Fitting the PSF

Clare Saunders

This document describes the method used to fit the PSF in the LPNHE pipeline for Subaru HSC data, but should be usable for any data in the required format. The associated code assumes a data file “calibrated.fz”, weight file “weight.fz”, and list of point sources “standalone_stars.list”. A mask of saturated or otherwise unusable pixels can be provided as the file “satur.fits.gz” (format assumes 1 = bad, 0 = good).

We assume a PSF with model

$$\Psi = (1 - c)\phi + R \tag{1}$$

where ϕ is an elliptical Moffat PSF and R is a polynomial of the form

$$R_{i,j} = R^0_{i,j} + R^1_{i,j}x + R^2_{i,j}y + ... \tag{2}$$

for a given pixel position i, j on the vignette centered on a given point source (an isolated star) and CCD position x and y. This model is chosen to maximize the amount of flux explained by an analytic function (here an elliptical Moffat), with relatively small corrections made be the discreet grid of R. Because the data grid (tied to the pixels) and the model grid (centered on the object) are different, R and ϕ should actually be R_s and ϕ_s, where s is for star s, but I suppress the subscripts for simplicity. ϕ at any position can be calculated analytically. Resampling R onto the data grid is discussed below.

In each of the following steps, the fit parameters are found using the multivariate Newton-Raphson method, by which

$$\mathbf{H}(\chi^2)\delta\mathbf{x} = \frac{d\chi^2}{d\mathbf{x}} \tag{3}$$

is solved using Cholesky decomposition in order to find the minimum of χ^2 as a function of $\mathbf{x}$. For non-linear problems, this process is iterated, with a Brent line-search after each Cholesky solution to find the minimum of χ^2 in the direction of $\delta\mathbf{x}$, which minimizes the steps necessary and also prevents taking a too-large step and getting a situation where $\chi^2 = \text{NAN}$.

First, the analytic part of the PSF is fit. This is done by first fitting an elliptical Moffat to each star, then fitting a degree-one (linear) polynomial function to the Moffat PSF parameters as a function of the CCD position.

For an individual star, we fit

$$d_{s,i,j} = f_s * \phi_{i,j} \tag{4}$$

by minimizing

$$\chi^2 = W(d_{s,i,j} - f_s * \phi_{s,i,j})^2 \tag{5}$$

where $$\phi_{i,j} = \frac{1}{N}(1+w_{xx,s}(x_i-x_{c,s})^2+w_{yy,s}(y_j-y_{c,s})^2+2w_{xy,s}(x_i-x_{c,s})(y_j-y_{c,s}))^{-2.5} \tag{6}$$

$$N = \frac{2.5-1}{\pi}(w_{xx,s}w_{yy,s} - w_{xy,s}^2)^{0.5} \tag{7}$$

and $d_{s,i,j}$ is the data for star s at pixel i, j. The fit parameters are the flux of the star, f_s, the star position, $x_{c,s}$ and $y_{c,s}$, and the Moffat PSF parameters, $w_{xx,s}$, $w_{yy,s}$, and $w_{xy,s}$.

Then, using the Moffat PSF parameters for the individual stars, we fit a global function to minimize:

$$\chi^2 = \sum_s W(w_{xx}(x_{c,s}, y_{c,s}) - w_{xx,s})^2 \quad \text{and similar equations for } w_{yy}, w_{xy} \tag{8}$$

where

$$w_{xx}(x_{c,s}, y_{c,s}) = w_{xx,0} + w_{xx,x}\ x_{c,s} + w_{xx,y}\ y_{c,s} \tag{9}$$

$$w_{yy}(x_{c,s}, y_{c,s}) = w_{yy,0} + w_{yy,x}\ x_{c,s} + w_{yy,y}\ y_{c,s} \tag{10}$$

$$w_{xy}(x_{c,s}, y_{c,s}) = w_{xy,0} + w_{xy,x}\ x_{c,s} + w_{xy,y}\ y_{c,s} \tag{11}$$

Solving this problem is linear, but we toss out 4 sigma outliers and refit the parameters until there are no more outliers. At the end, we check that the PSF is okay (i.e. the norm is a real number) at the corners of the CCD. If not, the algorithm exits and the CCD is considered a failure (until we improve the algorithm to prevent this situation).

At this point, the analytic part of the PSF is done and we can fit whatever systematic behavior remains.

Given data d, we want to minimize the equation

$$\chi^2 = \sum_{i,j,s} W_{i,j,s}(d_{i,j,s} - f_s\Psi)^2 = \sum_{i,j,s} W_{i,j,s}[d_{i,j,s} - f_s((1-c)\phi + R)]^2 \tag{12}$$

We also want to enforce the condition that $\sum_{i,j}\Psi = 1$. Given the fact that ϕ is normalized, this will be true if $\sum_{i,j} R = c$, since we will have

$$\sum_{i,j}\Psi = \sum_{i,j}[(1-c)\phi + R] = 1 - c + c = 1 \tag{13}$$

To enforce this condition on the χ^2 minimization, i.e. $0 = \sum_{i,j} R_{p,i,j} - c_p$, we use Lagrange multipliers and form the Lagrangian

$$\mathcal{L} = \left\{\sum_{i,j,s} W_{i,j,s}[d_{i,j,s} - f_s((1-c)\phi + R)]^2\right\} - \sum_p \lambda_p(\sum_{i,j} R_{p,i,j} - c_p) \tag{14}$$

where there is a λ term for each term of the polynomial R. Then this is the equation that we want to minimize as a function of f_s, c_p, $R_{p,i,j}$ and λ_p.

