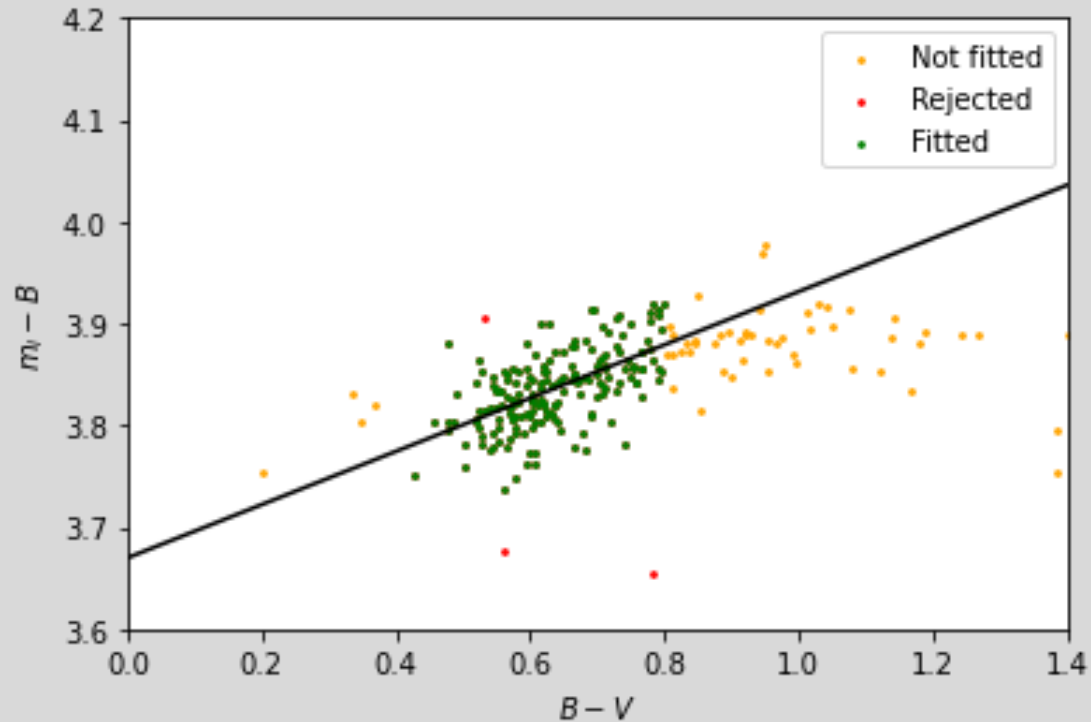
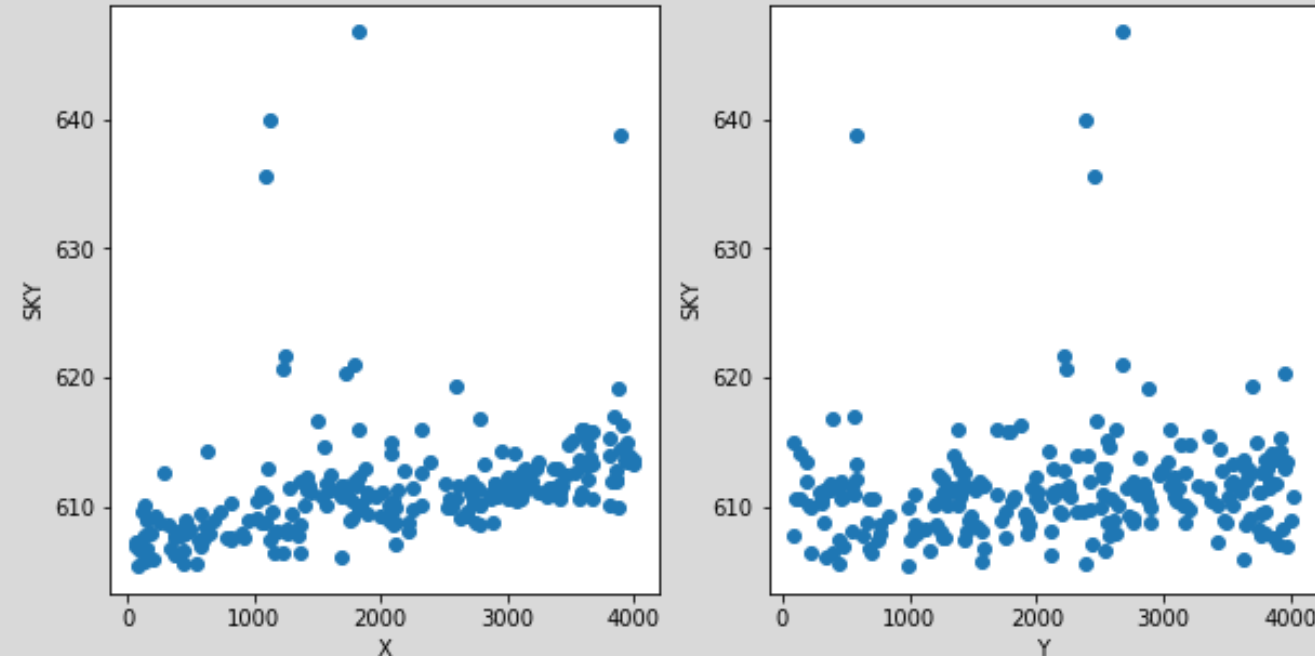


