KASP Technical notes #04:
KECK NIRC2 images processing with KASP

O. Beltramo-Martin[*], C. M. Correia

Contents

1 Purpose

This KASP note dedicates to NIRC2 images processing in the purpose to be compared to reconstructed PSF coming from the AO telemetry. We focus on the different steps of cleaning NIRC2 images and how PSF characteristics, especially Strehl ratio and FWHM, are estimated.

In the goal of producing a better understanding of the PSF reconstruction gain and limitations, we also focus on astrophysics metrics, as the relative photometric and astrometric error. We aim at determining what is the real impact of reconstruction residual onto the science behind and providing clear messages to the community. We detail how they're derived and illustrate how to use the existing routine delivering these estimations.

[*] olivier.martin@lam.fr

2 Processing pipeline

We start from a raw image *rawImg* that feeds a cleaning procedure split into the following steps :

- **Unbias from background and flat**
  Input : rawImg (image), bg (image) and flat (image) - Output : Clean image
  The calibrated background is subtracted to the raw image and then compensated from the flat variation across pixels. We get the following operation :
  $$img = \frac{rawImg - bg}{flat} \tag{1}$$

- **Bad pixels management**
  Input : rawImg (image), badPixelMap (image) - Output : Clean image
  We are provided by a calibrated bad pixel map (logical values, 1 for bad pixel, 0 either) that is passed to a routine (psfrTools.corrDeadPixFrame) designed to identify valid neighbors pixels to each bad one. The bad pixel value is then extrapolated to :
  $$\mathrm{PSF}(i,j) = \sum_{k=-1}^{1}\sum_{l=-1}^{1} v_{kl} \times w_{kl} \times \mathrm{PSF}(i+k, j+l), \tag{2}$$
  with
  $$w_{kl} = \begin{cases} 2 & \text{if} \quad |k|+|l| = 1 \\ 1 & \text{if} \quad |k|+|l| = 2 \\ 0 & \text{else} \end{cases}, \tag{3}$$
  and v_{kl} set to (i) either 0 if the pixel shifted by (k,l) displacement is identified as a bad pixel as well (ii)or one if the pixel is not identified as a bad one. To ensure a proper extrapolation, we may verify we have enough good pixels surrounding the bad one : we set an arbitrary threshold $\sum_{k=-1}^{1}\sum_{l=-1}^{1} v_{kl} \times w_{kl} = 4$. It is reached when the four diagonal neighbors are valid or two edges ones for instance. If this criteria is not verified, the bad pixel is still considered as bad until the end of the procedure. At this stage, if some bad pixels remain, the badPixelMap is duplicated and set to zero at previous bad pixel position that have been corrected during the first passage through the procedure. The corrected image is getting through the routine recursively until there's no more bad pixel.

- **Glitchs filtering**
  Input : rawImg (image) - Output : clean image
  Perform a median filter on the image for removing hot pixels still contented into the image.

- **Setting the field of view**
  Input : rawImg (image), N (#pixels) - Output : Cropped image
  The PSF is cropped to reach a support size the input N. This step requires to determine the PSF position to keep the PSF at the center location before cropping, which is performed by retrieving the maximum value in the image. The maximum of the image is set at the middle pixel.

- **PSF recentering**
  Input : oversampling (scalar) - Output : PSF recentered.
  The PSF is oversampled to a high resolution by zero-padding the OTF on oversampling× the initial support size. The peak position is then determined on this PSF and set up to the image center. We then get back to the initial resolution by cropping the OTF. if oversampling values zero, no centering is performed.

- **Filtering high angular frequencies**
  Input : OTF - Output : filtered OTF.
  We filter high angular frequencies higher than the telescope cut-off frequency D/λ :

$$\mathrm{OTF}\left(\frac{|\boldsymbol{u}|}{\lambda} > D/\lambda\right) = 0. \tag{4}$$

  The frequency content maybe be not equal to zero because numerical artifact when computing the OTF from the PSF. Physically talking, these frequencies does not exist because the telescope pupil filtering and must be removed is they exist before getting back to the PSF domain.

