PUAKO notes #04: hybrid PSF reconstruction with PRIME

O. Beltramo-Martin*

January 6, 2021

Contents

1 Rationale

PUAKO offers the possibility to explore an extended PSF reconstruction strategy, which is the hybrid PSF reconstruction. The rationale of this approach is to use jointly the AO telemetry and the focal-plane image to enhance the PSF reconstruction process, so as to be able to extrapolate the PSF accurately at any position in the field and wavelength. The PSF reconstruction process requires several system calibration (WFS optical gains for instance) and methods to identify the seeing for instance. Therefore, the point of the hybrid PSF reconstruction is to skip the calibration and estimation process and keep the required parameters as free degrees of freedom that are adjusted by comparing the PSF reconstruction output to the focal-plane image. Such a facility is provided by the algorithm PRIME [4, 3] currently implemented into PUAKO. PRIME is a model-fitting facility that relies on the *psfReconstruction* class to describe the PSF model and calls non-linear fitting algorithms of Matlab to adjust few atmosphere and system parameters. I present in this document how to use the *PRIME* class of PUAKO and details related to the model-fitting process wit illustration to Keck data.

*olivier.beltramo-martin@lam.fr

2 The PRIME class

The *PRIME* class is called by

```
psfp = prime(psfr,'aoinit',[],'aobounds',[],'MaxIter',300,'TolX',1e-12,'TolFun',1e-12,
    'MaxFunEvals',1e3,'InitDamping',1,'display','iter','weighting',false,'x_fixed',{[]},
'fitR0',true,'fitCn2',false,'fitGains',[true,true,true],'nZernMode',[],'fitStatZer',[],
    'fitBg',false,'flagJacobian',false)
```

where *psfr* is required the *psfReconstruction* object instantiated by the user or directly by using the PUAKO property `psfr`. Optional parameters are:

- `aoinit` and `aobounds` (2 rows) are arrays containing respectively the initial guess and hard-bounds on atmosphere and system-related parameters. If set to empty, PUAKO does instantiate its own initial guess and bounds.

- The minimization algorithm performance can be tuned by playing on `TolX` (minimal changes on parameters), `TolFun` (minimal changes on model), `MaxFunEvals` (maximal number of call to the model) and `InitDamping` (initial change in the meta-parameter of the Levenberg-Marquardt algorithm). By default, PUAKO shows the result of each iteration (`display` is true).

- The user can change the criterion by setting `weighting` to true to enable weighted minimization (see Sect. 3.1 and decides to adjust the r_0 (`fitR0` to true), the full C_n^2 profile (`fitCn2` to true) and the gains $g_{\rm ao}$, $g_{\rm tt}$ and $g_{\rm al}$ (`fitGains` set to [true,true,true]) (see the model description in Sect. 3.1). On top of that, PRIME offers the possibility to adjust model Zernike gains for the tip-tilt excluded AO corrected modes by giving `nZernMode` as a vector of Noll's indexes of Zernike mode to get a specific optical gain. If `nZernMode` = [4], PUAKO will retrieve one gain for the focus and another one for all the others modes. Furthermore, the user can also adjust additional static Zernike modes coefficients by setting `'fitStatZer'` as a list of Noll's indexes of modes. If put as [4], PUAKO will retrieves a focus static term on top of the existing static map defined in the *psfReconstruction* class. If `fitBg` is set to true, the criterion to be minimized includes a constant background value to be retrieved. Finally, `x_fixed` must be defined as a cell to provide which parameters to be kept fixed and to which value. For instance, setting `x_fixed` to *'gal'*, [1] will force the parameter $g_{\rm al}$ to remain equal to one.

- By default, the non-linear algorithm calculates the criterion gradient empirically, which demands to evaluate the model as many times as parameters to be retrieve for each iteration. I've tried to implement the Jacobian matrix derivation analytically, but some work must be pursued to finalize and test the implementation. If you trust the calculation, you can set the field `flagJacobian` to true.

Once again, the user does not need to instantiate the *PRIME* object. It is enough to use the *getPrimePSF* function of PUAKO as follows

```
p = puako('path_imag',path_imag,'path_trs',path_trs,'path_calibration',path_calib);
p.getPrimePSF({'n0004'},'fov',2*trs.cam.resolution,'fitbg',true)
p.psfp
```

PUAKO will automatically instantiate the *telemetry* and *psfReconstruction* objects in which will be performed the noise and seeing estimations (see PUAKO note #02) and the OTF and phase structure functions reconstruction (see PUAKO note #03). The adjusted image is accessible from the field `psf` of PRIME (p.psfp.psf) which is a *psfStat* object (see PUAKO note #05) containing the final reconstructed image (p.psfp.psf.image) with

scaled photometry and tuned astrometry, as well as metrics such as the Strehl-ratio or the FWHM. Estimated atmospheric parameters and gains can be found in the structure `atm_fit` (p.psfp.atm_fit) and `gains_fit` (p.psfp.gains_fit) respectively, while the final static aberration map information is in the property `map_fit` (p.psfp.map_fit). Finally, stellar parameters are gathered in the structure `catalog_fit` (p.psfp.catalog_fit) that contains the astrometry (x,y), photometry (flux,mag) as well as associated 3-σ precision. this structure contains also the fraction of variance unexplained (see PUAKO note #05).
Calling the *getPrimePsf* function or instantiating PRIME will automatically run the mode-fitting process by performing the following steps

1. Normalizing the detector image $\Im$ to get the sum of pixels equal to 1 and define the weight matrix by accounting for the read-out noise of variance σ^2_{e-} and the Poisson noise as follows

$$\begin{aligned} y_{ij} &= \frac{\Im_{ij}}{\sum_{ij}\Im_{ij}} = \alpha \times \Im_{ij} \\ w_{ij} &= \frac{1}{\sqrt{\max(y_{ij},0)/\alpha + \sigma^2_{e-}}}, \end{aligned} \tag{1}$$

   where σ^2_{e-} is directly identified on the normalized image by taking the stand-deviation of pixels at the four corners outside the circumscribe circle.

