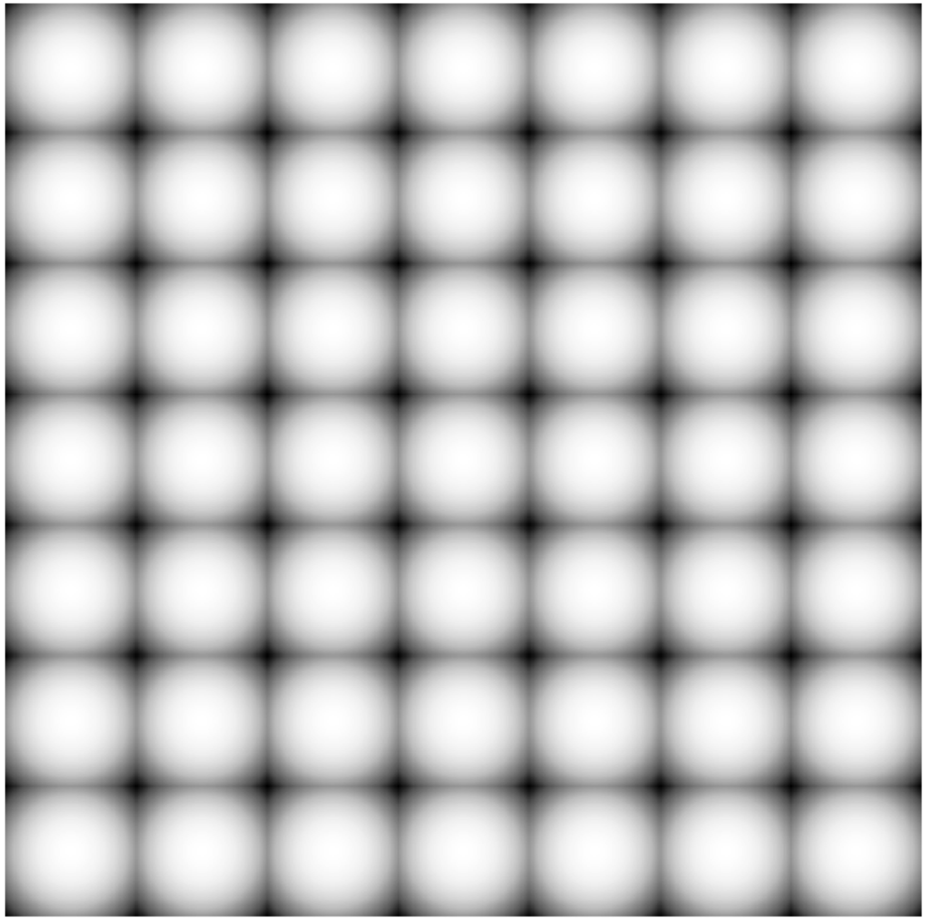
Simulated CCD Photometry: An Application for *K2* Sputtering

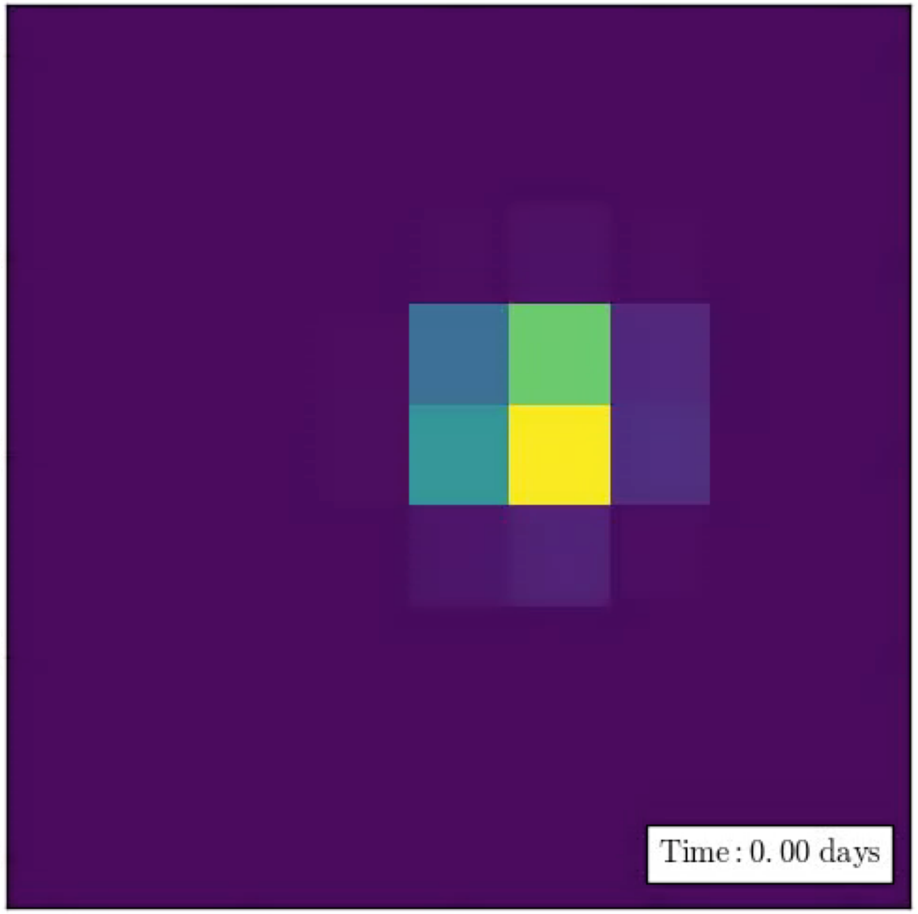
Nicholas Saunders

NASA Ames

May 31, 2018

K2 Sputtering



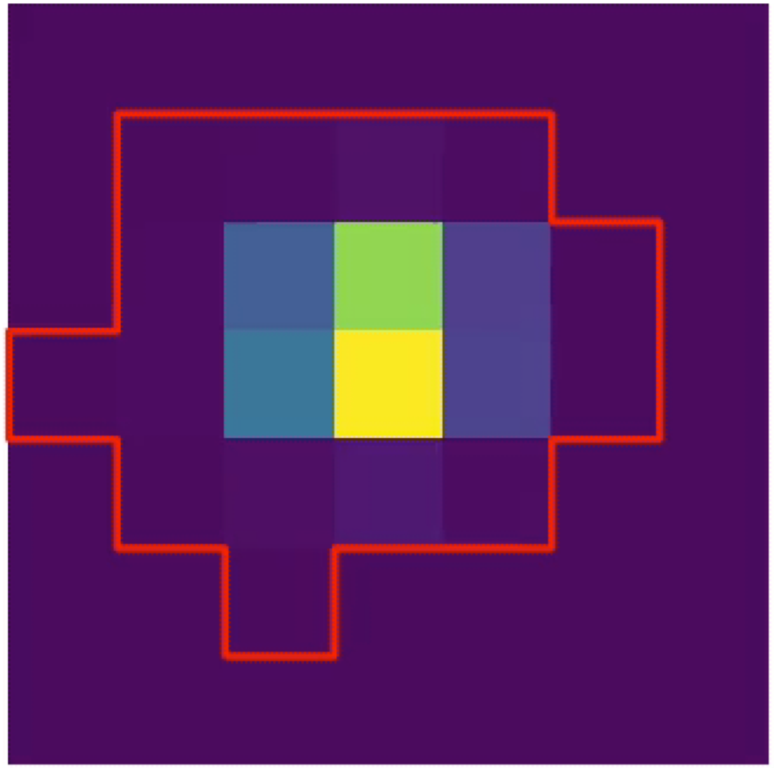


Everest Pipeline
(Luger et al. 2016, 2017)

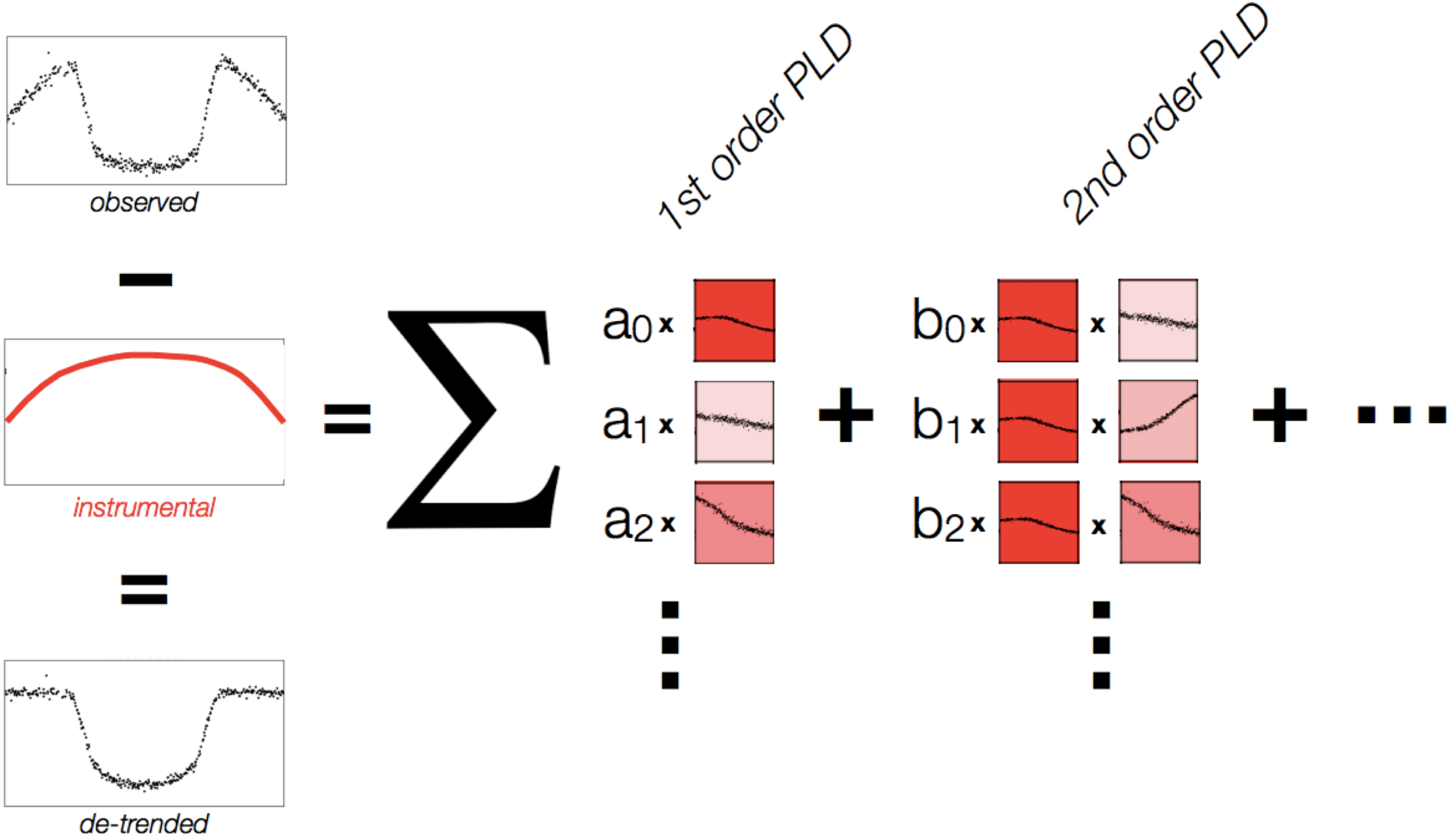
Pixel Level Decorrelation
(Deming et al. 2015)

