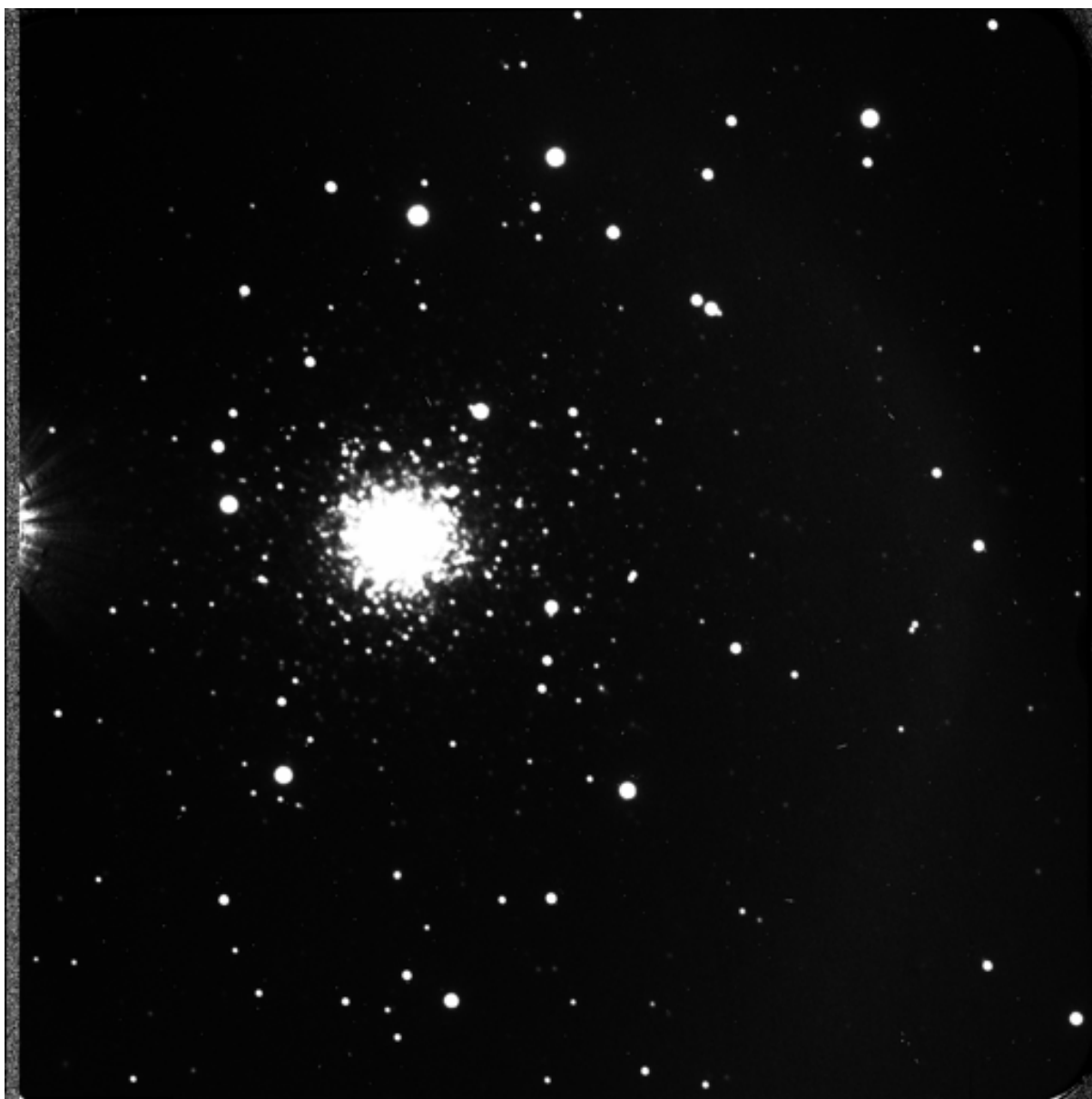


Figure 7: V-band Final Image



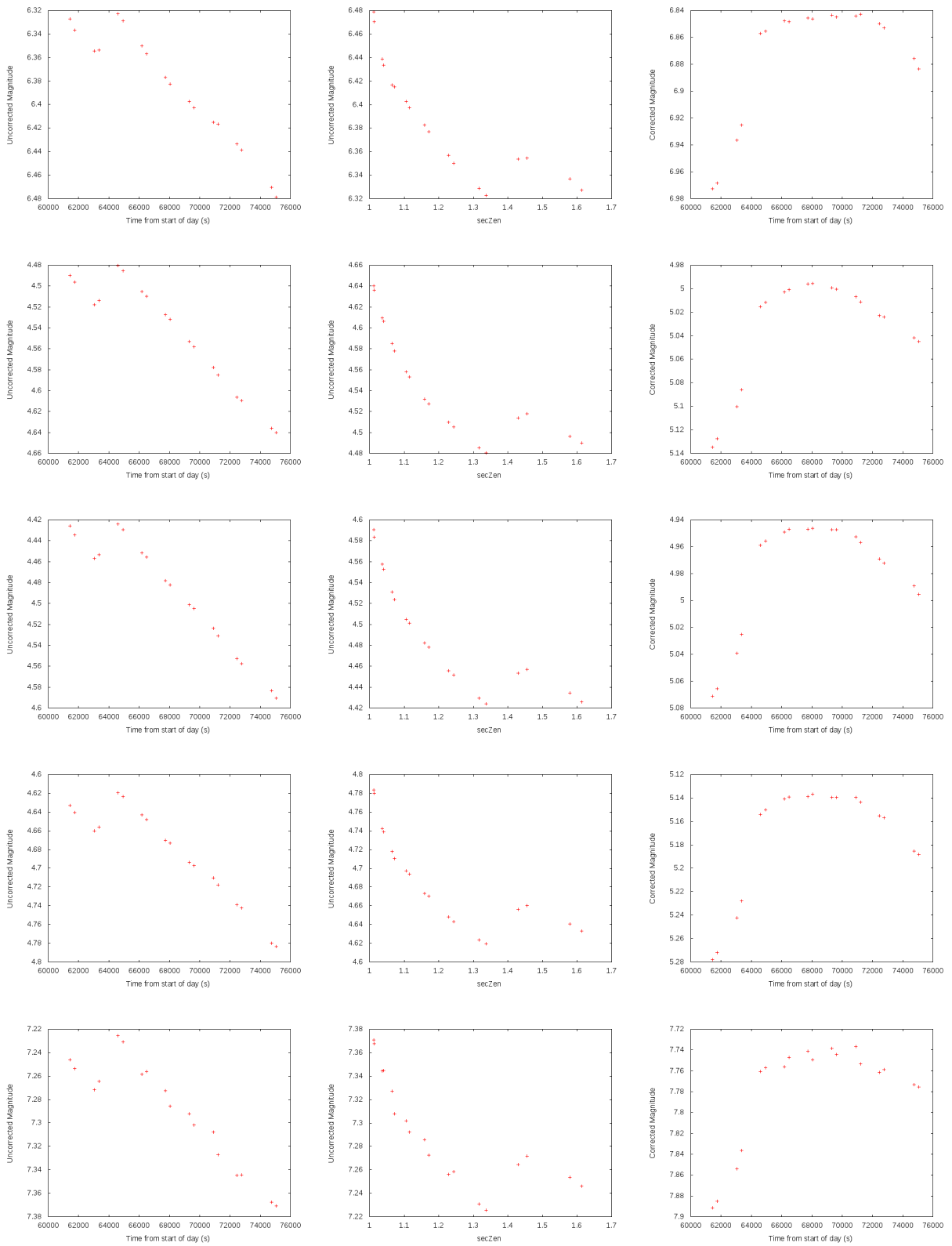


Figure 8: Uncorrected Time Series, Magnitude vs Zenith Angle, Corrected Time Series

## Bibliography

- [1] M. Safonova, D. Mkrtichian, P. Hasan, F. Sutaria, N. Brosch, E. Gorbikov, P. Joseph.  
*Search for low-mass objects in the globular cluster M4. I: Detection of variable stars.*
- [2] Vincenzo Ripepi, Gisella Clementini, Marcella Di Criscienzo, Claudia Greco, Massimo Dall’Ora, Luciana Federici, Luca Di Fabrizio, Ilaria Musella, Marcella Marconi, Lara Baldacci, Marcella Maio.  
*On the remote Galactic globular cluster NGC 2419.*  
arXiv:0705.0966v2.
- [3] Harris, W. E. 1996, AJ, 112, 1487
- [4] Harris, W. E., et al. 1997, AJ, 114, 1030
- [5] Suntzeff, N. B., Kraft, R. P., —& Kinman, T. D., 1988, AJ, 95, 91
- [6] Steve B. Howell.  
Handbook of CCD Astronomy
- [7] Aperture Photometry Tool  
<http://www.aperturephotometry.org/>
- [8] AstroPy  
<http://www.astropy.org/>
- [9] ccdproc  
<https://github.com/astropy/ccdproc>
- [10] photutils  
<https://github.com/astropy/photutils>