**Vera C. Rubin Observatory
Data Management**
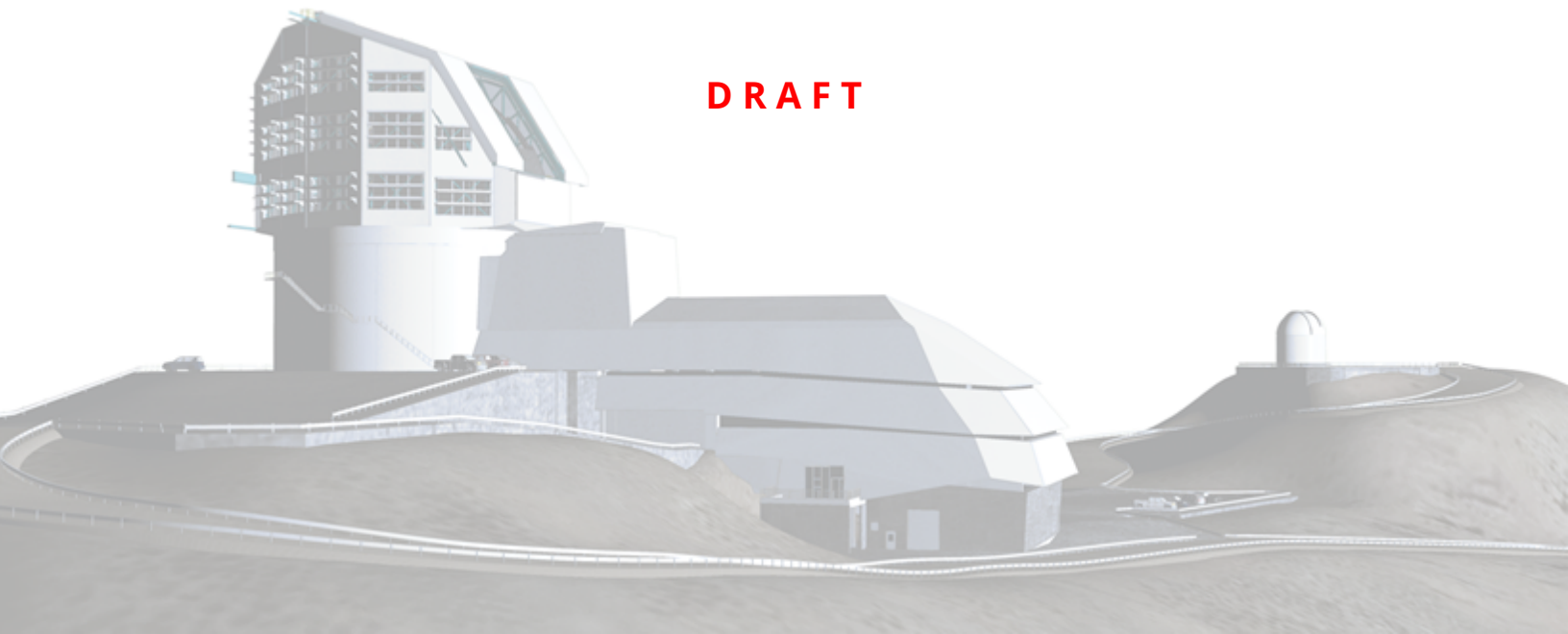
Consistent galaxy colors with Gaussian-Aperture and PSF photometry

Arun Kannawadi

DMTN-190

Latest Revision: 2022-01-10

D R A F T

Abstract

Gaussian-Aperture and PSF photometry algorithm measures consistent colors for extended sources such as galaxies that are largely independent of seeing. This technote briefly presents the mathematical description of the algorithm and focuses on its implementation in the Rubin Science Pipelines. This document also serves as a reference for several of the analytical calculations that are tested in the unit tests.

Change Record

Version	Date	Description	Owner name
1	YYYY-MM-DD	Unreleased.	Arun Kannawadi

Document source location: `https://github.com/lsst-dm/dmtn-190`

Contents

Consistent galaxy colors with Gaussian-Aperture and PSF photometry

1 Introduction

Galaxy photometry is a hard problem, if not impossible. Unlike stellar photometry (in non-crowded regions) where we know the true, intrinsic profile (a point source), we do not usually know the profile for extended sources (galaxies). This is especially true at the outskirts of the galaxies where they smoothly blend in with the sky background and we need to model the wings of the PSFs accurately. Errors in either of these models lead to systematic errors.