+

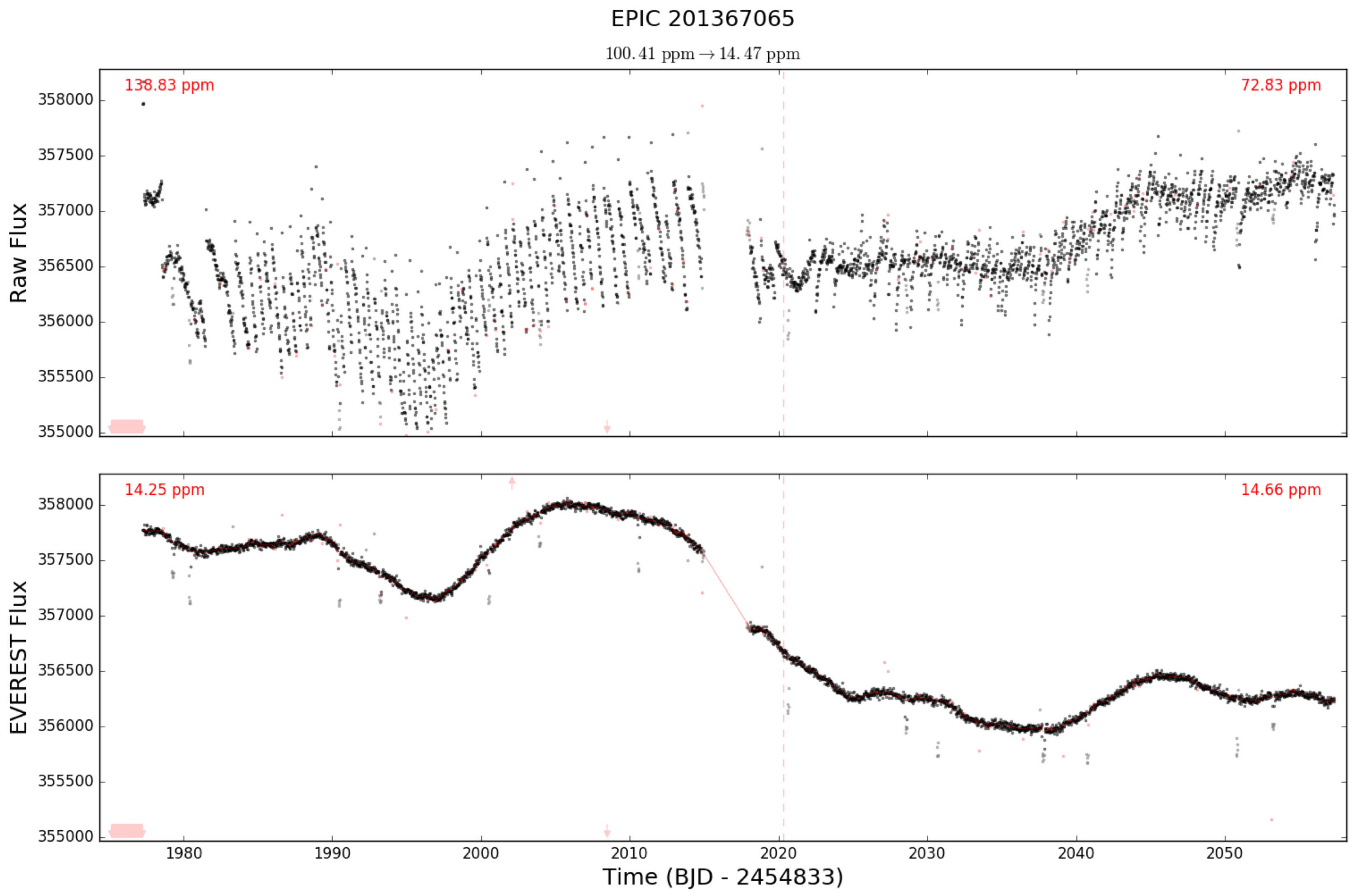
Aperture-Fitting



K2 Sputtering

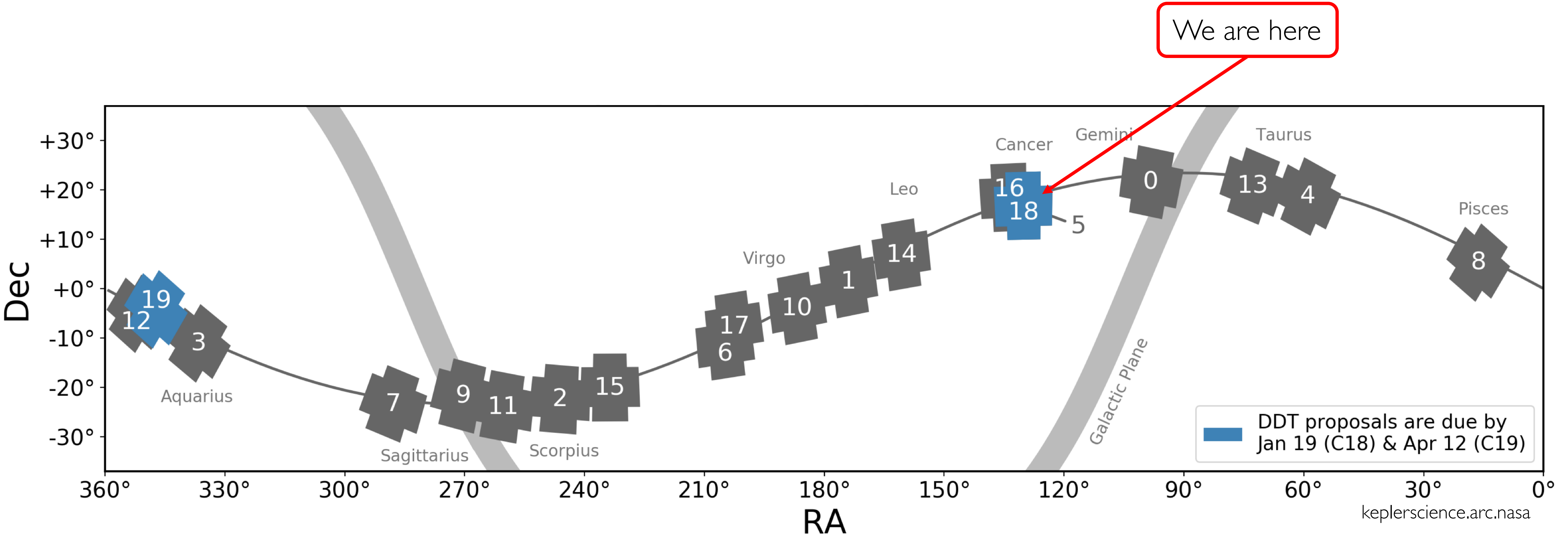


K2 Sputtering



K2 Sputtering

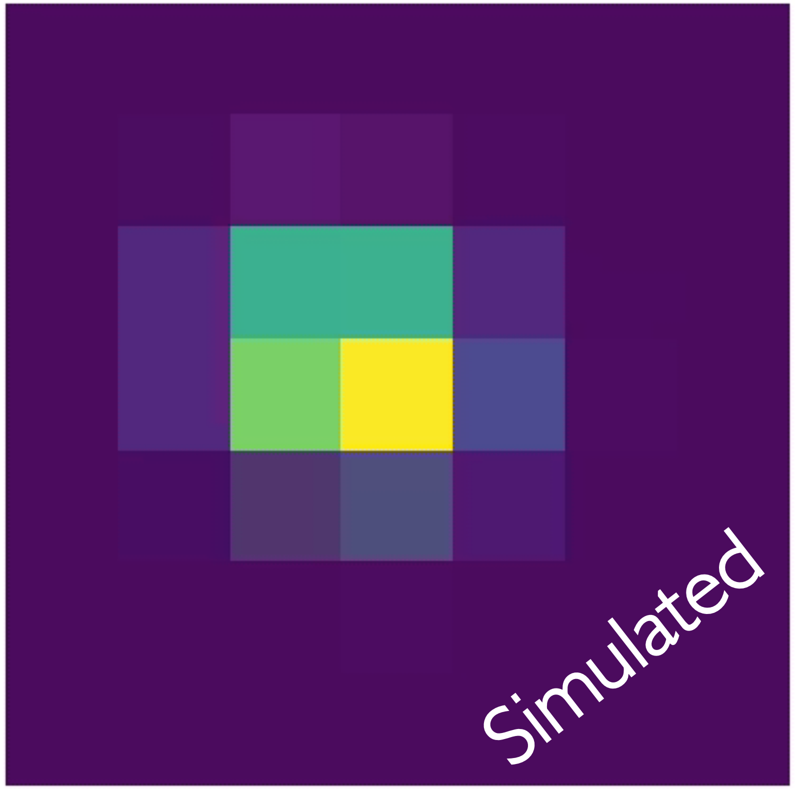
Spacecraft is running out of fuel → Motion could get worse