2. Defining the fitting options (*fittingOption* function) and the list of parameters to be retrieved as well as initial guess and hard bounds (*fittingSetup* function). Then PUAKO defines the image model (in file *imageModel.m*) and performs the non-linear minimization using the native Matlab function *lsqcurvefit*.

3. Update the PRIME object using the function *updateResults* that fills in the fields `atm_fit`, `gains_fit`, `map_fit`, `catalog_fit` and creates the *psfStats* object that contains the adjusted model and the figure of merits.

3 Model-fitting framework

3.1 Criterion

PRIME is designed to minimize the following image-based criterion

$$\mathcal{J}(\boldsymbol{\mu},\gamma,\boldsymbol{\alpha}) = \sum_{i,j}^{n_{\text{px}}} w_{ij}\left[\gamma\times\delta_{\boldsymbol{\alpha}} * h_{ij}(\boldsymbol{\mu}) - y_{ij} + \nu\right]^2 \tag{2}$$

where $. * .$ is the convolution product and

- h_{ij} and y_{ij} are the (i,j) pixel intensity values of respectively the numerical PSF model and sky observation,
- $\boldsymbol{\mu} = [C_n^2, \boldsymbol{g}_{\text{ao}}, g_{\text{tt}}, g_{\text{al}}, \boldsymbol{a}_z]$ is the set of parameters to be adjusted,
- $\delta_{\boldsymbol{\alpha}}$ is the Dirac distribution shifted by the astrometric position $\boldsymbol{\alpha}$ and scaled by the photometric factor γ,
- ν is an additional degree of freedom to account for a residual background,
- w_{ij} is the weighting coefficient for the (i,j) pixel.

The PSF model is obtained by the Fourier transform of the OTF model which is given by the following multiplication

$$\tilde{h}(\boldsymbol{\rho}/\lambda, \boldsymbol{\mu} = [C_n^2, \boldsymbol{g}_{\text{ao}}, g_{\text{tt}}, g_{\text{al}}, \boldsymbol{a}_z]) = \tilde{h}_{\text{tel}}(\boldsymbol{\rho}/\lambda, \boldsymbol{a}_z).\tilde{k}_{\text{atm}}(\boldsymbol{\rho}/\lambda, C_n^2(z), \boldsymbol{g}_{\text{ao}}, g_{\text{tt}}, g_{\text{al}}).\tilde{k}_{\text{det}}(\boldsymbol{\rho}/\lambda), \tag{3}$$

where

- $\tilde{h}_{\text{tel}}$ is the static OTF including the calibrated static aberrations, i.e. residual NCPA, field-dependent and static fitting aberrations (see PUAKO note #03), on top of which we add an aberrations described over n_{m} Zernike modes as follows

$$\begin{aligned}\tilde{h}_{\text{tel}}(\boldsymbol{\rho}/\lambda, \boldsymbol{a}_z) &= \iint_{\mathcal{P}} \mathcal{P}(\boldsymbol{r})\mathcal{P}^*(\boldsymbol{r}+\boldsymbol{\rho}) \exp\left(-i\phi_{\text{stat}}(\boldsymbol{r}) + i\phi_{\text{stat}}(\boldsymbol{r}+\boldsymbol{\rho})\right) d^2\boldsymbol{r} \\ \phi_{\text{stat}}(\boldsymbol{r}) &= \frac{2\pi}{\lambda}\left(\delta_{\text{calib}}(\boldsymbol{r}, \theta) + \sum_i^{n_{\text{m}}} a_z(i) Z_i(\boldsymbol{r})\right)\end{aligned} \tag{4}$$

- $\tilde{k}_{\text{atm}}$ is the residual turbulence spatial filter that is parametrized this way

$$\begin{aligned}&\tilde{k}_{\text{atm}}(\boldsymbol{\rho}/\lambda, C_n^2(l), \boldsymbol{g}_{\text{ao}}, g_{\text{tt}}, g_{\text{al}}) = \\ &\exp\left(-\frac{2\pi^2}{\lambda^2}\left(r_0^{-5/3}\left(\mathfrak{d}_\perp(\boldsymbol{\rho}) + g_{\text{al}}\mathfrak{d}_{\text{al}}(\boldsymbol{\rho})\right) + g_{\text{tt}}\mathcal{D}_{\text{tt}}(\boldsymbol{\rho}) + \sum_{j=1}^{n_{\text{g}}+1} \boldsymbol{g}_{\text{ao}}(j)\mathfrak{d}_{\text{ao}}^j(\boldsymbol{\rho}) + \sum_{l=1}^{n_{\text{l}}} C_n^2(z_l)\mathfrak{d}_\Delta^l(\boldsymbol{\rho})\right)\right),\end{aligned} \tag{5}$$