- **CCD pixel response compensation**
  Input : rawOTF - Output : filtered OTF.
  PSF reconstruction algorithm is producing a PSF free from any pixel response. To enhance contrast on the NIRC2 PSF and compared the real structure to the reconstructed PSF, we must unbias the OTF from the CCD pixel response as follows :

$$\mathrm{OTF}(\boldsymbol{u}/\lambda) = \frac{\mathrm{rawOTF}(\boldsymbol{u}/\lambda)}{|\mathrm{sinc}(u_x/\lambda/2)| \times |\mathrm{sinc}(u_y/\lambda/2)|}, \tag{5}$$

  with $\mathrm{sinc}(x) = \sin(x)/x$. In the case of NIRC2 images, the PSF is well sampled (1.5 pixels in the diffraction in H-band) and the operation made in Eq. 5 does not require to take particularly care about any division by zero.

- **Denoising**
  Input : OTF, nfit (scalar) - Output : filtered OTF.
  This step is not used anymore because not so trusted in terms of efficiency. The point here was to model fit the nfit first values of the OTF, except the peak, by a polynomial function of order nfit. This step provides what the zero value of the OTF should be a when comparing to the real value we get an estimation of the noise level. The OTF peak is forced to the fitted value retrieved by the procedure.

- **Thresholding**
  Input : rawImg (image) - Output : Clean image
  Instead of using the previous routine to take care of the noise, we directly subtract the median value on all pixels. We set to zero all pixels with negative values.

3 PSF estimates

3.1 the *psfStats* class

This class is the output of the *psfR* class that performs the reconstruction from the *system* class. The goal of the *psfStats* is to gather all information about the PSF into a single facility when providing the image coming from a simulation of a real on-sky acquisition.

Calling this class allows automatically to run several routines for determining PSF characteristics as Strehl ratio, FWHM, ellipticity and ensquared energy. The user can also provide a reference PSF (psfRef) that allows to estimate relative photo-astrometry errors accuracy caused by difference between the PSF passed to psfStats and this reference. Note that these quantities are not absolute values; they only traduce difference onto PSF in terms of understandable metrics. Tab. 3.1 summarizes the psfStats properties and their meaning.

Dependent classes	
src	Class *source* including imaged source properties.
tel	Class *telescope* including telescope properties.
Image properties	
image	Matrix containing the image.
psfResolution	Number of pixels defining the image support size.
pixelScaleInMas	Pixel scale in mas.
nyquistSampling	Image sampling, value 1 for a pixel scale set to $\lambda/2/D$
psfFieldOfView	Image field of view in mas and owes psfResolution×pixelScaleInMas.
focalGrid	Definition of angular separation grid from psfResolution and pixelScaleInMas.
PSF characteristics	
Strehl	Strehl ratio value.
StrehlAccuracy	Accuracy on the Strehl ratio estimation if not provided.
FWHM	Full width at half maximum. Set up by the Moffat or Gaussian fitting if called.
FWHMAccuracy	Accuracy on the FWHM when fitting the PSF.
FluxInFWHM	Percentage of PSF energy contained in the FWHM.
Ellipticity	Ratio between largest and smallest FWHM.
EnsquaredEnergy	Ensquared energy as function of separation for multiple of λ/D.
Photo-astrometry characteristics	
exposureTime	Camera exposure time.
bg	Median value of the background.
ron	Estimated read-out noise standard-deviation on the image.
flux	Estimated flux in the image in counts units.
nPhoton	Defined by Flux/tel.area/exposureTime.
magnitude	Defined by -2.5log10(nPhoton/src.zeroPoint)
psfRef	Reference PSF for measuring photo-astrometric accuracy.
magAccuracy	Relative magnitude error when comparing image to psfRef
xSrc	x location of the PSF in image
ySrc	y location of the PSF in image
astroAccuracy	Relative astrometric error when comparing image to psfRef
Moffat fitting	
flagMoffat	Set to 1 for fitting the PSF by a Moffat function.
MoffatImage	Fitted Moffat image onto image.
MoffatResidual	Fitting residual.
MoffatParam	Retrieved Moffat parameters.
MoffatStd	Standard-deviation on MoffatParam values.
Gaussian fitting	
flagGaussian	Set to 1 for fitting the PSF by a Gaussian function.
GaussianImage	Fitted Gaussian image onto image.
GaussianResidual	Fitting residual.
GaussianParam	Retrieved Gaussian parameters.
GaussianStd	Standard-deviation on GaussianParam values.