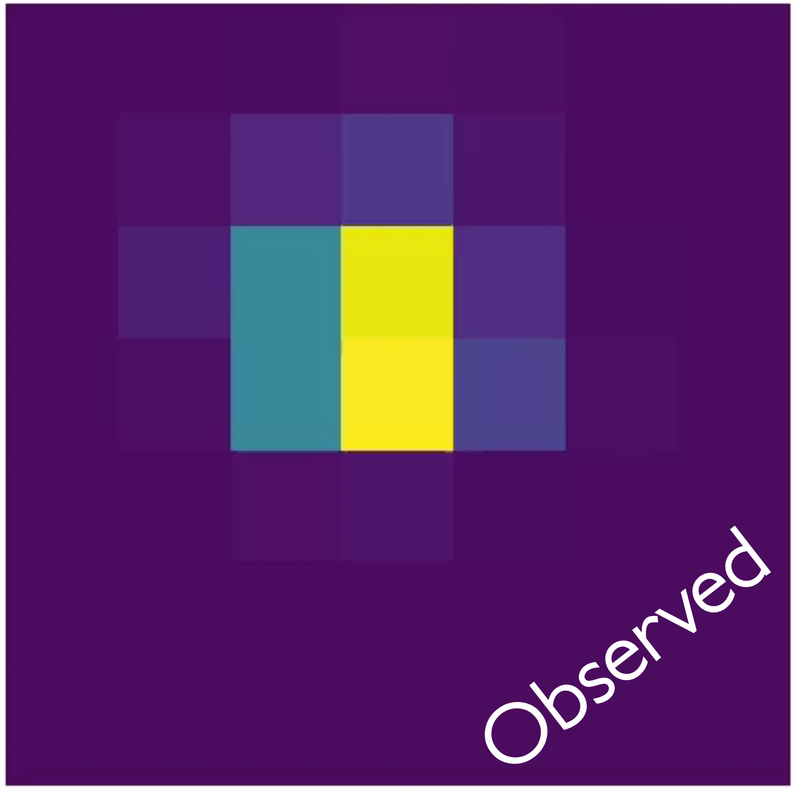


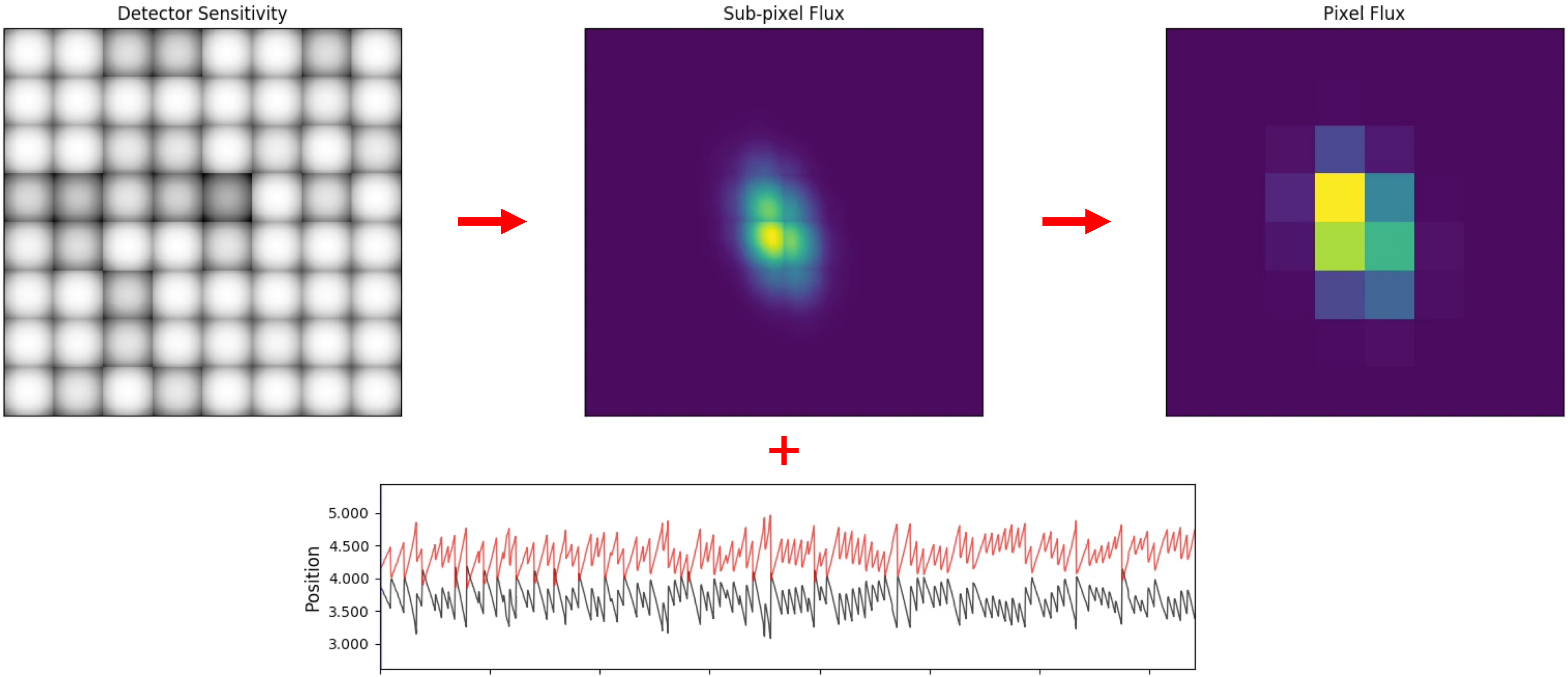
sputtering will cause higher magnitude and less predictable motion, creating more noise in *K2* light curves

I simulated high motion targets
to test noise-removal techniques

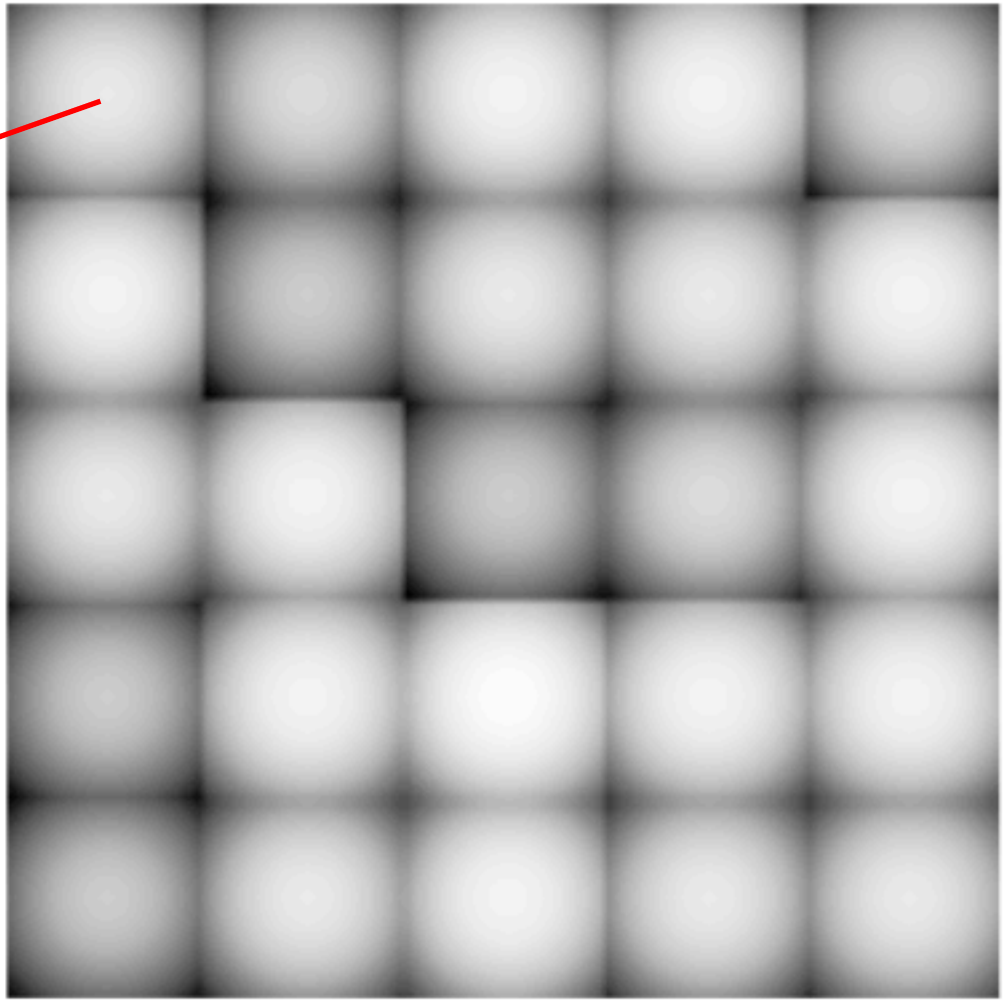
Which star is real?