  where

  - $\mathfrak{d}_\perp$ and $\mathfrak{d}_{\text{al}}$ are respectively the DM fitting and aliasing structure function normalized to $r_0^{-5/3} = \sum_{l=1}^{n_{\text{l}}} C_n^2(z_l) = 1$. These structure function are rescaled to physical units by summing the adjusted C_n^2 profile discretized over n_{l} layers. The user can decide to keep the C_n^2 profile as the one measured by the MASS-DIMM and estimate the r_0 value only (`fitR0` is true and `fitCn2` is false). Note that in this situation, the retrieved r_0 will be different than the DIMM's, i.e. we assume that differences are due to dome seeing that does not impact the anisoplanatism calculation supposedly. On top of that, if `fitGains(3)` is set to true, the scalar factor g_{al} is fitted as well, which allows to mitigate any wind speed estimation problems that matters in the aliasing modeling (see PUAKO note #03).
  - $\mathcal{D}_{\text{tt}}$ is the tip-tilt structure function as defined in the PUAKO note #03. The scale gain g_{tt} allows to calibrate the TT WFS gain.
  - $\mathfrak{d}_{\text{ao}}^j$ is the structure function for a specific Zernike modes reconstructed from the AO telemetry and is obtained from

$$\begin{aligned}\text{if } j < n_{\text{g}} &\implies \mathfrak{d}_{\text{ao}}^j = \mathcal{V}_{ii}\left[\frac{4\pi^2}{\lambda^2}\boldsymbol{F}_j\left(C_{\text{ao}} - C_\eta\right)\boldsymbol{F}_j^t\right] \\ \text{if } j = n_{\text{g}}+1 &\implies \mathfrak{d}_{\text{ao}}^j = \mathcal{D}_{\text{ao}} - \sum_j^{n_{\text{g}}} \mathfrak{d}_{\text{ao}}^j,\end{aligned} \tag{6}$$

    where $\mathcal{V}_{ii}$ is the operator that applied the Vii algorithm [7] to obtained the point-wise phase structure function from the modal covariance matrix, $\boldsymbol{F}_j$ is the filtering matrix that filters any others Zernike modes than the j^{th} one in the list provided by the user and C_{ao} and C_η are respectively the residual phase and noise covariance matrix in the DM actuators space (see PUAKO note #03). The Filtering matrix is defined as

$$\begin{aligned}\boldsymbol{F}_j &= \boldsymbol{P}_j\boldsymbol{P}_j^{-\dagger} \\ \boldsymbol{P}_j &= \boldsymbol{M}^{-\dagger}Z_j,\end{aligned} \tag{7}$$

with $\boldsymbol{M}$ the DM influence function matrix, Z_j the j^{th} Zernike mode ordered in a column vector and $\boldsymbol{X}^{-\dagger}$ the generalized inverse of matrix $\boldsymbol{X}$ performed thanks to the *pinv* function. If the user provides three values for $\boldsymbol{g}_{\text{ao}}$ and set `nZernMode` to [4,5,6], PUAKO will estimate a specific gains for focus and astigmatism terms and fourth one for any other modes. Regardless, the retrieved gains, the sum of $\mathfrak{d}_{\text{ao}}^j$ functions is forced to be $\mathcal{D}_{\text{ao}}$.

- $\mathfrak{d}_{\Delta}^l$ is the anisoplanatic structure function defined from a sole layer at altitude z_l and with a normalized energy $C_n^2(z_l) = 1$. It includes both focal-angular and tip-tilt anisoplanatism [1, 6].

There is no limit in the number of parameters PUAKO can try to retrieve. Of course, I recommend to limit this number to a reasonable value depending on how many stars can be used to perform the fit and the observing conditions. For a single star on-axis, it is fine to estimate stellar parameters with background (4 parameters) with the r_0 and three gains (4 parameters), which means 8 parameters to fit. If necessary, we can request to adjust few static Zernike modes or modal gains, but on-sky tests showed that it is not necessary to fit them to improve results compared to forward reconstruction. For LGS cases, we could potentially retrieve a C_n^2 profile using the focal anisoplanatism that blurs the PSF. I would suggest to model the profile over few bins, like 2 or 3, but not more. In practice, I don't think the anisoplanatism model is the main limitation, at least for LGS data sets I've treated so far. I've never got the chance to test PUAKO on off-axis observations, so I did not try to estimate the C_n^2 from anisoplanatic PSFs. If it turns out to be feasible one day, the discretization should be set up regarding the observing conditions and the PSF position in the field. For a 10"×10" field of view, I think it's sufficient to describe the profile over three bins only [5, 2, 1].

Finally, the user can fix some of the parameters and attribute any value using the `x_fixed` property when calling PRIME. Those parameters are: 'r0' (scalar), 'cn2' (vector), 'gao' (vector), 'gtt' (scalar), 'gal' (scalar), 'az' (vector) and 'staticmap' (boolean). If 'staticmap' is set to false (true by default), PUAKO does not use the calibrated static map in the PSF model, i.e. δ_{calib} is set to zero. To perform a fit with an additional static focus term of 100 nm, the command is

```
p = puako('path_imag',path_imag,'path_trs',path_trs,'path_calibration',path_calib);
p.getPrimePSF({'n0004'},'fitStatZer',[4],'x_fixed',{'az',[100]})
```