Fortunately, having total fluxes is sufficient, and not necessary, to define colors of objects. It is often easier to obtain color information from multiband images, and is sufficient for obtaining scientifically interesting quantities such as photometric redshifts. Total flux or magnitudes is useful to quantify stellar luminosities (SED of total light emitted in all the bands) and masses. However, in addition to a total magnitude in a reference band, ratios of fluxes in different bands (colors) are sufficient for studies involved in photometric redshifts, stellar populations, star-formation rates etc. They usually have different requirements, and will coincide for un-resolved/point sources.

1.1 Notation

Throughout this technote, we use the following notation consistently.

- $g(\mathbf{x})$: unobservable pre-seeing galaxy profile

- $G(\mathbf{x})$: observable image of galaxy with a Gaussian PSF

- $I(\mathbf{x})$: observed image of the galaxy with some PSF

- $P(\mathbf{x})$: PSF profile in the observed image $I(\mathbf{x})$, including the pixel response.

- $K(\mathbf{x})$: convolution kernel that converts $I(\mathbf{x})$ to $G(\mathbf{x})$, with a desired Gaussian PSF.

- $w(\mathbf{x})$: aperture applied on the pre-seeing image $g(\mathbf{x})$ (referred to as 'pre-seeing aperture')

- $A(\mathbf{x})$: aperture applied on the observed image $I(\mathbf{x})$ (referred to as 'post-seeing aperture')

1.2 Consistent colors

Definition: If we apply the same weight to each star in a galaxy across all bands, then it is a consistent color. In practice, this means use the same weight function $w(\mathbf{x})$ across all bands. If the galaxy has the same morphology in two bands, then a consistent color is the same as 'complete' color, which is the ratio of total fluxes.

If we model a galaxy purely as a collection of stars, $g^{(b)}(\mathbf{x}) = \sum_* f_*^{(b)} \delta_D(\mathbf{x} - \mathbf{x}_*)$. Its total flux $F^{(b)} = \int \mathrm{d}^2\mathbf{x}\, g^{(b)}(\mathbf{x}) = \sum_* f_*^b$. We define *complete color* (or total color) as the color of a galaxy obtained from its total fluxes. Given two passbands b_1 and b_2, a *complete color* is defined as

$$C(b_1, b_2) \equiv \frac{F_w^{(b_1)}}{F_w^{(b_2)}} = \frac{\int \mathrm{d}^2\mathbf{x}\, g^{(b_1)}(\mathbf{x})}{\int \mathrm{d}^2\mathbf{x}\, g^{(b_2)}(\mathbf{x})} \tag{1}$$

It is worth noting that a complete color is not the same as the average color of its constituent stars, i.e., $C(b_1, b_2) \neq \left\langle \frac{f_*^{(b_1)}}{f_*^{(b_2)}} \right\rangle_*$

Given two passbands b_1 and b_2, a *consistent color* is defined as

$$C_w(b_1, b_2) \equiv \frac{F_w^{(b_1)}}{F_w^{(b_2)}} = \frac{\int \mathrm{d}^2\mathbf{x}\, w(\mathbf{x}) g^{(b_1)}(\mathbf{x})}{\int \mathrm{d}^2\mathbf{x}\, w(\mathbf{x}) g^{(b_1)}(\mathbf{x})} \tag{2}$$

A complete color is a consistent color, as it corresponds to the choice of $w(\mathbf{x}) \equiv 1$. A consistent color defined with some weight w is insensitive to w and identical to complete color in the absence of color gradients. While $F_w^{(b)}$ is not an estimator of *total* flux, the ratio $F_w^{(b_1)}/F_w^{(b_2)}$ (consistent color) matches the ratio of total fluxes $F^{(b_1)}/F^{(b_2)}$ (complete color) if the morphology is the same in the two bands, i.e., $g^{(b)}(\mathbf{x}) \propto F^{(b)} g(\mathbf{x})$ for $b = b_1, b_2$. While this is generally not true for galaxies, it is true for stars and other point sources. In fact, $F_w^{(b)} = F^{(b)}$ for point sources if $w(\mathbf{x})$ is normalized to have a unit amplitude. For extended sources, $C_w(b_1, b_2)$ is sensitive to the weight function. This means that in the presence of a non-trivial weight $w(\mathbf{x})$, galaxy color is not invariant to moving the stars around within the galaxy, which the ratio of total fluxes is insensitive to. The complete color is only sensitive to changes in the fraction of different stellar populations within a galaxy.

Having a compact weight function allows extracting the color information from the bright regions of a galaxy by upweighting the central regions. If the color gradients were absent, one

could choose the formally optimal (or a near-optimal) weight function $w(\mathbf{x})$ for each galaxy to maximize the signal-to-noise ratio of its color estimate. However, since the color of a galaxy is sensitive to w in practice, varying w for each galaxy may lead to difficulty in calibrating photometric redshift codes. Furthermore, choosing the shape[1] of w from the same image we measure the flux from can lead to the estimate being non-linear and biased. It might be preferable to use a weight function that is circular and slightly sub-optimal that is appropriate to a population of galaxy. Having a variety of weight functions allows one to study the population of stars as a function of radial distance.