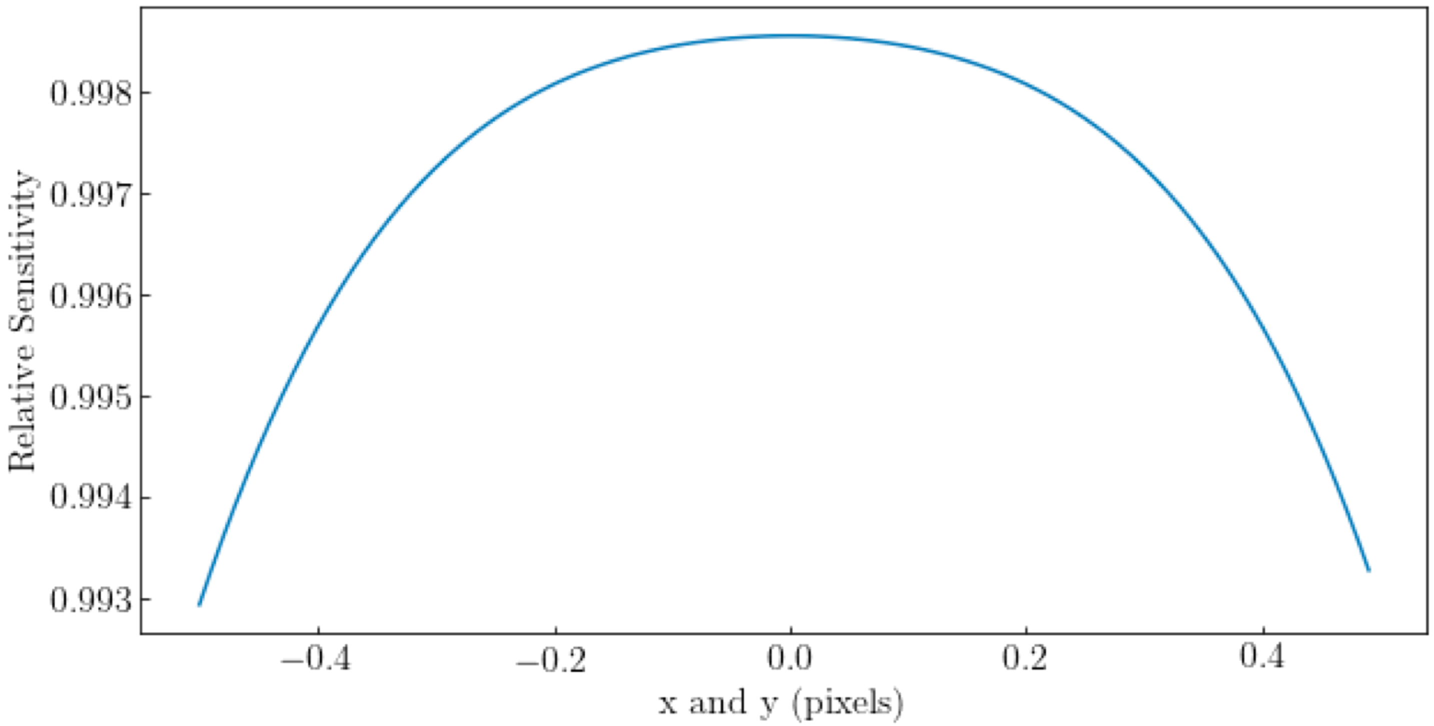


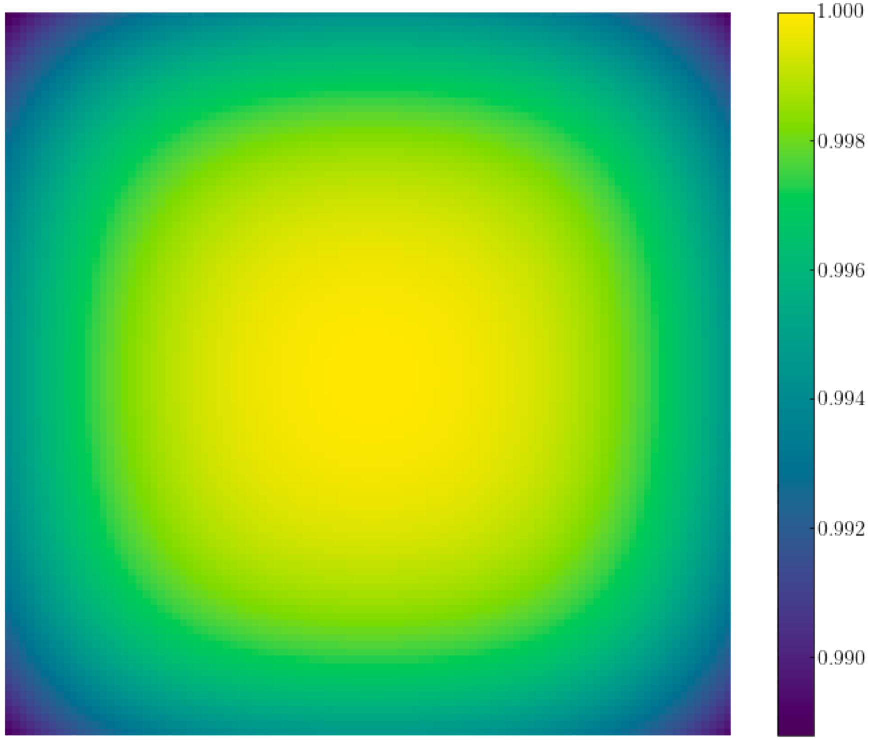

Detector Sensitivity
Sub-pixel Flux
Pixel Flux
+
Position
5.000
4.500
4.000
3.500
3.000