3.2 Non-linear solver

Two algorithms can be deployed to solve the non-linear criterion by estimating iteratively the update parameters from iteration t to iteration $t+1$. This step is usually done from the following approximation

$$h(\boldsymbol{\mu}_{t+1}) \simeq h(\boldsymbol{\mu}_t) + \mathbf{J}(\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t) \tag{8}$$

where $\mathbf{J}$ is the Jacobian matrix of the model $\boldsymbol{h}$ in $\boldsymbol{\mu}_t$. This Jacobian matrix is calculated empirically by default. Providing an analytical expression of th Jacobian would speed up drastically the minimization (that takes less than 30 s-1 mn usually), but such calculations are not yet implemented for all parameters. Taking the derivative with respect to $\boldsymbol{\mu}_{t+1}$, we obtain that

$$(\mathbf{J}^T\mathbf{J})(\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t) \simeq \mathbf{J}^T(\boldsymbol{y} - h(\boldsymbol{\mu}_t)) \tag{9}$$

where $\boldsymbol{y}$ is the observation.
which are

- *Levenberg-Marquardt algorithm (LMA)*. This algorithm combines the Gauss-Newton and Gradient-descent techniques to iteratively update the PSF model parameters until convergence. This one is supposed to be more stable than the Gauss-Newton algorithm but can be potentially slower. The LMA approximates Eq. 9 by damping it using a damping factor Υ as follows

$$(\mathbf{J}^T\mathbf{J} + \Upsilon \mathrm{diag}(\mathbf{J}^T\mathbf{J}))(\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t) \simeq \mathbf{J}^T(\boldsymbol{y} - h(\boldsymbol{\mu}_t)) \tag{10}$$

where Υ is updated at every iteration regarding how fast goes the minimization. For fast diminishing of the criterion $\mathcal{J}$, the minimization is efficient and the damping factor is set to low values (Gauss-Newton-like algorithm). On the contrary, for less efficient iteration, a larger value of Υ is chosen. The user can change the initial damping value that is set up to 10^{-5} by default, which looks to be the value that provides the convergence in a minimal number of iterations. Over cases I've tested, playing with this value has only changed the convergence speed, not the found solution. Finally, I did see only marginal difference with the TRRA in terms of convergence speed and retrieved solution. I recommend rather to use the TRRA one as it offers the possibility to bound the solution space so as to introduce a regularization.

- *Trust-region-reflective algorithm (TRRA)*. This is the default and Matlab-recommended algorithm for solving non-linear criteria. Contrary to the LMA, the TRRA will solve the quadratic form of the PSF model under the following constrain

$$\begin{aligned} &\min{}_{\boldsymbol{\mu}_{t+1}} \quad h(\boldsymbol{\mu}_t) + \mathcal{G}_t^T(\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t) + \frac{1}{2}(\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t)^T\mathcal{B}_t(\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t) \\ &\left\|\boldsymbol{\mu}_{t+1} - \boldsymbol{\mu}_t\right\|_2 < \Delta_t, \end{aligned} \tag{11}$$

where $\mathcal{G}_t^T$ is the PSF model gradient a the current iterate, $\mathcal{B}_t$ is a symmetric matrix which approximates the Hessian of the PSF model and Δ_t is the trust region radius that is updated iteratively. The TRRA should exhibit better performances each time a negative curvature is encountered and have thus better performances than the LMA.

3.3 Some ideas to derive the Jacobian matrix analytically

If `flagJacobian` is set to true, the user must ensure that the function *imageModel* returns the image model Jacobian matrix as a second outputs. According to Eq. 2, the Fourier transform of the image model PUAKO relies is

$$\tilde{I}(\boldsymbol{\rho}/\lambda) = \gamma\tilde{h}(\boldsymbol{\rho}/\lambda, \boldsymbol{\mu})\exp\left(-i\pi(\rho_x\alpha_x + \rho_y\alpha_y)\right) \tag{12}$$

where α_x, α_y are the x/y PSF displacement. We can derive the partial derivatives of the image model in the Fourier domain. With respect to stellar parameters, we have

$$\begin{aligned} \frac{\partial\tilde{I}(\boldsymbol{\rho}/\lambda, \boldsymbol{\mu})}{\partial\gamma} &= \tilde{I}(\boldsymbol{\rho}/\lambda, \boldsymbol{\mu})/\gamma \\ \frac{\partial\tilde{I}(\boldsymbol{\rho}/\lambda, \boldsymbol{\mu})}{\partial\alpha_x} &= -i\pi\tilde{I}(\boldsymbol{\rho}/\lambda, \boldsymbol{\mu})\rho_x \\ \frac{\partial\tilde{I}(\boldsymbol{\rho}/\lambda, \boldsymbol{\mu})}{\partial\alpha_y} &= -i\pi\tilde{I}(\boldsymbol{\rho}/\lambda, \boldsymbol{\mu})\rho_y. \end{aligned} \tag{13}$$

With respect to AO parameters, we have

$$
\begin{aligned}
\frac{\partial \tilde{I}(\boldsymbol{\rho}/\lambda,\boldsymbol{\mu})}{\partial r_0^{-5/3}} &= -\frac{2\pi^2}{\lambda^2}\tilde{I}(\boldsymbol{\rho}/\lambda,\boldsymbol{\mu})\left(\mathfrak{d}_\perp(\boldsymbol{\rho}) + g_{\mathrm{al}}\mathfrak{d}_{\mathrm{al}}(\boldsymbol{\rho})\right)\\
\frac{\partial \tilde{I}(\boldsymbol{\rho}/\lambda,\boldsymbol{\mu})}{\partial g_{\mathrm{al}}} &= -\frac{2\pi^2 r_0^{-5/3}}{\lambda^2}\tilde{I}(\boldsymbol{\rho}/\lambda,\boldsymbol{\mu})\mathfrak{d}_{\mathrm{al}}(\boldsymbol{\rho})\\
\frac{\partial \tilde{I}(\boldsymbol{\rho}/\lambda,\boldsymbol{\mu})}{\partial g_{\mathrm{tt}}} &= -\frac{2\pi^2}{\lambda^2}\tilde{I}(\boldsymbol{\rho}/\lambda,\boldsymbol{\mu})\mathcal{D}_{\mathrm{tt}}(\rho)\\
\frac{\partial \tilde{I}(\boldsymbol{\rho}/\lambda,\boldsymbol{\mu})}{\partial \boldsymbol{g}_{\mathrm{ao}}(j)} &= -\frac{2\pi^2}{\lambda^2}\tilde{I}(\boldsymbol{\rho}/\lambda,\boldsymbol{\mu})\mathfrak{d}_{\mathrm{ao}}^{j}(\boldsymbol{\rho})\\
\frac{\partial \tilde{I}(\boldsymbol{\rho}/\lambda,\boldsymbol{\mu})}{\partial C_n^2(z_l)} &= -\frac{2\pi^2}{\lambda^2}\tilde{I}(\boldsymbol{\rho}/\lambda,\boldsymbol{\mu})\mathfrak{d}_{\Delta}^{l}(\boldsymbol{\rho}).
\end{aligned}
\tag{14}
$$