2 Gaussian Aperture and PSF photometry

GAaP is essentially an aperture photometry algorithm that adjusts the aperture depending on the seeing, such that the "flux" measured is seeing-independent for a wide range of seeing. Intuitively, if the PSF is large, then one needs to use a smaller aperture and vice versa. We derive this formalsm mathematically below.

2.1 Formalism

The flux F_A of a galaxy image $I(\mathbf{x})$ with an aperture function $A(\mathbf{x})$ is defined as

$$F_A := \int d^2\mathbf{x}\, A(\mathbf{x}) I(\mathbf{x}) \tag{3}$$

$$= \int d^2\mathbf{x}\, A(\mathbf{x}) \int d^2\mathbf{y}\, g(\mathbf{y}) P(\mathbf{x} - \mathbf{y}) \tag{4}$$

$$= \int d^2\mathbf{x} \int d^2\mathbf{y}\, A(\mathbf{x}) g(\mathbf{y}) P(\mathbf{x} - \mathbf{y}) \tag{5}$$

$$= \int d^2\mathbf{y} \left(\int d^2\mathbf{x}\, A(\mathbf{x}) P^*(\mathbf{y} - \mathbf{x}) \right) g(\mathbf{y}), \tag{6}$$

where in the last step, we swapped the order of integration in the last equation and defined a flipped PSF P^* such that $P^*(\mathbf{r}) = P(-\mathbf{r})$.

The pre-seeing aperture $w(\mathbf{y})$ is related to the post-seeing aperture $A(\mathbf{x})$ via the flipped PSF P^*.

$$w(\mathbf{y}) := \int d^2\mathbf{x}\, A(\mathbf{x}) P^*(\mathbf{y} - \mathbf{x}) \tag{7}$$

---

[1]It is unavoidable to use the detection peaks from the same image.

Thus, if the post-seeing aperture A is adjusted for seeing such as the pre-seeing aperture remains the same for all bands, then the fluxes estimated with w result in consistent colors. Given the choice of the intrisic weight function (pre-seeing aperture) $w(\mathbf{x})$, the post-seeing aperture used for photometring the source is formally given as

$$A(\mathbf{x}) = \left(w \otimes^{-1} P^*\right)(\mathbf{x}), \tag{8}$$

where $\otimes^{-1}$ denotes deconvolution operator. Thus, the use of this post-seeing aperture for photometring the source is designed to give colors that are insensitive to the seeing (DMS-REQ-0276).

In GAaP algorithm, the pre-seeing aperture is chosen to be a Gaussian function $w(\mathbf{x}) = \exp\left(-\frac{1}{2}\mathbf{x}^T W_{\text{int}}^{-1}\mathbf{x}\right)$ for some 2×2 positive-definite [2] matrix W_{int}. The choice of a Gaussian function over a flat top-hat aperture allows one to obtain colors from the bright central region of the galaxy well above the sky background and gives vanishingly low weight to the tail that is hidden in the noise. Furthermore, in addition to the SNR argument, the central parts of the galaxies are redder in color as they contain older population of stars. These populations are relatively easier[3] to obtain photometric redshifts from, thereby increasing the fidelity of the photo-z measurements.

Rather than using the formal solution to find the post-seeing aperture, the implementation of the algorithm uses a two-step procedure:

1. Convolve the image $I(\mathbf{x})$ by a kernel function K to obtain an PSF-Gaussianized image $G(\mathbf{x})$ which has a circular Gaussian PSF of size p

2. Photometer the resulting image using the $A(\mathbf{x}) = \exp\left(-\frac{1}{2}\mathbf{x}^T(W_{\text{int}} - p^2\mathbf{1})^{-1}\mathbf{x}\right)$, where $W_{\text{int}} - p^2\mathbf{1}$ is positive-definite.

Hence, the algorithm is named Gaussian-Aperture and (Gaussian) PSF.

Based on Eq. A16 of Kuijken et al. (2015), the GAaP flux with an aperture parameter $\mathbf{W}$ is

---

[2] A 2×2 positive-definite matrix M is a symmetric matrix with real entitites if and only if $\mathbf{x}^T M\mathbf{x} > 0$ for all $\mathbf{x} \in \mathbb{R}^2$. In practice, this implies $M_{11} > 0$, $M_{22} > 0$ and $\det(M) > 0$.