intra-pixel

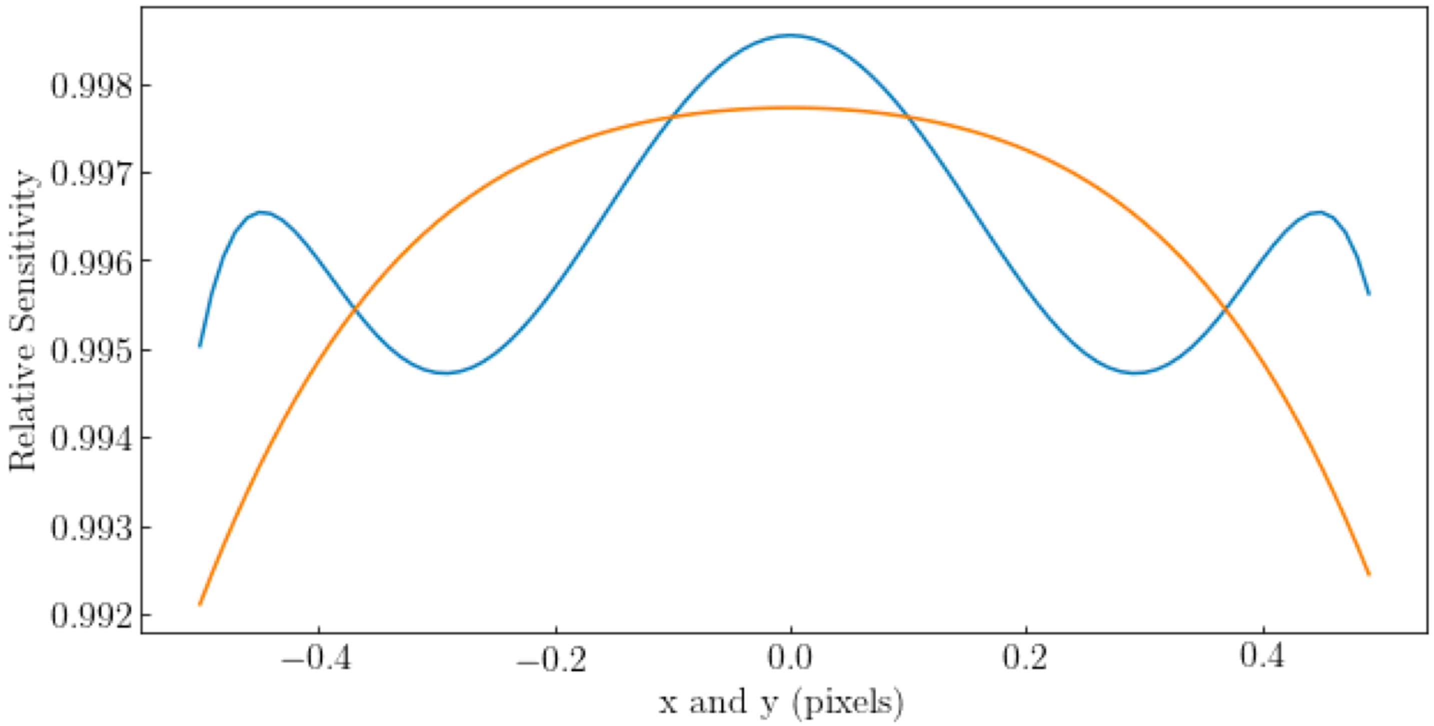


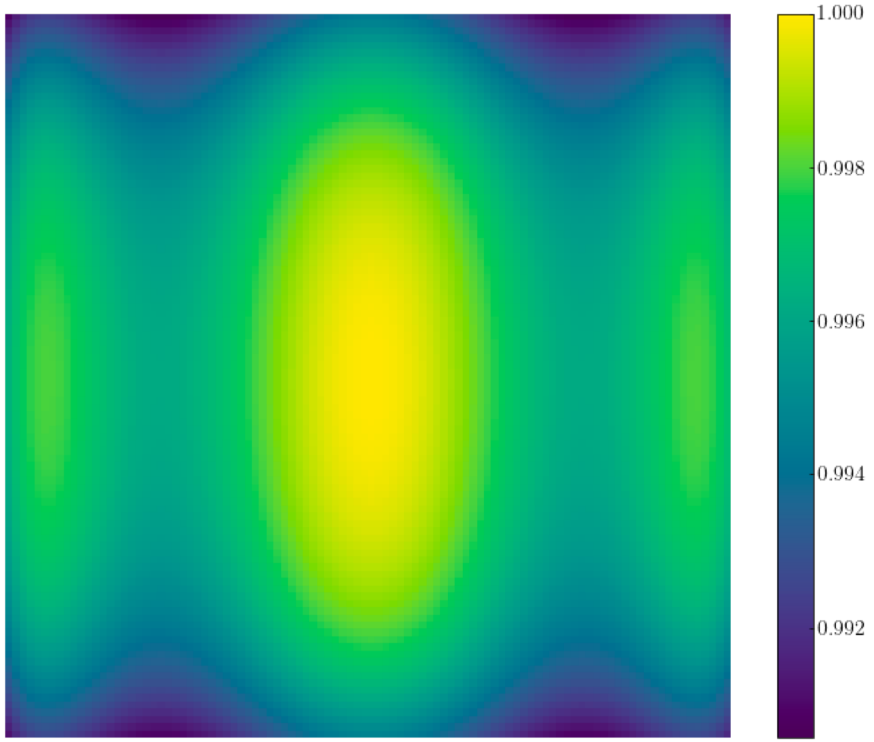
inter-pixel



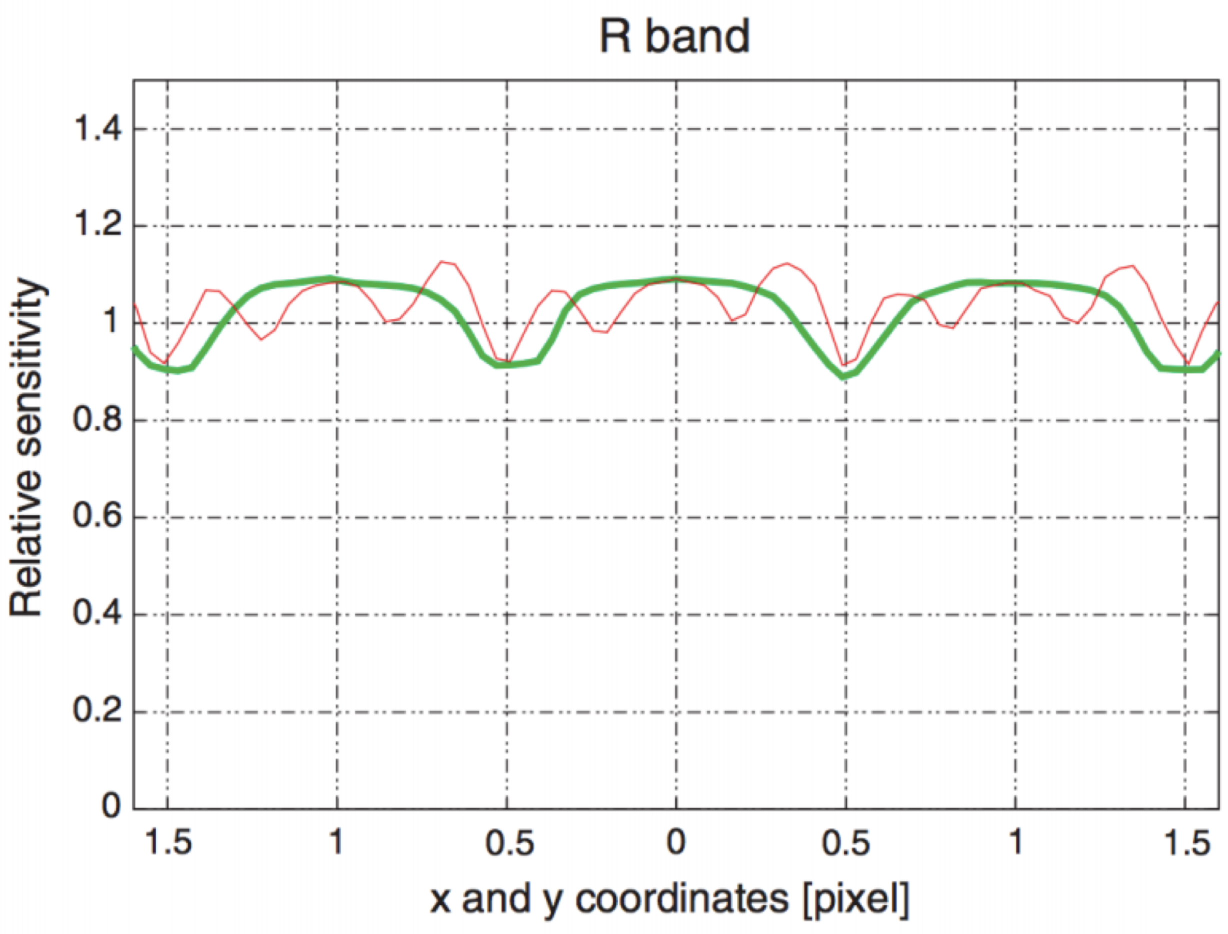


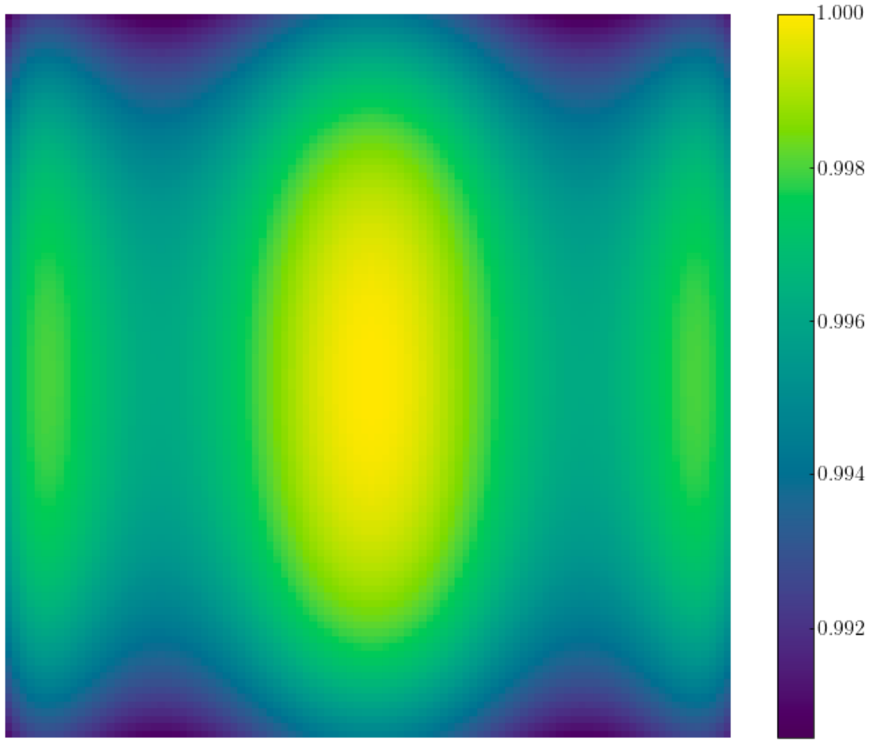
$$s(x,y) = \sum_{n,m} a_n x^n * b_m y^m$$