With respect to the static Zernike modes

$$
\begin{aligned}
\frac{\partial \tilde{I}(\boldsymbol{\rho}/\lambda,\boldsymbol{\mu})}{\partial a_z(i)} = -\frac{2i\pi\gamma}{\lambda}&\iint_{\mathcal{P}}\mathcal{P}(\boldsymbol{r})\mathcal{P}^*(\boldsymbol{r}+\boldsymbol{\rho})\exp\left(-\phi_{\mathrm{stat}}(\boldsymbol{r}) + \phi_{\mathrm{stat}}(\boldsymbol{r}+\boldsymbol{\rho})\right)\left(Z_i(\boldsymbol{r}) - Z_i(\boldsymbol{r}+\boldsymbol{\rho})\right)d^2\boldsymbol{r}\\
&\tilde{k}_{\mathrm{atm}}(\boldsymbol{\rho}/\lambda,\boldsymbol{\mu})\tilde{k}_{\mathrm{det}}(\boldsymbol{\rho}/\lambda,\boldsymbol{\mu})\exp\left(-i\pi(\rho_x\alpha_x + \rho_y\alpha_y)\right)
\end{aligned}
\tag{15}
$$

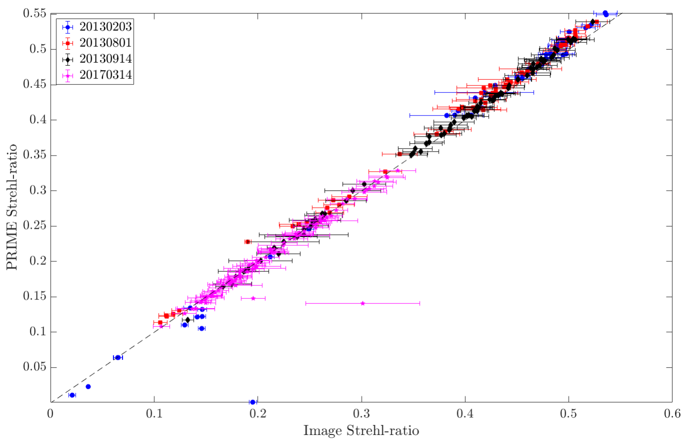
However, as the criterion is an image-based quantity, we have to derive the Jacobian matrix of the image and not of its Fourier transform. The problem is that for non-Nyquist sampled PSFs, the PSF is obtained through a game of zero-padding or interpolation from the OTF model give in Eq. 3 and I'm not so sure about how we can substitute the Fourier and partial derivative operators.

A way to skip this issue would to define the criterion in the Fourier domain, but one must ensure that the shape of the criterion suits to parameters retrieval. For instance, the anisoplanatism, all aberrations generally, sharpens the OTF, meaning that large errors spread over less pixels in the Fourier domain than in the direct plan, while we would like to ensure to have a maximal sensitivity to those large errors that impact significantly the PSF shape. So far, I've implemented the Jacobian derivation following previous equation and applying the Fourier operator to partial derivatives, which is probably not completely accurate. I'd grateful to have someone spending sometimes to look whether it is feasible or not. Keep in mind that the interest would be to speed up the minimization with PRIME , by a factor 5 up to 10 I think according to some preliminary tests. Knowing that the current version takes less than 1 mn using the empirical Jacobian matrix, I do not think algorithm speed is a priority so far, at least there are plenty of stars we want to use simultaneously to calibrate the PSF model.

4 Application to Keck data

4.1 Hybrid PSF reconstruction

Results of PRIME are already published with applications to Keck [3] and SPHERE/ZIMPOL [4]. To summarize, going hybrid allows the PSF reconstruction to reach ~1% of accuracy on PSF metrics rather than 3 up to 10% with classical PSF reconstruction. On top of that, PRIME reaches 1.4% overall the 304 data sets comparatively to 2.4% with PSF-R and 4.8% with a Gaussian model. When considering NGS data only, these numbers go down to 1%, 1.8% and 5.0%, meaning that thanks to PRIME, we can reduce the overall residual by a factor 5. Moreover, PSF-R reaches already a good performance in NGS mode, claiming for concentrating the efforts on the LGS case. The Gaussian function does not fit the NGS PSF as well as the LGS PSF, which traduces that the LGS PSFs we've got were farther from the diffraction and so more blurry. As discussed in the PUAKO note #03, some work must be continued to understand the system limitation in LGS mode, which will also help improving PSF reconstruction.