Table 1: Summary of *psfStats* class properties. Bold input are required parameters, while italic ones can be provided.

3.2 Strehl ratio

Strehl ratio is commonly defined as the PSF maximum intensity referred to the diffraction limit case. It can be defined by the energy contained into the angular frequencies that leads naturally to the following definition :

$$\widehat{\mathrm{SR}} = \frac{\iint_{-D/\lambda}^{D/\lambda} \mathrm{OTF}(\boldsymbol{u}/\lambda), d\boldsymbol{u}/\lambda}{\iint_{-D/\lambda}^{D/\lambda} \mathrm{OTF}_{\mathrm{DL}}(\boldsymbol{u}/\lambda), d\boldsymbol{u}/\lambda}, \tag{6}$$

where OTF is the Optical transfer function derived from the PSF and $\mathrm{OTF}_{\mathrm{DL}}$ the diffraction limit OTF. The first one is computed from the Fourier transform of the PSF while the second one is derived from the telescope pupil auto-correlation :

$$\begin{aligned} \mathrm{OTF}(\boldsymbol{u}/\lambda) &= \iint \mathrm{PSF}(\boldsymbol{\alpha}) \exp(-2i\pi\boldsymbol{\alpha}\boldsymbol{u}/\lambda) d\boldsymbol{\alpha} \\ \mathrm{OTF}_{\mathrm{DL}}(\boldsymbol{u}/\lambda) &= \iint \mathcal{P}(\boldsymbol{r})\mathcal{P}^*(\boldsymbol{r}+\boldsymbol{u}) d\boldsymbol{r}. \end{aligned} \tag{7}$$

Practically, term OTF is sampled from $-D/\lambda$ to D/λ by a step imposed by *psfResolution* when computing from the Matlab *fft2* routine. The diffraction limit $\mathrm{OTF}_{\mathrm{DL}}$ is derived by the zero-padded telescope pupil to a factor *nyquistSampling* to ensure the same resolution and then interpolate to *psfResolution* to get the same field of view.

From Eq. 6, Strehl ratio value is also estimated by :

$$\begin{aligned} \widehat{\mathrm{SR}} &= \frac{\mathrm{PSF}(0,0)}{\mathrm{flux}} \times \frac{\mathrm{flux}_{\mathrm{DL}}}{\mathrm{PSF}_{\mathrm{DL}}(0,0)}, \\ &= c_{\mathrm{DL}} \times \frac{\mathrm{PSF}(0,0)}{\mathrm{flux}} \end{aligned} \tag{8}$$

where flux is the PSF flux and c_{DL} a constant value depending on the diffraction-limit PSF defined by ([1]):

$$c_{\mathrm{DL}} = \frac{\pi^2 \times (D^2 - o^2) \times ps^2}{4\lambda^2}, \tag{9}$$

with D the telescope diameter, o the central obstruction ratio, ps the camera pixel scale and λ the imaging wavelength. In the case of perfect determination of c_{DL}, the accuracy on the Strehl ratio turns to be :

$$\frac{\Delta\mathrm{SR}}{\mathrm{SR}} = \frac{\Delta\mathrm{PSF}(0,0)}{\mathrm{PSF}(0,0)} + \frac{\Delta\mathrm{flux}}{\mathrm{flux}}. \tag{10}$$

Because the additive read-out noise η reaching a standard-deviation value ron, the estimation of the maximum value of the PSF suffers by the same standard-deviation, while the estimation of the flux propagate a factor N, the number of pixels the flux estimation is based on. We end up to have the following accuracy on the Strehl ratio :

$$\frac{\Delta\mathrm{SR}}{\mathrm{SR}} = \frac{\mathrm{ron}}{\mathrm{PSF}(0,0)} + \frac{N \times \mathrm{ron}}{\mathrm{flux}}. \tag{11}$$

Note the previous formula does not take into account any photo noise and assume the pupil is perfectly known.