[3] https://project.lsst.org/meetings/law/sites/lsst.org.meetings.law/files/Galaxy%20Photometry%20-%20Konrad%20Kuijken.pdf

defined as follows:

$$F_{\mathbf{W}} \equiv \int \mathrm{d}^2\mathbf{x}\, g(\mathbf{x}) \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{W}^{-1}\mathbf{x}\right) \tag{9}$$

$$= \frac{1}{2}\frac{\det(\mathbf{W})^{1/2}}{\det(\mathbf{W}-p^2\mathbf{1})^{1/2}} \times 2\int \mathrm{d}^2\mathbf{x}\, G(\mathbf{x}) \exp\left(-\frac{1}{2}\mathbf{x}^{\mathsf{T}}(\mathbf{W}-\mathbf{p^2 1})^{-1}\mathbf{x}\right), \tag{10}$$

where $G(\mathbf{x})$ is the image of $g(\mathbf{x})$ after convolution by a Gaussian PSF of size p. Given a set of passbands $\{b_1, b_2, \ldots b_n\}$, the corresponding $F_{\mathbf{W}}$s can be used to obtain any of the $(n-1)$ independent color terms.

In the limit $g(\mathbf{x}) \to F\delta(\mathbf{x})$, the GAaP flux $F_{\mathbf{W}}$ equals F. The fact that GAaP fluxes are estimates of total fluxes for point sources is a useful one in practice. This enables the use the standard photometric calibrations developed for other flux measurement algorithms usable for GAaP as well.

3   Implementation in the Rubin science pipelines

GAaP algorithm is implemented as a measurement plugin that can be run in forced mode.

3.1   Definitions

This subsection consists the definition of various configuration parameters, internal variables and flags that may be needed to use the algorithm effectively.

The size p is set as some factor f times the trace radius of the original PSF for a given source in a given band. The scaling factor f is enforced to be larger than unity to avoid deconvolution. The scaling factors can be configured using the `scalingFactors` field. By default, $f = [1.15]$.

We use the basic building blocks in `modelPsfMatchTask` to Gaussianize the PSF, with one key difference: by default we convolve the PSF-matching kernel in Fourier space to speed up the computation (this may be configured to carry out the convolution in real-space or overlap-add method if preferred). Occasionally, the solution for the kernel fails, and no measurement is possible for any values of W_{int}. We mark such failures by setting `_flag_gaussianization`.

The factor multiplying the integral, especially $\det(\mathbf{W}-p^2\mathbf{1})^{1/2}$, is required to keep $F_{\mathbf{W}}$ PSF-

independent. Internally, we refer to this factor, the square root of the ratio of determinants, as `fluxScaling`. We scale the `instFlux` value from the `computeFixedMomentsFlux` method with `fluxScaling`.

The simplest case of apertures are those described by pre-determined circular apertures $\mathbf{W}_{\text{int}} = \sigma_w^2 \mathbf{1}$ for some σ_w. The set of σ_w is configurable using `sigmas` parameter. For such circular apertures, $\text{fluxScaling} = \frac{1}{2}\frac{\sigma_w^2}{\sigma_w^2 - p^2}$, where $\mathbf{W} = \sigma_w^2 \mathbf{1}$. A necessary condition for the measurement to be valid is that σ_w must be larger than p. When this condition is not met, usually for bad seeing images, `_flag_bigPsf` is set for the corresponding measurement.

Additionally, we also use one elliptical aperture that is adapted to the galaxy. We adopt the following heuristic to increase the signal-to-noise ratio: The post-seeing aperture is chosen to be the adaptive moments of the source in the reference band. For the purpose of optimal photometry alone, we assume that the PSF in the reference band is Gaussian and add the PSF adaptive moments to the source adaptive moments to obtain W_{int}. The intrinsic aperture W_{int} is then used in all measurement bands. This simple heuristic gives us a near-optimal photometry. This differs from true optimal aperture depending on how non-Gaussian the PSF is. However, given the low sensitivity to the aperture parameters, the heuristic aperture leads to an increase in SNR, with the improvement being larger for galaxies with larger ellipticity values.

3.2 Evaluating the integrals

The integral in Eq. 10 is computed using `computeFixedMomentsFlux`[4] by passing the PSF-Gaussianized image and shape parameter ($\mathbf{W} - p^2 \mathbf{1}$). This method evaluates the Gaussian weight at pixel centers and computes the inner product with the pixel values of the PSF-Gaussianized image. For Nyquist-limited samples of band-limited function,

$$\int \mathrm{d}^2 x \, f(x)g(x) = \sum_n g[n]\,f[n] \tag{11}$$

Even though the observed image will be band-limited to some extent, the Gaussian aperture we apply is not. As a result, the inner product will be biased estimator of the integral. In par-