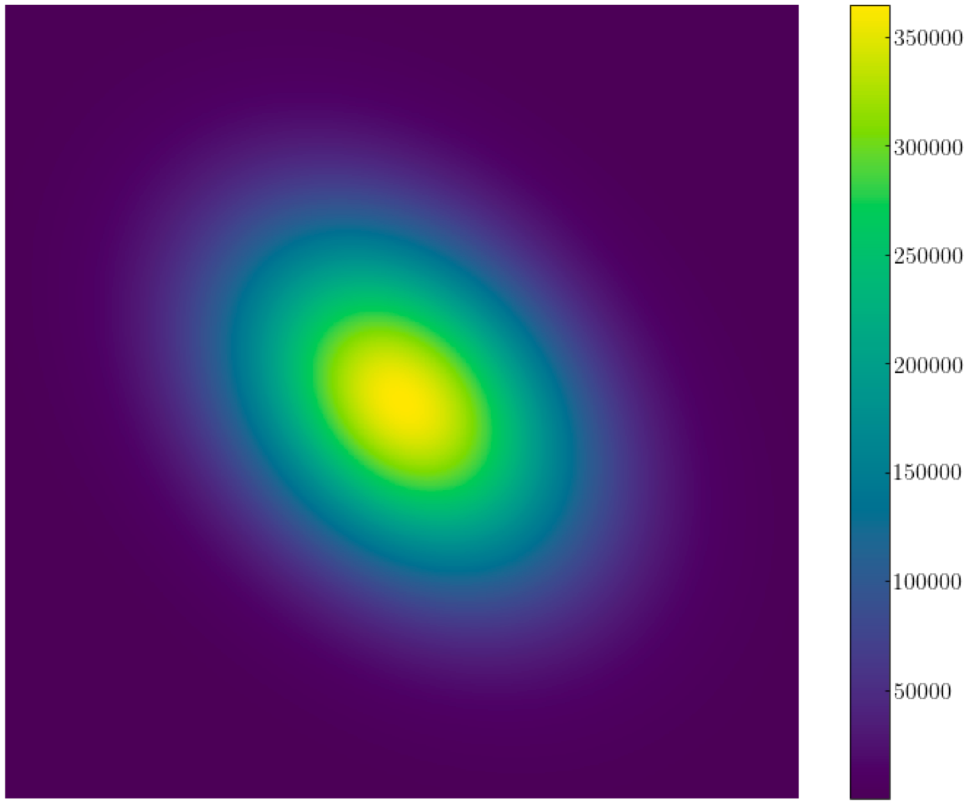


$$s(x,y) = \sum_{n,m} a_n x^n * b_m y^m$$

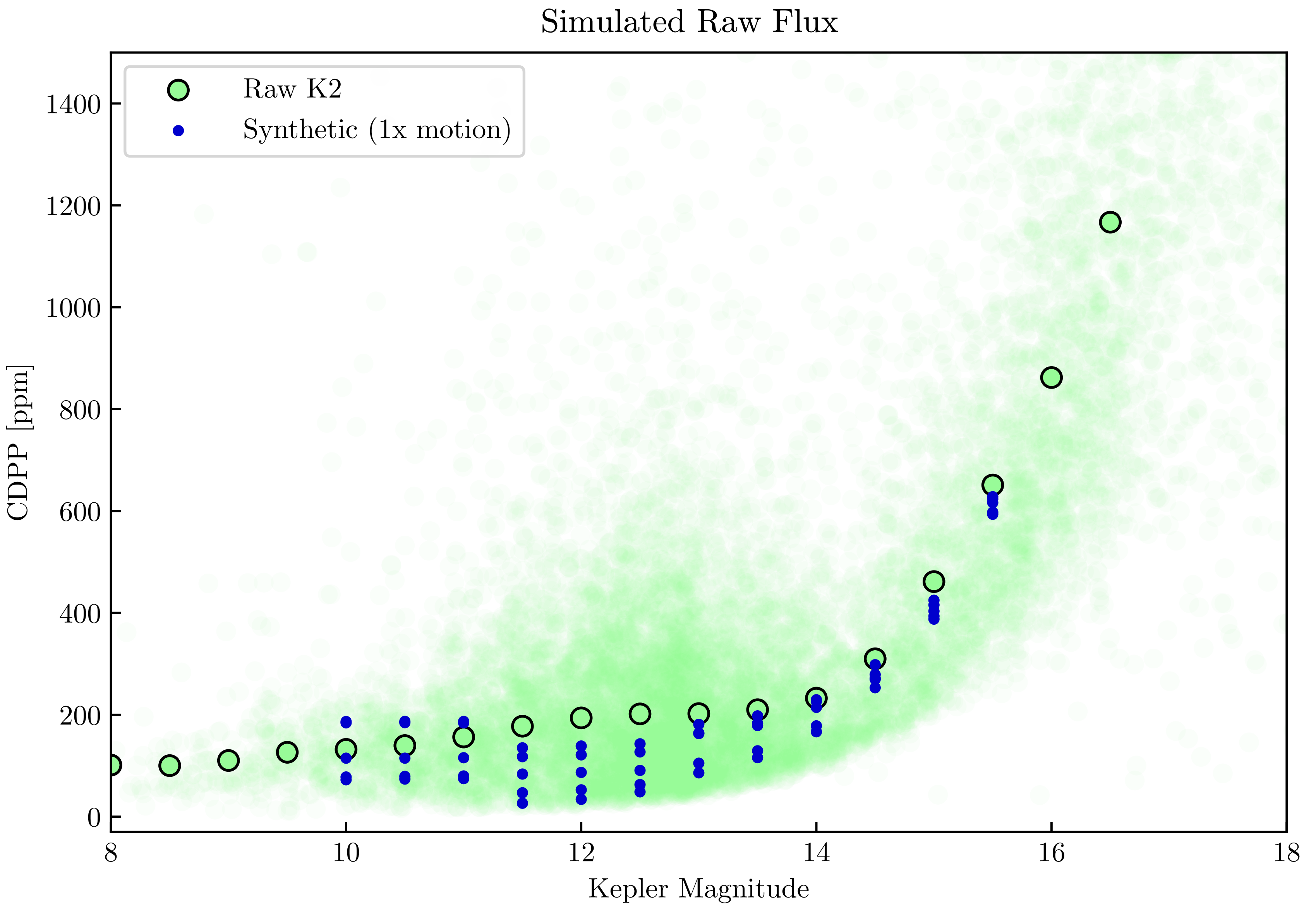


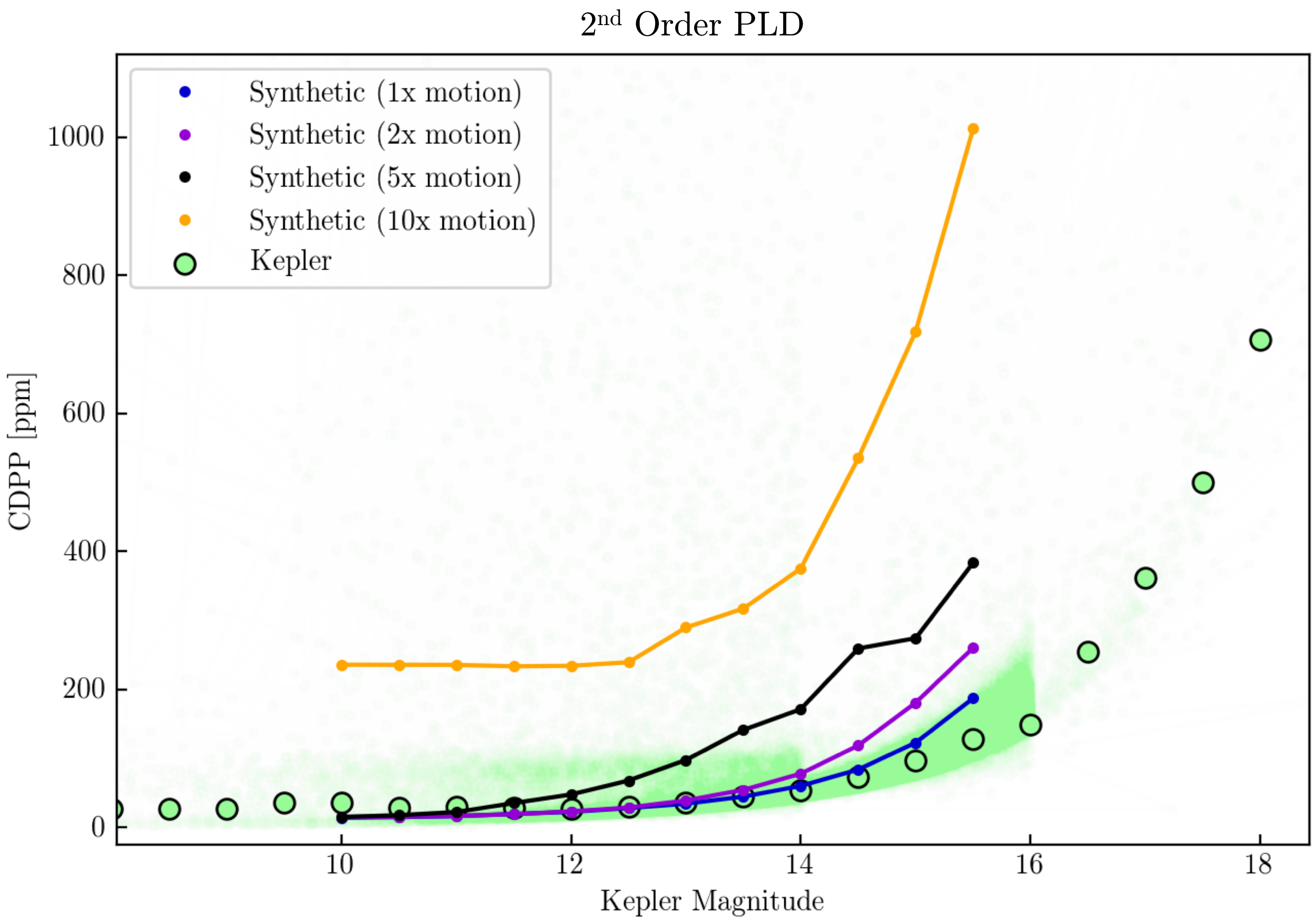


Toyozumi (2005)



Parameter	Description
A	PSF amplitude
σ_x	Standard deviation in x
σ_y	Standard deviation in y
$\rho_{x,y}$	Covariance term
(x_0, y_0)	Centroid position


Simulated Raw Flux
Raw K2
Synthetic (1x motion)
CDPP [ppm]
0
200
400
600
800
1000
1200
1400
8
10
12
14
16
18
Kepler Magnitude


2nd Order PLD
Synthetic (1x motion)
Synthetic (2x motion)
Synthetic (5x motion)
Synthetic (10x motion)
Kepler
CDPP [ppm]
0
200
400
600
800
1000
10
12
14
16
18
Kepler Magnitude

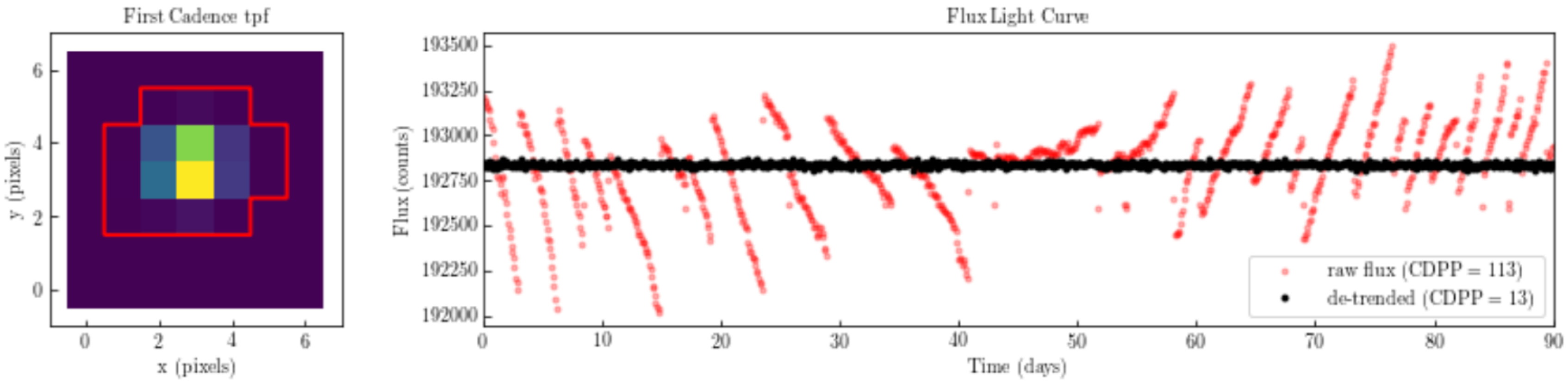
scope: Simulated CCD Observations for Photometric Experimentation

In [1]:
```
import scope

star = scope.Target()
tpf, flux, err = star.GenerateLightCurve()
```

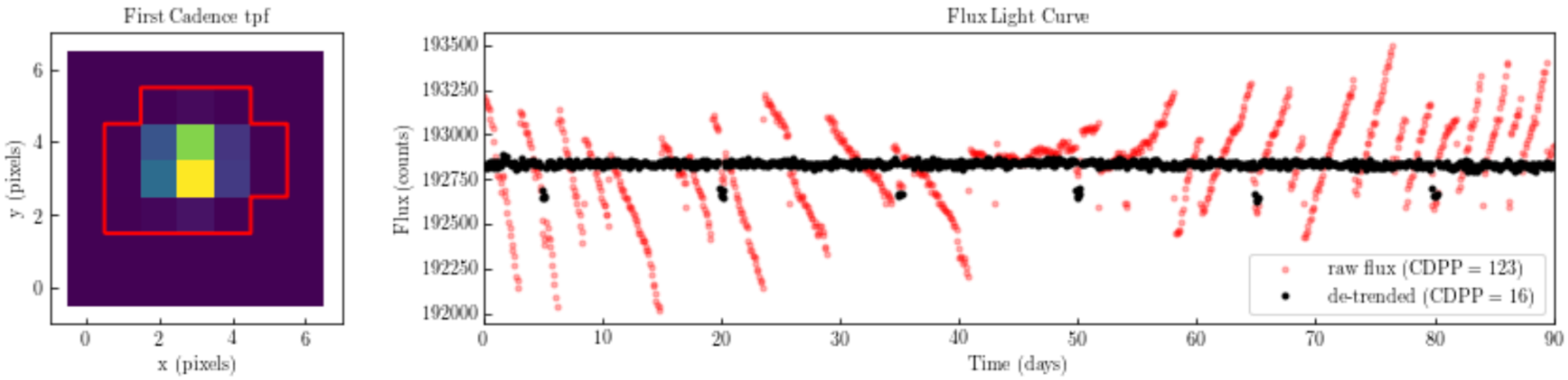
```
100%|██████████| 1000/1000 [01:56<00:00,  8.61it/s]
```

In [2]:
```
star.Plot()
```



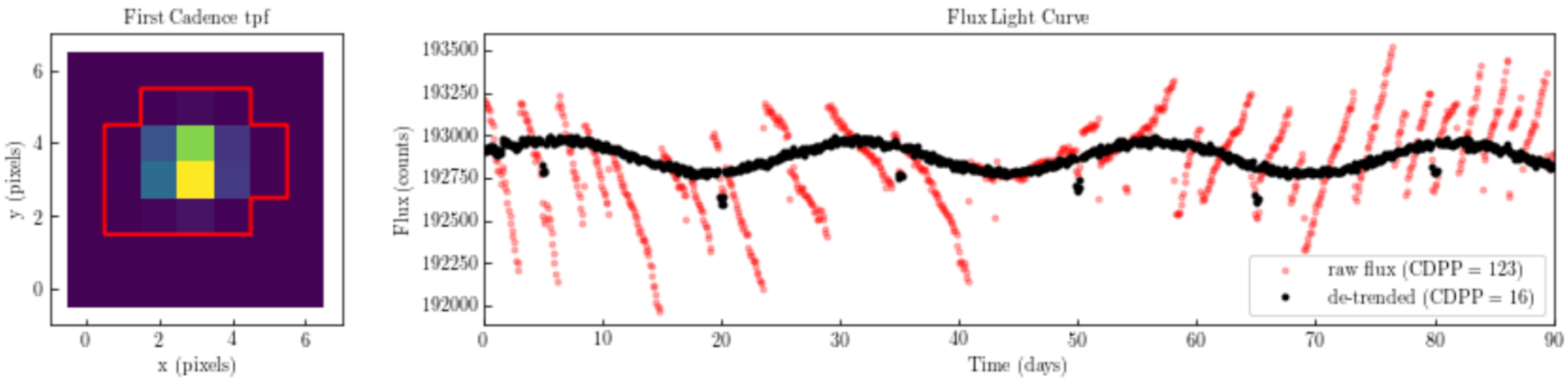
scope: Simulated CCD Observations for Photometric Experimentation