3.3 FWHM

3.3.1 Moffat fitting

Class *psfStats* offers the possibility to fit onto the PSF a Moffat function defined by :

$$\mathcal{M}(\alpha_x,\alpha_y) = I_0 \times \left(1 + \left(\frac{(\alpha_x - dx)\cos(\theta) + (\alpha_y - dy)\sin(\theta)}{l_x}\right)^2 + \left(\frac{(\alpha_y - dy)\cos(\theta) - (\alpha_x - dx)\sin(\theta)}{l_y}\right)^2\right)^{-\beta}, \quad (12)$$

where :

- I_0 is the amplitude,
- l_x, l_y is the spread factors in the two directions,
- d_x, d_y is the center position,
- θ is the rotation angle,
- β is slopes factor.

A internal routine in *psfStats* is called for least-squared minimizing the cost function :

$$\varepsilon^2\left(\left[I_0, l_x, l_y, d_y, d_y, \theta, \beta\right]\right) = \left\|\mathrm{PSF}\left(\alpha_x, \alpha_y\right) - \mathcal{M}\left(\alpha_x, \alpha_y, \left[I_0, l_x, l_y, d_y, d_y, \theta, \beta\right]\right)\right\|^2, \quad (13)$$

producing a set of estimated Moffat parameters. FWHM in x and y directions is derived then by :

$$\begin{aligned}\widehat{\mathrm{FWHM}}_x &= 2 \times \widehat{l_x} \times \sqrt{2^{1/\widehat{\beta}} - 1}\\ \widehat{\mathrm{FWHM}}_y &= 2 \times \widehat{l_y} \times \sqrt{2^{1/\widehat{\beta}} - 1}.\end{aligned} \quad (14)$$

The minimization is ensured by the Matlab routine *nlinfit* that allows to retrieve the standard-deviation on the estimates when calling the *nlparci* function. The *psfStats* FWHM field is then updated by taking the maximum value of FWHM and the ellipticity is defined as the ratio max value on min value.

3.3.2 Gaussian fitting

The follow exact same step as done with the Moffat fitting, except on the model function that is now defined as :

$$\mathcal{G}(\alpha_x,\alpha_y) = I_0 \times \exp\left(-0.5 \times \left(\frac{(\alpha_x - dx)\cos(\theta) + (\alpha_y - dy)\sin(\theta)}{l_x}\right)^2 + \left(\frac{(\alpha_y - dy)\cos(\theta) - (\alpha_x - dx)\sin(\theta)}{l_y}\right)^2\right), \quad (15)$$

where the inputs parameters are describing the same physical properties. FWHM values are estimated by :

$$\begin{aligned}\widehat{\mathrm{FWHM}}_x &= 2 \times \widehat{l_x} \times \sqrt{2\log(2)}\\ \widehat{\mathrm{FWHM}}_y &= 2 \times \widehat{l_y} \times \sqrt{2\log(2)}.\end{aligned} \quad (16)$$

3.4 Ensquared energy

Ensquared energy measures the ratio of PSF energy in an angular box size α to the total energy contained into the PSF. In the angular frequencies domain, this step resumes to multiply the OTF by the Fourier transform of a gate function, i.e a cardinal sinus :

$$\mathbf{EE}(\alpha) = \left(\frac{\alpha}{D}\right)^2 \iint_{-D\lambda}^{D\lambda} \text{OTF}(u_x/\lambda, u_y/\lambda) \times \text{sinc}(\pi\alpha u_x) \times \text{sinc}(\pi\alpha u_y) du_x du_y. \tag{17}$$

Practically the integral is derived using the *trapz* method from Matlab.