---

[4]Note: `computeFixedMomentsFlux` computes the flux of an image $I(\mathbf{x})$ weighted by an aperture $\mathbf{Q}$ as $2 \int \mathrm{d}^2\mathbf{x}\, I(\mathbf{x}) \exp\left(-\frac{1}{2}\mathbf{x}^T \mathbf{Q}^{-1}\mathbf{x}\right)$, with a factor 2 in the normalization. This is such that if $I(\mathbf{x})$ is a Gaussian with shape $\mathbf{Q}$ and has a total flux F, the integral evaluates to F.
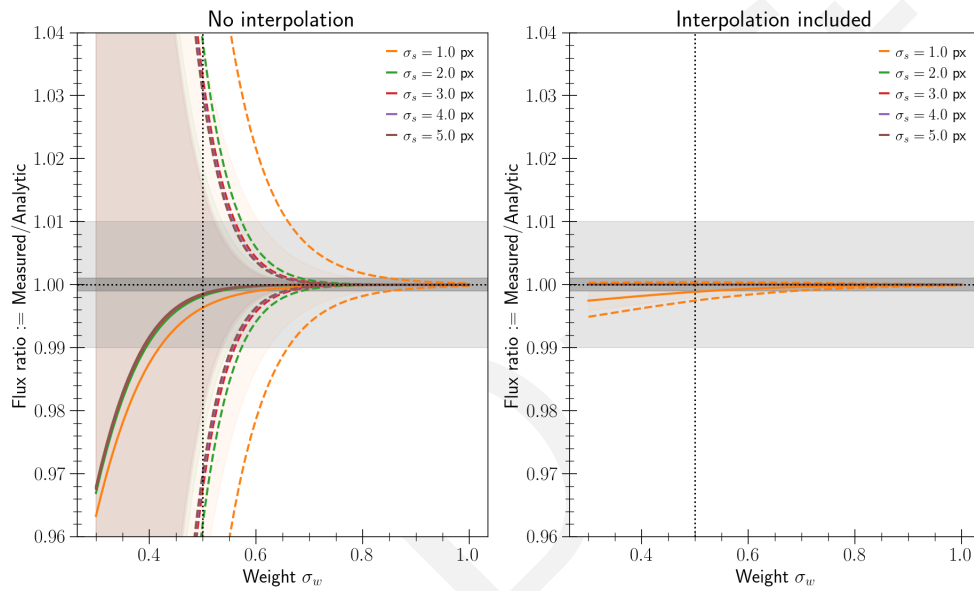
FIGURE 1: Plots of the ratio of measured flux to the expectation from analytical expression vs. size of the pre-seeing weight function (σ_w) for five different source sizes (σ_s). The right panel shows the ratio with the `sinc` interpolation turned on, and the panel on the left shows without. The solid lines show the mean ratio when averaged over all possible sub-pixel offsets. The filled regions show the 16-84 percentile region while the dashed lines show the minimum and maximum possible values. The dark and the light gray shaded horizontal regions represent 0.1% and 1% relative errors in the measured flux, respectively.

ticular, the measured flux becomes sensitive to where the centroid is located with a pixel. In principle, one needs to interpolate the pixels to form a continuous function, and then integrate the function with the Gaussian aperture (see Bickerton & Lupton (2013)). However, this procedure is computationally expensive.

To quantify whether we can nevertheless make this approximation, and if so, when this approximation breaks down, the following setup is used. For simplicity, consider a circular Gaussian source with an effective PSF (includes pixel response) such that the observed profile is a circular Gaussian. We denote its size (in pixels) as σ_s and the size (in pixels) of the circular Gaussian weight function by σ_w. The image is rendered with sub-pixel offsets uniformly varying from zero to half a pixel offset in either directions. The centroid of the Gaussian aperture coincides with the centroid of the source for each of the cases. The GAaP flux is measured for a range of σ_w and σ_s values and the results are plotted in the left panel of Fig. 1. For large values of σ_s and σ_w, the measured flux is in agreement with the value expected analytically. For $\sigma_w \lesssim 0.7$, the measured flux becomes strongly dependent on the sampling locations, even for large values of σ_s. After marginalizing over the centroid locations, the mean flux is still an underestimate of the expected value.