In [3]:
```
star.AddTransit()
star.Plot()
```



scope: Simulated CCD Observations for Photometric Experimentation

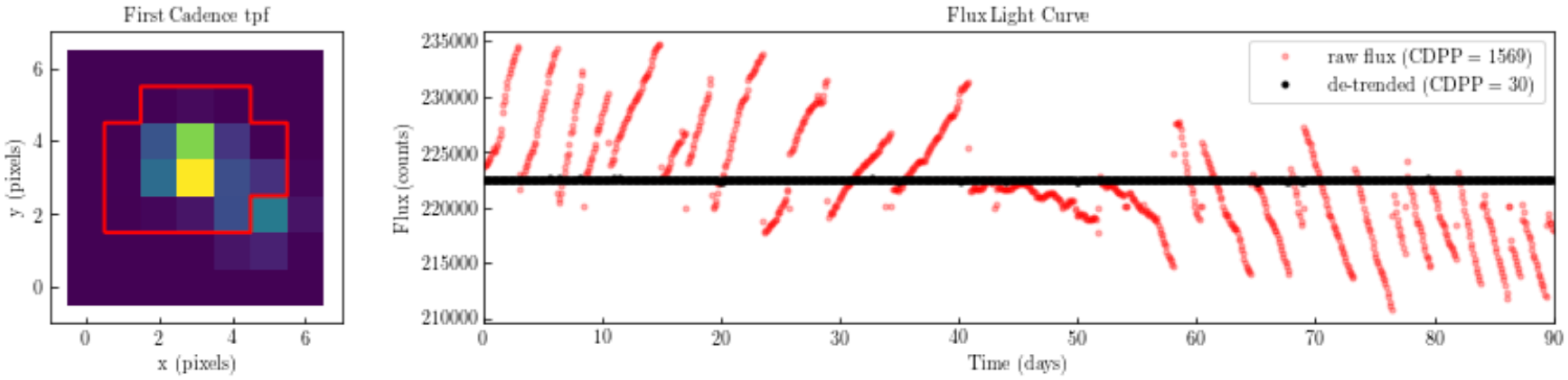
In [4]:
```
star.AddVariability()
star.Plot()
```



scope: Simulated CCD Observations for Photometric Experimentation

```
In [5]: star.AddNeighbor()
        star.Plot()
```

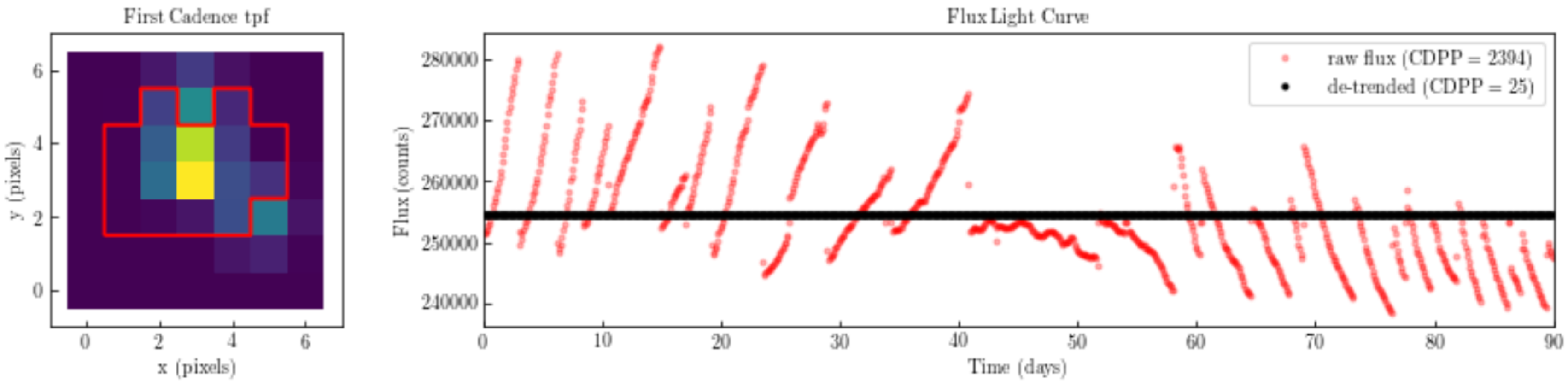
```
100%|██████████| 1000/1000 [02:00<00:00,  8.27it/s]
```



scope: Simulated CCD Observations for Photometric Experimentation

```
In [6]: star.AddNeighbor()
        star.Plot()
```

```
100%|██████████| 1000/1000 [02:04<00:00,  8.06it/s]
```



Current method:

PSF model (2D Gaussian)

$$\psi(x,y) = \sum_m b_m e^{\left[-\frac{1}{2(1-\rho_m^2)}\left(\frac{(x-x_{0,m})^2}{2\sigma_{x,m}^2}+\frac{(y-y_{0,m})^2}{2\sigma_{y,m}^2}-\frac{2\rho_m(x-x_{0,m})(y-y_{0,m})}{\sigma_{x,m}\sigma_{y,m}}\right)\right]}$$

$$b_m = \left(2\pi\sigma_{x,m}\sigma_{x,n}\sqrt{1-\rho_m^2}\right)^{-1}$$

Pixel sensitivity model

$$P(x,y) = \sum_{i,j} c_{ij} x^i y^j$$

Flux in k^{th} pixel

$$f_k(t) = s(t) \underbrace{\iint_{pixel} \psi(x,y)}_{\text{integral of Gaussian is an error function} \rightarrow \text{non-analytic} \rightarrow \text{slow to solve}} P(x,y)dxdy$$

Polynomial basis:

PSF model

$$\psi(x,y) = \sum_{i,j} b_{ij}(x-x_0)^i(y-y_0)^j$$

Pixel sensitivity model

$$P(x,y) = \sum_{i,j} c_{ij}x^i y^j$$

Flux in $\mathrm{k^{th}}$ pixel

$$f_k(t) = s(t)\underbrace{\iint_{pixel} \psi(x,y)P(x,y)dxdy}$$

analytic! → only need to solve once → <u>much</u> faster

$$\boxed{f_k(t) = s(t)\sum_{i,j} d^k_{ij} x^i_0(t) y^j_0(t)}$$

$$d^k_{ij}(b,c)$$

But wait, there's more…

$$f_k(t) = s(t) \sum_{i,j} d_{ij}^k x_0^i(t) y_0^j(t)$$

Faster — Skip numerical integration

More accurate — Not limited to Gaussian PSF fits

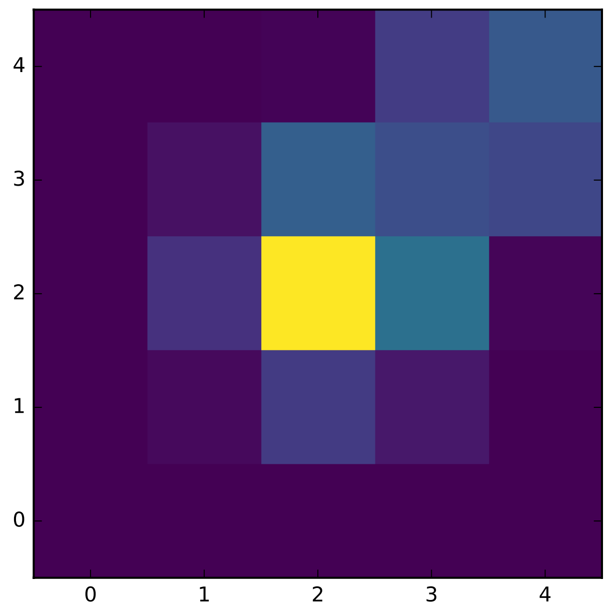
Improves PLD — Polynomial framework acts as prior on structure of the design matrix

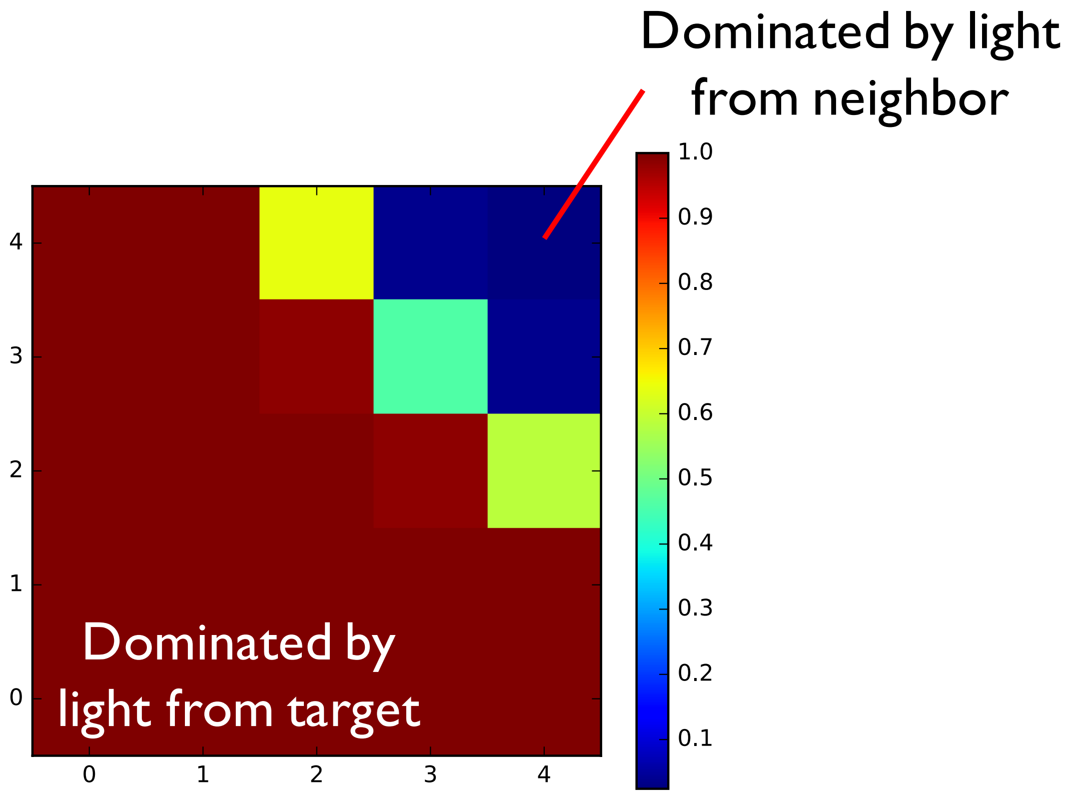
$\rightarrow$ simpler de-trending process, only depends on motion of $(\mathrm{x}_0,\ \mathrm{y}_0)$

Bayesian PLD?