# Recapping what we did last time, for each image

## Photometric Calibration



## Sky Fit and Subtraction



$$m_{inst} - m_B = C_B(B - V) + ZP_{B,IMAGE}$$

$$C_B \text{ (slope)} = 0.262 \pm 0.027$$

$$ZP_{B,IMAGE} \text{ (intercept)} = 3.670 \pm 0.017$$

all this info  
written into  
image headers

$$SKY = X \times \nabla_{SKY,X} + Y \times \nabla_{SKY,Y} + SKY_0$$

$$\nabla_{SKY,X} = 1.468 (\pm 0.082) \times 10^{-3} \text{ ADU/pix}$$

$$\nabla_{SKY,Y} = 0.440 (\pm 0.084) \times 10^{-3} \text{ ADU/pix}$$

$$SKY_0 = 606.50 \pm 0.27 \text{ ADU}$$

This process (calibration and sky subtraction) prepares each image for combining. We do this for all images in the B and V image sets.

**CalibrateImages.ipyb** calculates photometric solutions for all images, fits the sky plane, then writes out new versions of each image with the sky levels subtracted off.

Look at photometric solutions and sky levels for the B-band images.

#### Differences in color term (C)

*Mostly random scatter*

#### Differences in zeropoint (ZP)

*Real, systematic changes. Images were taken at different airmasses*

#### Differences in sky level (SKY0)

*Real, systematic changes. Images were taken at different airmasses and times.*

imname	C	Cerr	ZP	ZPerr	SKY0	SKY0err
rpobj0419029.fits	0.262	0.027	3.670	0.017	606.504	0.265
rpobj0419030.fits	0.269	0.022	3.646	0.014	572.340	0.260
rpobj0419031.fits	0.266	0.024	3.637	0.016	553.642	0.312
rpobj0419032.fits	0.269	0.024	3.622	0.016	533.853	0.265
rpobj0419038.fits	0.263	0.029	3.605	0.018	486.710	0.251
rpobj0419039.fits	0.319	0.028	3.573	0.018	478.417	0.222
rpobj0419040.fits	0.284	0.021	3.596	0.014	474.634	0.262
rpobj0419041.fits	0.288	0.023	3.599	0.015	481.319	0.249

# What's next?

Each image also has contamination/noise in it due to:

- CCD Read noise (remember: subtracting off the master eliminates the fixed pattern noise, but not the random read noise. That cannot be removed from an individual image.
- Variations in sky intensity
- Scattered light and reflections inside the telescope
- Satellites passing through the field of view
- Bad pixels and columns in the CCD
- Cosmic rays

To reduce noise and remove contamination, we want to

- **shift** each image so that M101 is at the same spot
- **scale** each image to a common intensity
- do a **median stack** of all the images.

