Inference of the K2 PSF

Michael Gully-Santiago[1,2]

[1] *Kepler/K2 Guest Observer Office*
NASA Ames Research Center
Moffett Blvd
Mountain View, CA 94035, USA
[2] *Bay Area Environmental Reaearch Institute*

Keywords: methods: statistical, techniques: image processing, methods: data analysis

We infer PSFs in Kepler data. Here is a citation: Olling et al. (2015).

PSF PHOTOMETRY FOR KEPLER

Astronomical photometry has conventionally followed two paths: aperture photometry or Point Spread Function (PSF) photometry. Point spread function photometry excels in crowded regions and in low Signal-to-Noise (S/N) ratio regimes. Kepler/K2 mission time-series imaging of star clusters and extended objects will benefit from PSF photometry. The large computational cost of PSF photometry has hampered its application to crowded fields in Kepler/K2, slowing the unbiased analysis of existing valuable data. In this research note, we lay the foundation for the application of GPU-acceleration applied to PSF photometry, demonstrating a speed-up over conventional computational methods. We apply our method to both synthetic data, recovering unbiased estimates of our inputs, and real K2 data. In the latter case we derive the Kepler PSF itself, the individual fluxes of stars, and the covariance among all inferred stars. Recent advances in GPU acceleration hardware and associated software libraries offer avenues for performance improvements in other domains in astrophysical data analysis.

Corresponding author: Michael Gully-Santiago
igully@gmail.com

METHODOLOGY

We make these basic assumptions about our problem structure:

1. All sources are point sources.
2. There are exactly $N_\star$ sources.
3. All sources share the same PSF.
4. All pixels share the same response function.
5. The PSF is drawn from a Gaussian Mixture Model with $N_{GMM} = 3$ unknown components.
6. Each source has 3 parameters– $x_{\star,i}, y_{\star,i}$ center locations and flux amplitude, f_i.
7. Each PSF component has 5 parameters– relative (x_j, y_j) location; amplitude z_j; and bivariate covariance matrix C_j, which is comprised of 3 parameters (ϕ_a, ϕ_b, ϕ_c).
8. No PSFs are saturated.
9. All pixels have known, homoscedastic Gaussian read noise σ_r.
10. The flat field is uniform, or equivalently has already been corrected perfectly.
11. There are N_{pix} pixels in the scene.

We fix the first Gaussian Mixture Model components $x_1, y_1 \equiv 0, 0$, enforce $z_{j+1} < z_j$, and enforce normalization $\int \mathcal{K}\, dx\, dy = 1$. The above assumptions yield a total number of parameters, $N_{\mathrm{param}} = 3N_\star + 6N_{GMM} - 3$.

We identified two nearly identical procedures for calculating the likelihood. In procedure I, we instantiate $N_\star$ delta functions in a sub-sampled pixel grid, convolve a model PSF with the delta functions, and sum the sub-sampled pixels in a Kepler Pixel. In procedure II, instantiate $N_\star$ model PSFs, evaluating the model at each pixel center. We anticipate that these two procedures produce the same results, but procedure II is likely to possess smoother derivatives than procedure I, resulting in higher numerical performance. Procedure II can only be carried out assuming statement number 4 above holds. We adopted procedure II. The source of the PSFs can be either source detection in the local image (*e.g. Source Extractor*), or an outside deep catalog in a similar bandpass (*e.g. Gaia*). We performed source detection for the current note.

The models are:

$$\mathcal{K}_{\mathrm{PSF}} = \sum_{j=0}^{N_{\mathrm{GMM}}} z_j \mathcal{G}(x - x_j, y - y_j, C_j) \tag{1}$$

$$\hat{I}(x, y) = \sum_{i}^{N_\star} f_i \cdot \mathcal{K}_{\mathrm{PSF}}(x - x_{\star,i}, y - y_{\star,i}) \tag{2}$$

We can rewrite the sums as matrix products. We assign d as the $N_{pix} \times 1$ column vector mapping all data pixel values from the 2D image onto a 1D vector. The vector a is the 2D PSF model for the i^{th} star mapped onto an $N_{pix} \times 1$ one-dimensional column vector. The matrix A represents the column-wise concatenation of the $N_\star$ models a_i, yielding an $N_{pix} \times N_\star$ matrix. The vector f contains the $N_\star$ PSF flux amplitudes f_i. We can analytically compute the "profile likelihood" flux amplitudes:

$$\mathsf{f}^* = (\mathsf{A}^\mathsf{T} \mathsf{C}^{-1} \mathsf{A})^{-1} (\mathsf{A}^\mathsf{T} \mathsf{C}^{-1} \mathsf{d}) \tag{3}$$

Computing the f^* analytically eliminates $N_\star$ linear parameters (the f_i's) from the N_{param} parameters, and delivers the covariance among the inferred amplitudes. We derive the non-linear parameters by minimizing χ^2:

$$\mathsf{m} = \mathsf{A} \cdot \mathsf{f}^* \tag{4}$$
$$\mathsf{r} = \mathsf{d} - \mathsf{m} \tag{5}$$
$$\ln p(\mathsf{d}|\mathsf{m}) = -\frac{1}{2}\left(\mathsf{r}^{\mathsf{T}}\mathsf{C}^{-1}\mathsf{r} + \ln\det\mathsf{C} + N_{pix}\ln 2\pi\right) \tag{6}$$

where C is the pixel noise covariance matrix. For our current assumption of homoscedastic read noise $\mathsf{C} = \sigma_r \mathcal{I}$ where $\mathcal{I}$ is the diagonal identity matrix.

APPLICATION- PSF PHOTOMETRY ON SYNTHETIC DATA

- Application- PSF photometry
- Synthetic data generation
- Application to synthetic data
- (Stretch goal): Application to real patch of FFI data
- (Stretch goal): Performance with CPU
- (Stretch goal): Performance with GPU

APPENDIX

Here we clarify the notation and offer additional mathematical transformations used in this note. The normalized Gaussian mixture model takes the form:

$$\mathcal{G} = \frac{1}{2\pi\sqrt{\det C_j}} e^{-\frac{1}{2}\vec{r}^{\mathsf{T}} C_j^{-1}\vec{r}} \tag{7}$$

with $\vec{r} = \vec{x} + \vec{y}$ representing the distance vector. The Kernel and data positions refer to their respective reference frames, with Kernels representing continuous variables and data spaces representing pixel coordinates. The bivariate Gaussian mixture model covariance matrix can be written down as:

$$C_j = \begin{bmatrix} e^{2\phi_a} & \phi_c e^{\phi_a} \\ \phi_c e^{\phi_a} & \phi_c^2 + e^{2\phi_b} \end{bmatrix} \tag{8}$$

which enforces positive scale factors and positive semidefinite matrices. Enforcing normalization $\int \mathcal{K}\, dx\, dy \equiv 1$ results in $\sum_{j=0}^{N_{\rm GMM}} z_j = 1$. We apply several transformations for computational expediency:

$$z_j - z_{j-1} = e^{-\theta_j} \tag{9}$$
$$z_1 = \frac{1}{1 + \sum_{j=2}^{N_{\rm GMM}} z_j} \tag{10}$$
$$\theta_j > 0 \tag{11}$$

This research has made use of NASA's Astrophysics Data System. The reproducible Jupyter Notebook that generated the figures in this document are freely available on GitHub.

REFERENCES

Olling, R. P., Mushotzky, R., Shaya, E. J., et al. 2015, Nature, 521, 332