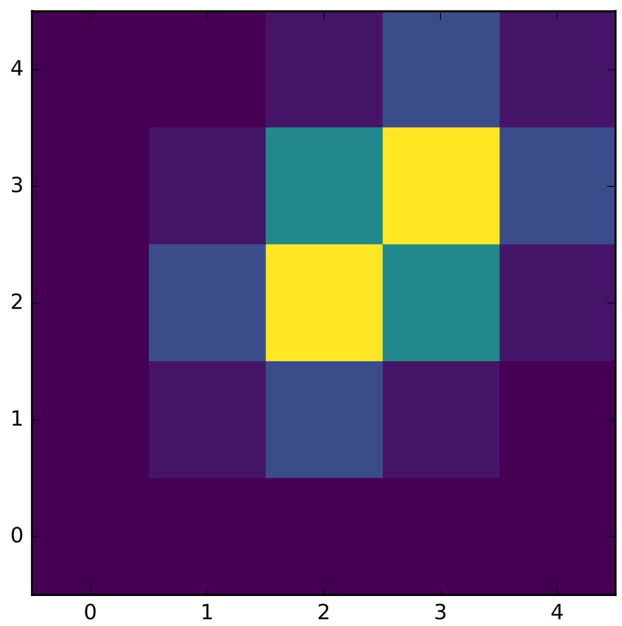
* Work in progress *

Crowded Fields

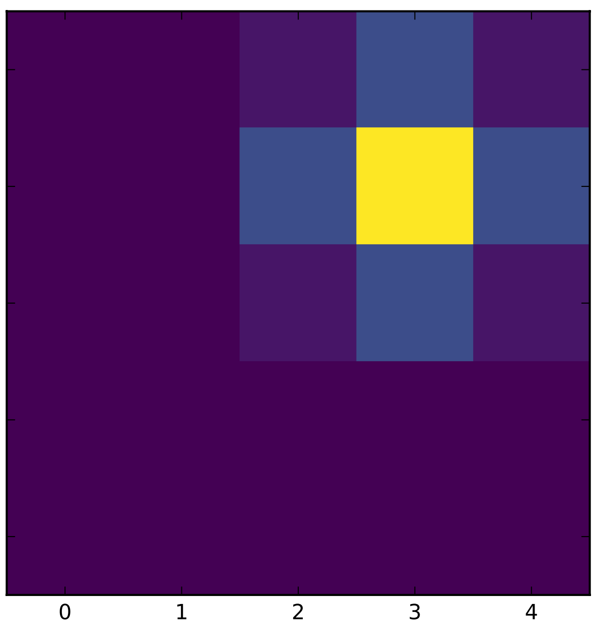




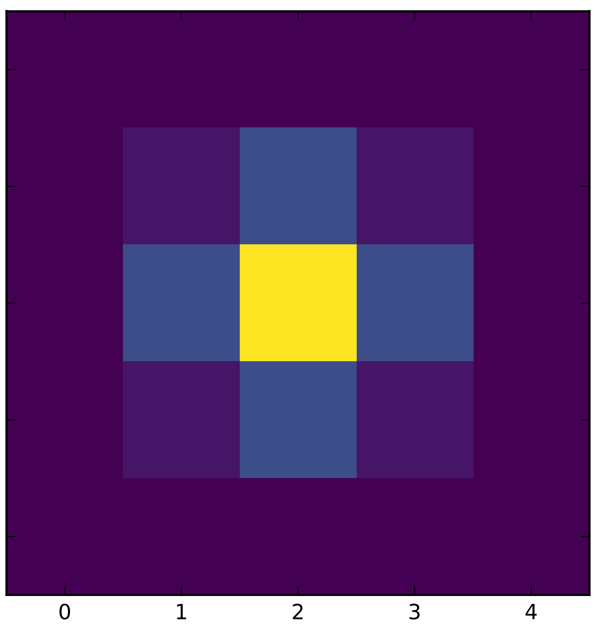
PSF Fitting



Fit PSF model to all stars in aperture

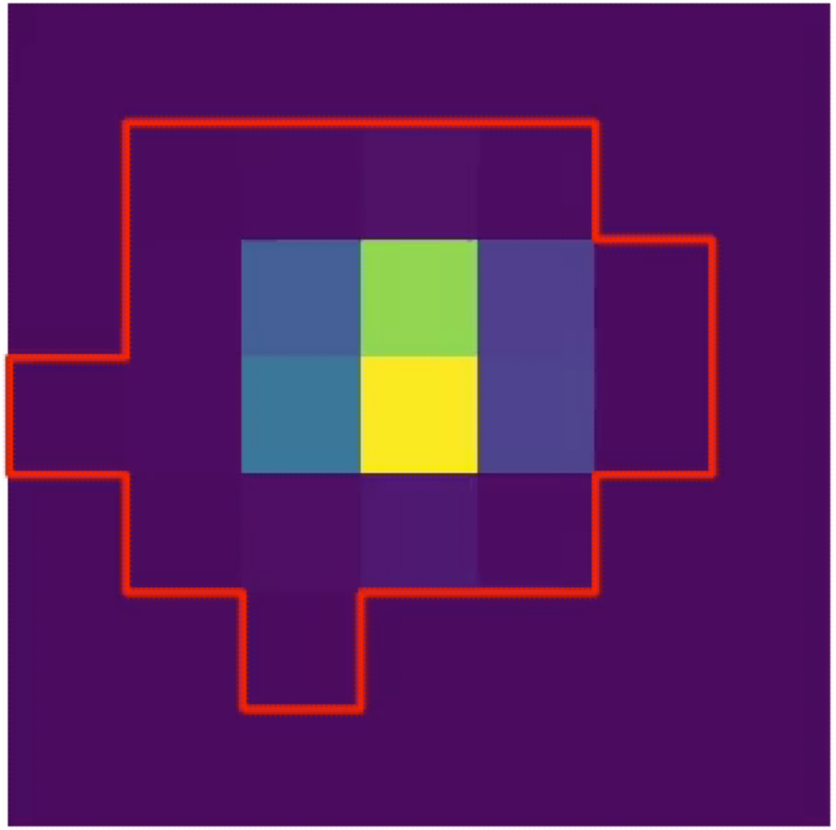


Isolate flux contribution from neighbors

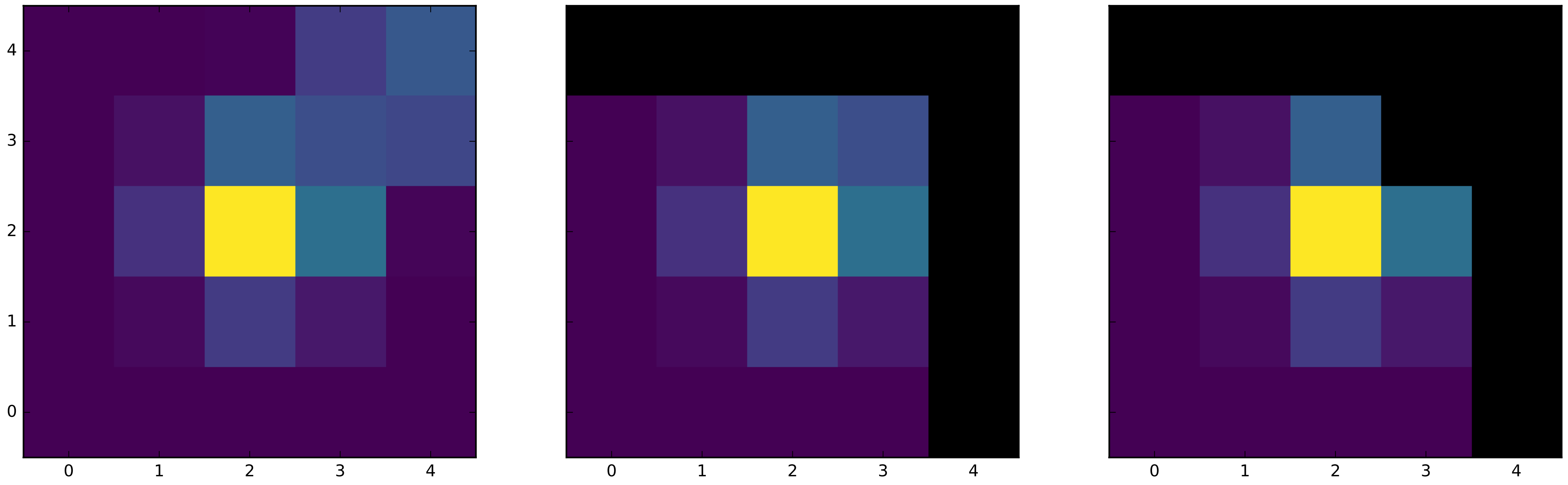


Subtract neighbor flux

Aperture Fitting

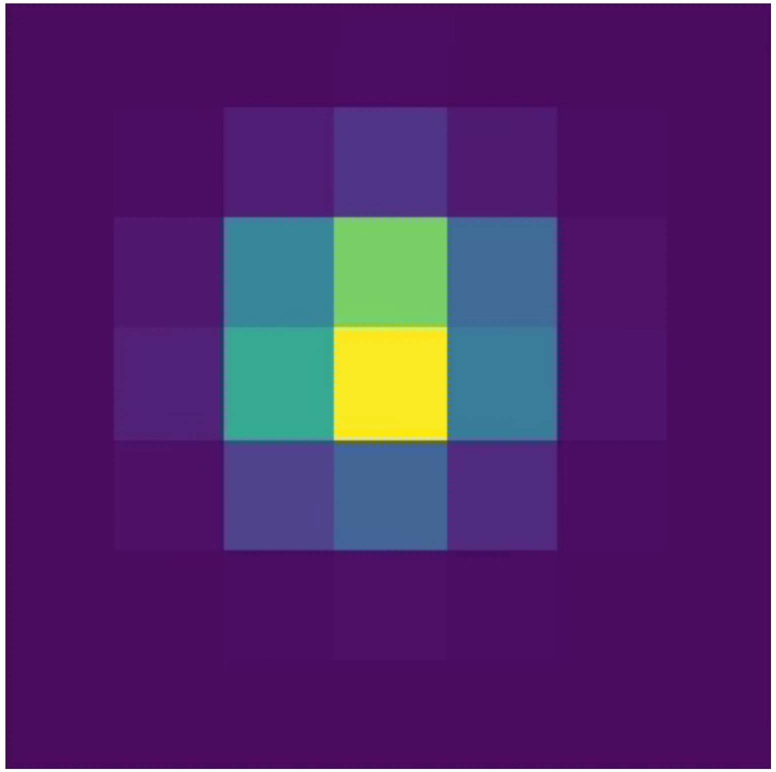


Aperture Fitting



Recovered Transit: (% of true depth) 67% 88% 98%

"Breathing" PSFs



Or when TESS's reaction wheels break

Thank you!

Dr. Rodrigo Luger & Dr. Rory Barnes
University of Washington Virtual Planetary Laboratory
NASA Grant NNX14AK26G