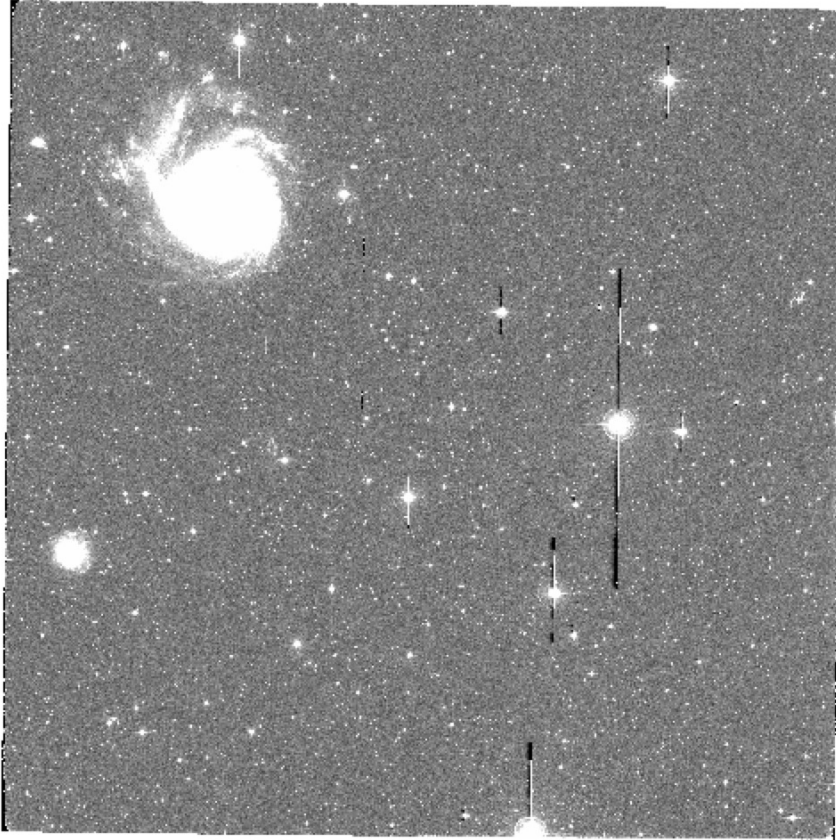


## Shifting the images to a common center: “Image re-registration”

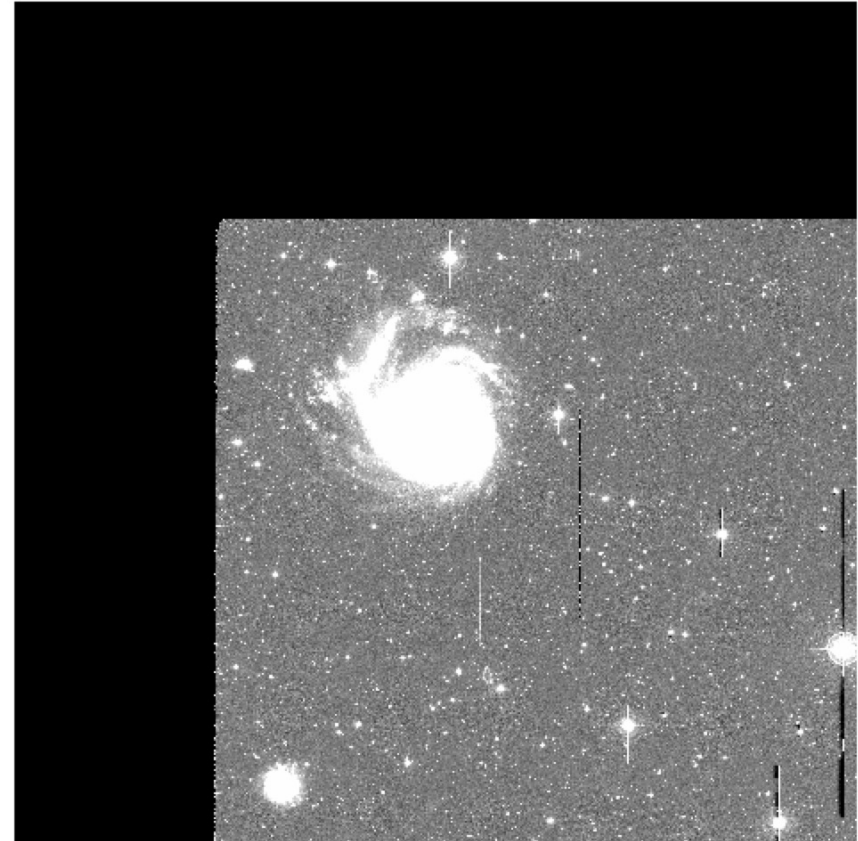
Simplest version: XY integer pixel shift (“shift flux in each pixel over 100 pixels and down 200 pixels”)