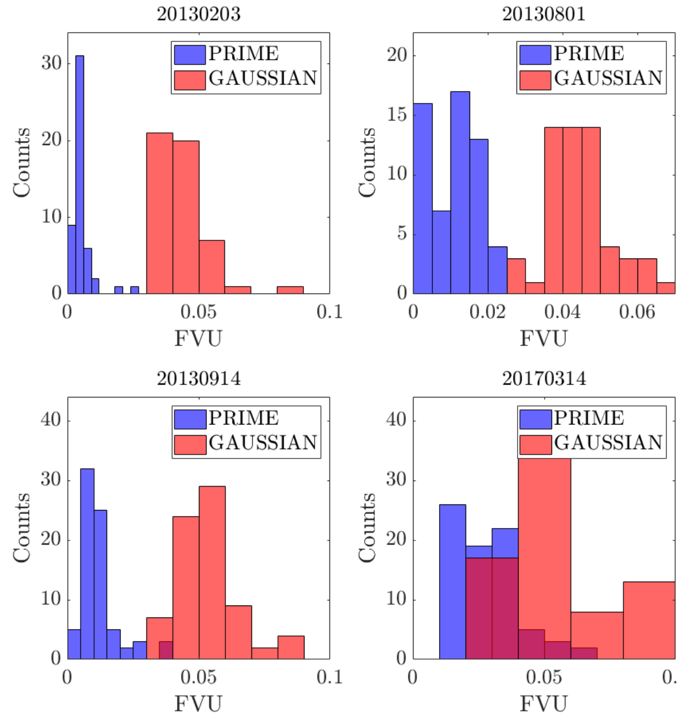


Figure 1: Top: Reconstructed Strehl-ratio with PRIME versus image-measured one. Right: Fraction of variance unexplained histograms obtained with PRIME and a Gaussian model.

4.2 Focus estimation

On top of the 304 data sets I've processed, I have also some phase diversity data with out of focus images. Obviously, as the focus is introduced in a way the WFS does not compensate for it (through WFS reference slopes or by moving the scientific camera), the classical PSF reconstruction can not reconstruct this focus. However, we have the focal-plane image and we can play the game of retrieving the amplitude of this focus with PRIME by requesting to fit the a_z coefficient that appears in Eq. 4 for a focus Zernike mode.

I present in Fig. 2 the results of such test when requesting PRIME to seek for the focus amplitude and r_0 value only (no fit of gains) for different settings. The retrieved PSF matches quite well the observation, claiming for a good focus estimation. I've tried also to adjust simultaneously gains and focus terms but it turns out that g_{ao} were set to meaningless tiny values, suggesting unsurprisingly that the focus is too large to be able to detect the structures on which PRIME could rely on for gains estimation. An interesting fact is that I've got really different retrieved solutions using the LMA or the TRRA. The first one was finding $g_{ao} = 0.28$, that is very small compared to tests on in focus PSFs for which I retrieve a median value of $g_{ao} = 1$, while the TRRA ended up with $g_{ao} = 0$. I summarize in Tab. 1 the outputs of retrieved parameters for case n0016 using the LMA or TRRA and with gains adjustment or not (fixed to one). When keeping gains fixed, both LMA and TTRA reach the very same solution with a r_0 estimates that is closed to ones obtained on in-focus images at the same date. However, results are really different when trying to estimate gains on such blurry images and retrieved gains are meaning less and very far away to ones estimated on in-focus images. My conclusion is that it is pointless to estimate jointly gains and focus term in such a configuration where the static aberrations overdominates the PSF shape compared to the dynamic residual turbulence.

For the case n0016, the focus term is estimated to 978.3 μm, while the reference value obtained from the header field containing the camera stage z position(field OBWF and AOFCNGFO for NGS or AOFCLGFO for LGS) is 991.9 μm knowing that we should have 0.198 μm/mm. Therefore, I tried to adjust gains and r_0 with a fixed value of focus term set up to 978.3 μm, 983.0 μm or 991.9 μm. My guess is that we can mitigate the image