4 Towards astrophysics metrics

PSF characteristics may fail in producing a real understandable scalar value in the purpose of evaluating PSF reconstruction efficiency. The ultimate goal of PSF reconstruction stands on a science need : knowledge of the PSF is mandatory to retrieve accurately photometry, astrometry or even morphology of observed targets. We aint at exploring relative photo-astrometric accuracy limited by the PSF shape mis-reconstruction. These quantities does not take into account absolute limitations as the CCD counts/magnitude relation calibration or the catalogs stars position accuracy.

4.1 Relative photometric error

We start scaling the reconstructed PSF to minimize the difference with the reference PSF, that could be a real on-sky image PSF for instance :

$$\varepsilon^2\left(a\right) = ||\text{PSF}_{\text{ref}}\left(\boldsymbol{\alpha}\right) - a \times \text{PSF}\left(\boldsymbol{\alpha}\right)||^2 , \tag{18}$$

leading to estimate the scale factor as :

$$\widehat{a} = \underset{a}{\text{argmin}}\, \varepsilon^2\left(a\right). \tag{19}$$

The minimization is ensured by the Matlab least-square procedure *nlinfit*. We then define the relative photometric error in magnitude units as :

$$\Delta\text{mag} = -2.5 \times \log_{10}\left(\frac{\widehat{a} \times \iint \text{PSF}\left(\boldsymbol{\alpha}\right) d\boldsymbol{\alpha}}{\iint \text{PSF}_{\text{ref}}\left(\boldsymbol{\alpha}\right) d\boldsymbol{\alpha}}\right). \tag{20}$$

In the case of a perfect reconstruction, we get $\widehat{a} = 1$ and a photometric error valuing zero.

4.2 Relative astrometric error

The PSF shape mis-reconstruction will cause a misleading on the PSF location. To traduce shape disparity onto astrometry, we start defining a binary star as :

$$\text{Binary}\left(\boldsymbol{\alpha}, \text{PSF}, \boldsymbol{l}\right) = \text{PSF}\left(\boldsymbol{\alpha}\right) + \text{PSF}\left(\boldsymbol{\alpha} + \boldsymbol{l}\right), \tag{21}$$

that superimposes the input PSF by shifting one by the $\boldsymbol{l}$ vector in keeping the first one exactly on-axis. Practically, the binary model is the covolution of a two points source with the input PSF, the separation is allowed to reach sub-pixel levels.

To measure the astrometric error, we define a model of binary using the reference PSF and a given separation $\boldsymbol{l}_{\text{ref}}$, then play the game of fitting the separation of another binary derived from the reconstructed PSF by minimizing :

$$\varepsilon^2\left(\boldsymbol{l}_{\text{ref}}, \boldsymbol{l}\right) = \left|\left|\text{Binary}\left(\boldsymbol{\alpha}, \text{PSF}_{\text{ref}}, \boldsymbol{l}_{\text{ref}}\right) - \text{Binary}\left(\boldsymbol{\alpha}, \text{PSF}, \boldsymbol{l}\right)\right|\right|^2 . \tag{22}$$

The least-squared procedure leads to retrieve the reference binary separation from :

$$\widehat{\boldsymbol{l}} = \underset{\boldsymbol{l}}{\text{argmin}}\, \varepsilon^2\left(\boldsymbol{l}_{\text{ref}}, \boldsymbol{l}\right) . \tag{23}$$

Finally, we define the relative astrometric accuracy as :

$$\Delta\text{astro} = \widehat{\boldsymbol{l}} - \boldsymbol{l}_{\text{ref}}. \tag{24}$$

In the future, we'll perform the identification of this term in introducing extra noise on the reference PSF for drawing plots of accuracy as function of the PSF signal to noise ratio.

5 Conclusions

We've detailed the pipeline for processing KECK NIRC2 images implemented into KASP and presented the *psfStats* class that gathers the PSF and all of its characteristics. We especially how we define the main scalar parameters and introduced two astrophysics metrics, the relative photometry and astrometry accuracy.

References

[1] E. Gendron and P. Léna. Astronomical adaptive optics. II. Experimental results of an optimized modal control. *Astron. & Astrophys.* , 111:153–+, May 1995.