If we instead interpolate the pixels with a `sinc` function to form a continuous function and then evaluate the integral, the measured flux is in much better agreement with the analytical expectations. Adapting the method suggested in Bickerton & Lupton (2013) for Gaussian apertures, we compute a set of coefficients to take inner product with the pixel values of the image. The measured fluxes agree with the expectation for large ($\sigma_s \gtrsim 2$) and biased only slightly for $\sigma_s \sim 1$ when $\sigma_w \lesssim 0.6$

If we measure its flux from pixellated images with a circular Gaussian aperture of size σ_w pixels, the expected GAaP flux is $F\left(1 + \frac{\sigma_s^2}{\sigma_w^2}\right)^{-1}$ (see Eq. 26). Fig. 1 shows that with the interpolation turned on, the measured flux is largely insensitive to the sub-pixel offset of the centroid unless both the source and the aperture are narrow, where a small negative bias is seen. Without the interpolation, a large scatter can be seen for $\sigma_w \lesssim 0.7$ pixels due to the location of the centroid within the central pixel, with the mean marginalized over the centroid location shows a negative bias as well, even for large sources. `computeFixedMomentsFlux` turns on the interpolation automatically when $\sigma_w \leq 0.5$ pixels. This will cause the measured flux to change abruptly that may be significant for small sources.

For $\sigma \gtrsim 0.85$ pixels, the flux is accurate to better than 0.1%. In practice, we do not expect to

use such narrow Gaussian apertures.

3.3 Estimating noise covariance

Let us calculate the covariance C^G between the errors on two pixel values $G(\mathbf{x})$ and $G(\mathbf{y})$. Rewriting Eq. A9 of Kuijken et al. (2015), we get

$$Cov(G(\mathbf{x}), G(\mathbf{y})) = \int \int \mathrm{d}^2\mathbf{x}' \mathrm{d}^2\mathbf{y}' \, Cov(I(\mathbf{x}'), I(\mathbf{y}')) \, K(\mathbf{x} - \mathbf{x}')K(\mathbf{y} - \mathbf{y}') \qquad (12)$$

In KiDS papers, including Kuijken et al. (2015), translational invariance is assumed and $Cov(I(\mathbf{x}'), I(\mathbf{y}'))$ is expressed as a function of $(\mathbf{x}' - \mathbf{y}')$ alone and an appropriate covariance matrix is constructed empirically.

However, in Rubin science pipelines, we keep track of the variance (and only variance) per pixel. So, our model for $Cov(I(\mathbf{x}'), I(\mathbf{y}')) = \sigma^2(\mathbf{x}')\delta_D(\mathbf{x}' - \mathbf{y}')$, where $\sigma^2(\mathbf{x}')$ is given by the variance plane. This is an approximation, as the noise *is* correlated on the coadds. However, we proceed with the information we have available. Substituting this in the above equation, we get

$$Cov(G(\mathbf{x}), G(\mathbf{y})) = \int \mathrm{d}^2\mathbf{x}' \, \sigma^2(\mathbf{x}')K(\mathbf{x} - \mathbf{x}')K(\mathbf{y} - \mathbf{x}') \qquad (13)$$

The kernel K is expected to be compact, and if the variance plane is slowly varying, we can approximate $\sigma^2(\mathbf{x}')$ by the variance value at the centroid of the source, say σ^2. This approximation is further justified because of Gaussian weighting we will employ in Eq. 15, which makes the effective kernel even more compact. Redefining $\mathbf{x}' \rightarrow \mathbf{x} - \mathbf{x}'$ ($\mathrm{d}^2\mathbf{x}' \rightarrow \mathrm{d}^2\mathbf{x}'$ because the dimensionality is even), we get

$$Cov(G(\mathbf{x}), G(\mathbf{y})) \approx \sigma^2\xi_K(\mathbf{r}) \equiv C^G(\mathbf{r}), \qquad (14)$$

where $\mathbf{r} = \mathbf{y} - \mathbf{x}$. Thus, the covariance is the auto-correlation function of the kernel K $\xi_K(\mathbf{r}) \equiv \int \mathrm{d}^2\mathbf{x} \, K(\mathbf{x})K(\mathbf{x}+\mathbf{r})$ scaled by the variance σ^2 at the location of the source. The auto-correlation function is computed by the _computeKernelAcf static method.

3.4   Estimating uncertainties

Eq. A17 of Kuijken et al. (2015) says

$$\text{Var}(F_\mathbf{W}) = \frac{\det(\mathbf{W})}{\det(\mathbf{W} - p^2\mathbf{1})} \pi \det(\mathbf{W} - p^2\mathbf{1})^{1/2} \int d^2\mathbf{x}\, C^G(\mathbf{x}) \exp\left(-\frac{1}{4}\mathbf{x}^T(\mathbf{W} - p^2\mathbf{1})^{-1}\mathbf{x}\right). \qquad (15)$$

Note the missing factor of $2^{1/2}$, which is an error in Kuijken et al. (2015). Since we represent

$$C^G(\mathbf{r}) = \sigma^2 \xi_K(\mathbf{r}), \qquad (16)$$

we get

$$\text{Var}(F_\mathbf{W}) = \frac{\det(\mathbf{W})}{\det(\mathbf{W} - p^2\mathbf{1})} \left(\frac{1}{2}\right)^2 \times 4\pi\sigma^2 \det(\mathbf{W} - p^2\mathbf{1})^{1/2} \times \int d^2\mathbf{r}\, \xi_K(\mathbf{r}) \exp\left(-\frac{1}{2}\mathbf{r}^T(2(\mathbf{W} - p^2\mathbf{1}))^{-1}\mathbf{r}\right). \qquad (17)$$

We categorize them into three terms, separated by $\times$.