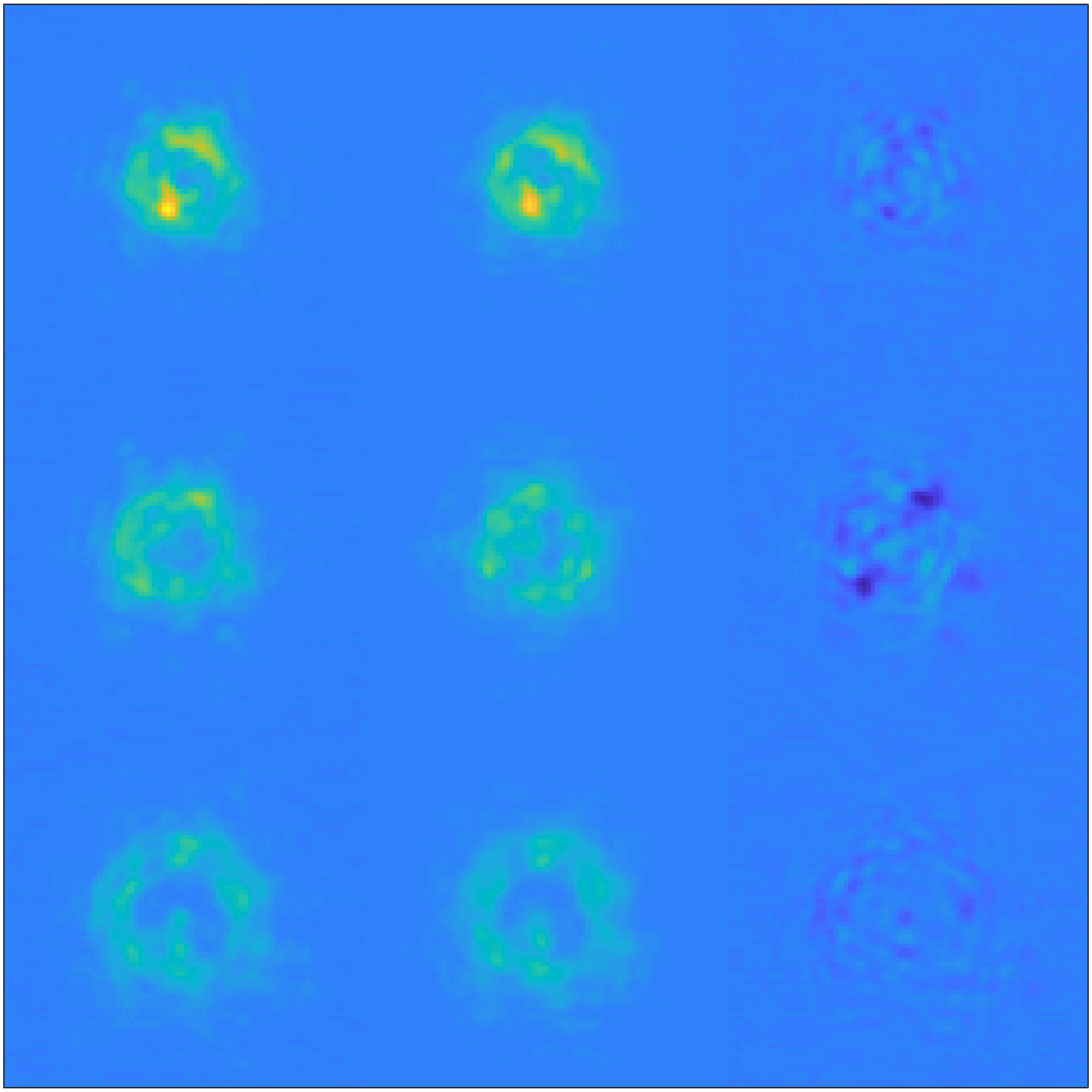


Figure 2: **Left**: out-of-focus image taken on-sky. **Middle**: reconstructed image with PRIME using AO telemetry with a fit of the r_0 and the focus amplitude (starting from 0 nm). **Right**: residual map. **From top to bottom**: case n0118, August 1st 2013, 630 μm of focus, case n0026, February 3rd 2013, 790 μm of focus, case n0016, August 1st 2013, 1 mm of focus.

gains sensitivity issues in presence of strong focus when regularizing the solution by keeping the focus constant. When choosing of of the two first values, the TRRA retrieves a $g_{\mathrm{ao}} = 0.12$, which is not yet consistent with in-focus -based estimation but much meaningful comparatively to the joint estimation attempt. However, when introducing a focus of 991.9 μm, results are slightly worse with $g_{\mathrm{ao}} = 0.05$ and a larger fraction of variance unexplained. Consequently, the focus term with measured with PRIME is really 978 μm ± 2.5 μm, which is slightly different from the reference value.

I report in Fig. 3 measured focus amplitude versus expectations for a total of 41 data sets. The figure shows that the measured focus is well correlated with the calibration, although some discrepancies exist. As I did not calibrate the optical gains simultaneously, the focus estimation is perhaps sensitive to gains mis-calibration,

Table 1: Comparison of retrieved parameters using either the LMA or TRRA and with gains fitting or not. Results are obtained for the single case of n0016 (out of focus image). Moreover, precision obtained from the Jacobian matrix and the residual map on the focus term was 2.5 μm.

	r_0 (cm)	g_{ao}	g_{tt}	g_{al}	a_4 (μm)	x (mas)	y (mas)	γ (ADU)	ν (ADU)
LMA	13.9	0.28	4.56	-1.68	-983.4	11.5	-11.1	$1.20 10^{7}$	-60.0
TRRA	15.8	$2 10^{-14}$	4.60	$8 10^{-9}$	-983.0	11.5	-11.1	$1.18 10^{7}$	-50.3
LMA Gains fixed	20.0	1	1	1	-978.3	11.2	-11.7	$1.14 10^{7}$	-35.2
TRRA Gains fixed	20.0	1	1	1	-978.3	11.2	-11.7	$1.14 10^{7}$	-35.2

especially for low values of focus for which the residual turbulence start playing a more significant role on the PSF. This can be easily verified using the internal fiber source to generate in and out of focus PSF with no turbulence. The point of this analysis on the focus retrieval was to ensure that PRIME can effectively retrieve a meaningful amplitude of static aberrations in such a scenario. Of course, with in-focus images and a proper preliminary static aberrations retrieval, there is no reason to seek for additional aberrations. If they are quasi-static, we might attempt to adjust low-order modes, but I do not know how much they should vary to reach enough sensitivity. Another application could be the calibration of field-dependent static aberrations and especially the field-curvature we see with the internal fiber [5, 8]. Using multiple PSFs across the field and a model of the anisoplanatism, it is feasible to adjust a field curvature term, i.e. a focus that varies quadratically with respect to the distance on-axis. I hope to get access to some off-axis data one day to have the chance to make this experiment.