In reality: non-integer XY shifts, plus rotation. Geometric transformation is mathematically intensive, and introduces additional uncertainty into the data.

Original Image



Re-registered Image



## Applying Zeropoints: Photometric Scaling

The same star will have different numbers of counts in each image due to the different zeropoints. We can define a “final zeropoint” and scale each image up or down in intensity to match this average zeropoint.

$$ZP\_FINAL = np.average(ZP\_image)$$

Since zeropoints are in magnitudes, we can say

$$ZP_{final} - ZP_{image} = -2.5 \log(I_{final}/I_{image})$$

Then we scale each image in intensity by a factor of

$$I_{final} = I_{image} \times 10^{-0.4(ZP_{final} - ZP_{image})}$$

imname	ZP	scale
rpobj0419029.fits	3.670	1.048
rpobj0419030.fits	3.646	1.026
rpobj0419031.fits	3.637	1.017
rpobj0419032.fits	3.622	1.003
rpobj0419038.fits	3.605	0.988
rpobj0419039.fits	3.573	0.959
rpobj0419040.fits	3.596	0.979
rpobj0419041.fits	3.599	0.983

That  $10^{\wedge}$  term is the photometric scaling we multiply each image by to get them on the same final zeropoint. After scaling all the images this way, a given star should have the same number of counts (+/- noise) in each image.



## Final Image Combine

1. Re-register each image so that M101 is at the center of the image.
2. Scale each image in intensity to match the average zeropoint:

$$I_{scaled} = I_{image} \times 10^{-0.4(ZP_{avg} - ZP_{image})}$$

3. Create a final image by calculating the median pixel intensity along a stack of the shifted, scaled images.

*This takes 3-5 minutes for each of the B and V stacks.....*



# Applying your photometric solution to the final combined images

## Part 1: Instrumental magnitudes

In the notebooks, we defined Instrumental magnitudes in terms of counts/second:

$$m_{inst} = -2.5 \log(I/t_{exp}) + 25$$

So in analyzing the reduced images, we need to define our instrumental magnitudes the same way.

And since we medianed the images (rather than summing them),  $t_{exp}$  is the exposure time of an *individual* image:

- V images: 900 seconds (15 mins)
- B images: 1200 seconds (20 mins)

So the first step in doing photometry is to measure counts in each of the B and V images, and turn those into instrumental magnitudes ( $m_{inst,B}$  and  $m_{inst,V}$ ) using those exposure times.

# Applying your photometric solution to the final combined images

## Part 2: Turn instrumental magnitudes into real magnitudes

Our photometric solution:

$$m_{inst,B} - m_B = C_B(B - V) + ZP_B$$

$$m_{inst,V} - m_V = C_V(B - V) + ZP_V$$

FINAL	B	V	
C	0.277	0.233	average color term for all images
ZP	3.619	3.541	average ZP that all images were scaled to.
EXPTIME	1200	900	exposure time per image



# Applying your photometric solution to the final combined images

## Part 2: Turn instrumental magnitudes into real magnitudes

Our photometric solution:

$$m_B = m_{inst,B} - C_B(B - V) - ZP_B$$

$$m_V = m_{inst,V} - C_V(B - V) - ZP_V$$

But wait....