1. The naive calculation of flux variance by `computeFixedMomentsFlux` yields the middle term $4\pi\sigma^2 \det(\mathbf{W} - p^2\mathbf{1})^{1/2}$ (given by `instFluxErr`^{2}).

2. The factor $\frac{1}{4}\frac{\det(\mathbf{W})}{\det(\mathbf{W} - p^2\mathbf{1})}$ appears because of the scaling factor in $F_\mathbf{W}$ (Eq. A16 of Kuijken et al. (2015)).

3. The square root of the integral is computed by `_getFluxErrScaling` and referred to as `fluxErrScaling`. This is the actual contribution due to correlations in the noise introduced by PSF-Gaussianization procedure. This is computed using `computeFixedMomentsFlux` on the auto-correlation function, with $2\times$ the shape parameter of aperture used to measure $F_\mathbf{W}$.

   If the PSF had been Gaussian to begin with, or if we were neglecting the effects of correlated noise on flux uncertainties, $K(\mathbf{r}) = \delta_D(\mathbf{r})$. This leads to the integral evaluating to 1.

3.4.1 Alternative representation

Note that from the definition of $\xi_K(\mathbf{r})$, it follows that

$$\int \mathrm{d}^2\mathbf{r}\, \xi_K(\mathbf{r}) = \int \mathrm{d}^2\mathbf{r} \int \mathrm{d}^2\mathbf{x}\, K(\mathbf{x})K(\mathbf{x}+\mathbf{r}) \tag{18}$$

$$= \int \mathrm{d}^2\mathbf{x}\, K(\mathbf{x}) \int \mathrm{d}^2\mathbf{r}\, K(\mathbf{x}+\mathbf{r}) \tag{19}$$

$$= \left[\int \mathrm{d}^2\mathbf{x}\, K(\mathbf{x})\right]^2 \tag{20}$$

In a similar manner, the integral for `fluxErrScaling` can also be expressed as

$$\int \mathrm{d}^2\mathbf{r}\, \xi_K(\mathbf{r}) \exp\left(-\frac{1}{2}\mathbf{r}^T(2(\mathbf{W}-p^2\mathbf{1}))^{-1}\mathbf{r}\right) = \left[\int \mathrm{d}^2\mathbf{x} \int \mathrm{d}^2\mathbf{y}\, K(\mathbf{x}-\mathbf{y}) \exp\left(-\frac{1}{2}\mathbf{y}^T(\mathbf{W}-p^2\mathbf{1})^{-1}\mathbf{y}\right)\right]^2. \tag{21}$$

The proof follows by considering $\int \mathrm{d}^2\mathbf{x} f(\mathbf{x}) = \tilde{f}(0)$, where $\tilde{f}$ is the Fourier transform of f.

4 Concluding remarks

This technote presents the Gaussian Aperture and PSF photometry algorithm and covers various technical details with regards to its implementation in the Rubin Science Pipelines. This note also captures various calculations that are used in testing the code, since they are cumbersome to leave as comments within the code itself.

A References

Bickerton, S.J., Lupton, R.H., 2013, MNRAS, 431, 1275 (`arXiv:1302.4764`), doi:10.1093/mnras/stt244, ADS Link

Kuijken, K., Heymans, C., Hildebrandt, H., et al., 2015, MNRAS, 454, 3500 (`arXiv:1507.00738`), doi:10.1093/mnras/stv2140, ADS Link

B Acronyms

Acronym	Description
DM	Data Management
DMS	Data Management Subsystem
DMS-REQ	Data Management System Requirements prefix
DMTN	DM Technical Note
PSF	Point Spread Function
SED	Spectral Energy Distribution
SNR	Signal to Noise Ratio

C Gaussian integrals

The integrals in computing various GAaP quantities can be computed analytically when the PSF and the intrinsic sources are Gaussian. We will derive the analytical results that are used in the unit testing of the code.