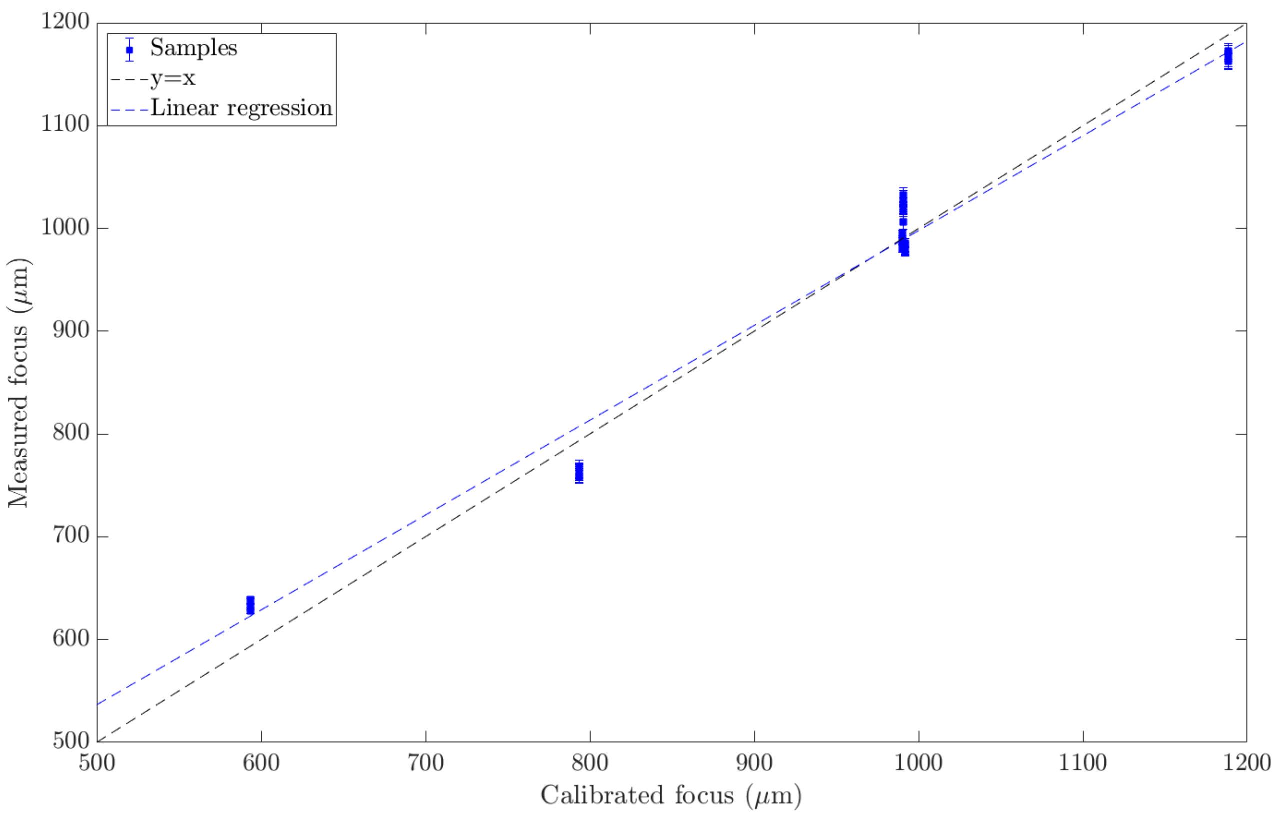


Figure 3: Retrieved focus from PRIME versus value obtained from the header and the calibration factor of 0.198 μm/mm. The black dashed line is the $y = x$ line while the blue dashed line is the 1st order linear regression.

References

[1] O. Beltramo-Martin, C. M. Correia, E. Mieda, B. Neichel, T. Fusco, G. Witzel, J. R. Lu, and J.-P. Véran. Off-axis point spread function characterization in laser guide star adaptive optics systems. *M.N.R.A.S* , 478:4642–4656, August 2018.

[2] O. Beltramo-Martin, C. M. Correia, B. Neichel, and T. Fusco. Focal-plane C_n^2(h) profiling based on single-conjugate adaptive optics compensated images. *M.N.R.A.S* , 481:2349–2360, December 2018.

[3] O. Beltramo-Martin, C. M. Correia, S. Ragland, L. Jolissaint, B. Neichel, T. Fusco, and P. L. Wizinowich. PRIME: PSF Reconstruction and Identification for Multiple-source characterization Enhancement - application to Keck NIRC2 imager. *M.N.R.A.S* , 487(4):5450–5462, Aug 2019.

[4] O. Beltramo-Martin, T. Fusco, B. Neichel, A Marasco, D Massari, and Milli J. Pushing Point spread function reconstruction to the next level. Application to SPHERE/ZIMPOL. *M.N.R.A.S* , page 1256, Dec 2019.

[5] O. Beltramo-Martin, G. Witzel, A. Ciurlo, P. Turri, and J. Lu. Sensitivity of astrometry and photometry estimation to spatial variations of the PSF. In *Sixth International Conference on Adaptive Optics for Extremely Large Telescopes*, page 70, November 2019.

[6] R Flicker. PSF reconstruction for Keck AO. Technical report, W.M. Keck Observatory, 65-1120 Mamalahoa Hwy, Waimea, HI 96743, United-States, 2008.

[7] E. Gendron, Y. Clénet, T. Fusco, and G. Rousset. New algorithms for adaptive optics point-spread function reconstruction. *Astron. & Astrophys.* , 457:359–363, October 2006.

[8] B. N. Sitarski, G. Witzel, M.P. Fitzgerald, L. Meyer, A. M. Ghez, R. D. Campbell, J. R Lu, K. Matthews, P. Wizinowich, and J. Lyke. Modeling instrumental field-dependent aberrations in the nirc2 instrument on the keck ii telescope. In *Adaptive Optics Systems IV*, volume 9148 of *Proc. SPIE* , August 2014.