FINAL	B	V	
C	0.277	0.233	average color term for all images
ZP	3.619	3.541	average ZP that all images were scaled to.
EXPTIME	1200	900	exposure time per image

Solve two equations with two unknowns:

$$m_B - m_V = [m_{inst,B} - m_{inst,V}] - [C_B - C_V](B - V) - [ZP_B - ZP_V]$$

$$B - V = [m_{inst,B} - m_{inst,V}] - [C_B - C_V](B - V) - [ZP_B - ZP_V]$$

$$(B - V)(1 + [C_B - C_V]) = [m_{inst,B} - m_{inst,V}] - [ZP_B - ZP_V]$$

$$(B - V) = ([m_{inst,B} - m_{inst,V}] - [ZP_B - ZP_V]) / (1 + [C_B - C_V])$$

# Applying your photometric solution to the final combined images

## Summary

*First* measure counts and calculate instrumental magnitudes in each filter:

$$m_{inst,B} = -2.5 \log(I_B/t_{exp,B}) + 25$$
$$m_{inst,V} = -2.5 \log(I_V/t_{exp,V}) + 25$$

*Then* calculate the color:

$$(B - V) = ([m_{inst,B} - m_{inst,V}] - [ZP_B - ZP_V]) / ((1 + [C_B - C_V]))$$

*Then* insert that color into the photometric solution to calculate magnitudes:

$$m_B = m_{inst,B} - C_B(B - V) - ZP_B$$

$$m_V = m_{inst,V} - C_V(B - V) - ZP_V$$

FINAL	B	V	
C	0.277	0.233	average color term for all images
ZP	3.619	3.541	average ZP that all images were scaled to.
EXPTIME	1200	900	exposure time per image

## One last step – correcting for galactic extinction

**After** all photometry is done and you have your “final” magnitudes and colors, you want to correct for galactic extinction. Dust in the Milky Way (which we are looking through) both dims and reddens the light from M101.

Look up the galactic extinction on NED, using the estimate from Schlafly and Finkbeiner (2011). Then correct for extinction in each band by doing:

$$\begin{aligned}m_{B,0} &= m_{B,obs} - A_B \\ m_{V,0} &= m_{V,obs} - A_V\end{aligned}$$

And then calculate the color one of two ways:

**correct the observed color:**

or

**calculate the color from the corrected magnitudes:**

$$(B - V)_0 = (B - V)_{obs} - (A_B - A_V)$$

$$(B - V)_0 = m_{B,0} - m_{V,0}$$

**But don't do both!** That is, don't calculate your color from the corrected magnitude and then *also* apply the reddening correction.

And don't try to correct for galactic extinction before calculating your final observed magnitudes and colors!

**Photometry.ipynb:** turns counts on the images ( $I_B, I_V$ ) into calibrated apparent magnitudes ( $m_B, m_V$ ).

```
In [ ]: import numpy as np
```

```
In [ ]: def photosol(IB,IV):  
  
    # UPDATE THESE VALUES!  
    exptime_B, ZP_B, cterm_B = 0., 0., 0.  
    exptime_V, ZP_V, cterm_V = 0., 0., 0.  
  
    m_inst_B=-2.5*np.log10(IB/exptime_B)+25.0  
    m_inst_V=-2.5*np.log10(IV/exptime_V)+25.0  
  
    BmV=((m_inst_B-m_inst_V)-(ZP_B-ZP_V))/(1.0+(cterm_B-cterm_V))  
  
    mB=m_inst_B-cterm_B*BmV-ZP_B  
    mV=m_inst_V-cterm_V*BmV-ZP_V  
  
    return mB,mV
```

```
In [ ]: IB, IV = 1.0,1.0  
        mB, mV = photosol(IB,IV)  
        print('mB = {:.2f}, mV = {:.2f}'.format(mB,mV))
```

*(then for the final answer, you also want to correct those magnitudes for Milky Way galactic extinction)*