For an intrinsic Gaussian source $g(\mathbf{x})$ of total flux F and shape $\mathbf{S}$ and a Gaussian aperture of shape $\mathbf{W}$, it follows from the definition 10 that its GAaP flux is

$$F_{\mathbf{W}} = \int d^2\mathbf{x} \left[\frac{F}{2\pi \det(\mathbf{S})} \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{S}^{-1}\mathbf{x}\right) \right] \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{W}^{-1}\mathbf{x}\right) \tag{22}$$

$$= \frac{F}{2\pi \det(\mathbf{S})} \int d^2\mathbf{x} \, \exp\left(-\frac{1}{2}\mathbf{x}^T(\mathbf{S}^{-1}+\mathbf{W}^{-1})\mathbf{x}\right) \tag{23}$$

$$= F \frac{2\pi \det\left((\mathbf{S}^{-1}+\mathbf{W}^{-1})^{-1}\right)}{2\pi \det(\mathbf{S})} \tag{24}$$

$$= F \frac{1}{\det(\mathbf{S})\det(\mathbf{S}^{-1}+\mathbf{W}^{-1})} \tag{25}$$

$$= F \frac{\det(\mathbf{S}^{-1})}{\det(\mathbf{S}^{-1}+\mathbf{W}^{-1})} \text{ (or) } \frac{F}{\det(\mathbf{1}+\mathbf{S}\mathbf{W}^{-1})}. \tag{26}$$

If $\mathbf{S} = \sigma_s^2\mathbf{1}$ and $\mathbf{W} = \sigma_w^2\mathbf{1}$, then $F_{\mathbf{W}} = F\left(1+\frac{\sigma_s^2}{\sigma_w^2}\right)^{-1}$.

Suppose $I(\mathbf{x})$ has a circular Gaussian PSF of size s and the target PSF is a Gaussian PSF of size $p = fs$ ($f > 1$), then the kernel K is also a Gaussian of size $s(f^2-1)^{1/2}$. In the GAaP plugin, the values for f are given by `scalingFactors`.

Since flux-conservation implies $\int d^2\mathbf{x}\, K(\mathbf{x}) = 1$, we can immediately write

$$K(\mathbf{x}) = \frac{1}{2\pi(f^2 - 1)s^2} \exp\left(-\frac{\mathbf{x}^T\mathbf{x}}{2(f^2 - 1)s^2}\right) \tag{27}$$

The auto-correlation function $\xi_K(\mathbf{r})$ is then given by

$$\xi_K(\mathbf{r}) = \frac{1}{4\pi(f^2 - 1)s^2} \exp\left(-\frac{\mathbf{r}^T\mathbf{r}}{4(f^2 - 1)s^2}\right). \tag{28}$$

This is easy to see by first recognizing that $\xi_K(\mathbf{r})$ must be a Gaussian with size $\sqrt{2}$ times larger than that of K and the normalization factor follows that $\xi_K(\mathbf{r})$ must integrate to 1 if K integrates to 1 (see Eq. 20).

The square of the `fluxErrScaling` parameter is then given by the Gaussian integral

$$\int d^2\mathbf{r}\, \xi_K(\mathbf{r}) \exp(-\frac{\mathbf{r}^T\mathbf{r}}{4(\sigma^2 - p^2)}),$$

where σ is the size of the Gaussian aperture for the pre-seeing source $g(\mathbf{x})$. In the GAaP plugin, these values are given by `sigmas` config parameter.

The exponents of the Gaussians is simplified as follows.

$$\frac{-\mathbf{r}^T\mathbf{r}}{4}\left(\frac{1}{(f^2 - 1)s^2} + \frac{1}{(\sigma^2 - f^2s^2)}\right) = \frac{-\mathbf{r}^T\mathbf{r}}{4}\left(\frac{\sigma^2 - s^2}{s^2(f^2 - 1)(\sigma^2 - f^2s^2)}\right) \tag{29}$$

, where p is replaced by fs. Evaluating the Gaussian integral, we get

$$\texttt{fluxErrScaling}^2 = \frac{1}{4\pi(f^2 - 1)s^2} \times \frac{\pi 4s^2(f^2 - 1)(\sigma^2 - f^2s^2)}{\sigma^2 - s^2} \tag{30}$$

$$= \frac{\sigma^2 - f^2s^2}{\sigma^2 - s^2} \tag{31}$$

As a consistency check, we get `fluxErrScaling` = 1 for $f = 1$, i.e., if we do not carry out the PSF-Gaussianization procedure. For $f > 1$, `fluxErrScaling` < 1. As an aside, if $\xi_K(\mathbf{r}) \geq 0$ for all $\mathbf{r}$ (as in the Gaussian kernel case), `fluxErrScaling` is guaranteed to be less than 1. In other words, the naive flux uncertainty overestimates the true uncertainty. For naive errors to be an underestimation of true errors, it is necessary that the kernel should be negative for sufficiently small $|\mathbf{r}|$. Intuitively, this makes sense; non-negative valued kernel smoothes

the noise in the image, reducing its power, whereas as kernel with both positive and negative values has the potential to amplify the random fluctuations in different pixels.