To do this, we need to calculate the partial derivatives and the Hessian of $\mathcal{L}$.

First the partial derivatives:

$$\frac{\partial \mathcal{L}}{\partial f_s} = -2\sum_{i,j} W_{i,j,s}[d_{i,j,s} - f_s((1-c)\phi + R)]((1-c)\phi + R) \tag{15}$$

$$\frac{\partial \mathcal{L}}{\partial R_{p,i,j}} = \left\{ -2\sum_{s} W_{i,j,s}[d_{i,j,s} - f_s((1-c)\phi + R)]f_s \frac{\partial R}{\partial R_{p,i,j}} \right\} - \lambda_p \tag{16}$$

$$\frac{\partial \mathcal{L}}{\partial c_p} = \left\{ 2\sum_{s} W_{i,j,s}[d_{i,j,s} - f_s((1-c)\phi + R)]f_s \phi \frac{\partial c}{\partial c_{p'}} \right\} + \lambda_p \tag{17}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda_p} = \sum_{i,j} R_{p,i,j} - c_p \tag{18}$$

Next, the Hessian matrix. This shows the full math, in practice all second derivative terms are dropped, which makes the minimization algorithm more robust (see discussion in Numerical Recipes). I only show the diagonal blocks and upper triangle of the matrix:

$$\frac{\partial^2 \mathcal{L}}{\partial^2 f_s} = 2\sum_{i,j} W_{i,j,s}[(1-c)\phi + R]^2 \tag{19}$$

(Non-diagonal terms are all zero.)

$$\frac{\partial^2 \mathcal{L}}{\partial R_{p,i,j} \partial f_s} = \left[-2\sum_{i,j} W_{i,j,s}[d_{i,j,s} - 2f_s((1-c)\phi + R)] \right] \frac{\partial R}{\partial R_{p,i,j}} \tag{20}$$

$$\frac{\partial^2 \mathcal{L}}{\partial c_p \partial f_s} = \sum_{i,j} \left[2W_{i,j,s} d_{i,j,s} \phi - 4W_{i,j,s} f_s((1-c)\phi + R)\phi \right] \frac{\partial c}{\partial c_p} \tag{21}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \lambda_p \partial f_s} = 0 \tag{22}$$

$$\frac{\partial^2 \mathcal{L}}{\partial R_{p',i',j'} \partial R_{p,i,j}} = 2\sum_{s,i,j} W_{i,j,s} f_s^2 \frac{\partial R}{\partial R_{p',i',j'}} \frac{\partial R}{\partial R_{p,i,j}} \tag{23}$$

$$\frac{\partial^2 \mathcal{L}}{\partial c_{p'} \partial R_{p,i,j}} = -2\sum_{s,i,j} W_{i,j,s} f_s^2 \phi \frac{\partial c}{\partial c_{p'}} \frac{\partial R}{\partial R_{p,i,j}} \tag{24}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \lambda_{p'} \partial R_{p,i,j}} = -\delta p, p' \tag{25}$$

$$\frac{\partial^2 \mathcal{L}}{\partial c_{p'} \partial c_p} = 2 \sum_{s,i,j} W_{i,j,s} f_s^2 \phi^2 \frac{\partial c}{\partial c_{p'}} \frac{\partial c}{\partial c_p} \tag{26}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \lambda_{p'} \partial c_p} = \delta p, p' \tag{27}$$

$$\frac{\partial^2 \mathcal{L}}{\partial \lambda_{p'} \partial \lambda_p} = 0 \tag{28}$$

For convenience, I define a vector $\mathbf{p}$ to facilitate expressions involving polynomials:

$$\mathbf{p} = \big(1, \quad x, \quad y, \quad x^2, \quad y^2, \quad xy, \quad ...\big) \tag{29}$$

The full form for c can be expressed as:

$$c = \mathbf{c} \cdot \mathbf{p} \tag{30}$$

where $\mathbf{c}$ is the vector of c_p values.

R has to also be convolved onto the pixel grid of the data, so R is actually R_s. This operation is performed by multiplying by a matrix $\mathbf{C_s}$. Thus

$$R_s = \mathbf{C_s} \cdot \mathbf{R} \cdot \mathbf{p} \tag{31}$$

where, similarly, $\mathbf{R}$ is a vector of the $R_{p,i,j}$ values.

This means that

$$\frac{\partial c}{\partial c_p} = p_p \quad \text{and} \quad \frac{\partial R_s}{\partial R_{p,i,j}} = \mathbf{C_s} p_p \tag{32}$$

After each time the Cholesky is solved, a Brent line search is used again to find the minimum in the direction of the Cholesky solution. After each iteration, the star positions and fluxes are refit independently and outlier stars are rejected. Once the decrease in the χ^2 is below a certain level or when five iterations have been done, outlier pixels are discarded and three more iterations are done.