

KASP Technical notes #09: Focal plane profiler with KASP: concept and results on the HeNOS bench.

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

Contents

1	Purpose	1
2	The FPP concept	2
2.1	Motivations	2
2.2	PSF model	3
2.2.1	Fitting model	3
2.2.2	Aliasing model	3
2.2.3	Anisoplanatism model	4
2.3	Algorithm description	5
3	Results on the NRC-HIA Henos testbed	6
3.1	System description	6
3.2	How to use the FocalPlaneProfiler class	6
3.3	FPP results	12
4	Conclusions	12

1 Purpose

We describe the Focal Plane Profiler (FPP) concept, as a model-fitting algorithm handling $C_n^2(h)$ values to minimize residual on PSF reconstruction. We detail the principle of the method of results obtained on HeNOS bench data.

*olivier.martin@lam.fr

2 The FPP concept

2.1 Motivations

PSF reconstruction involves two capabilities: the first one requires to handle AO telemetry to reconstruct the PSF on-axis, while the second one consists in extrapolating the PSF anywhere across the field. This latest step is based on the atmospheric turbulence description as the seeing value, $C_n^2(h)$ and outer scale profiles. The common approach takes these information as priors, either provided by an external profiler (MASS/DIMM, SLODAR or SCIDAR for instance) or estimated through AO telemetry for multiple-WFS based AO systems.

PSF-R must achieve a proper PSF reconstruction for science data processing purpose. In particular we aim at improving astrophysics estimates, such photometry and astrometry, by providing a better estimation of what the PSF is really is, compared to focal-plane methods, that could suffer by sources overlapping and confusion noise.

However focal plane images contain the information that really matters for, i.e. the PSF. There's no point in performing PSF-R if the output product is far from what we observe at the science instrument focal plane.

Here comes the idea of gathering AO telemetry and focal-plane images to produce the best estimation of the PSF. Basically, we propose a *Learn & Apply* fashion for improving the reconstruction based on three steps :

- Produce a reconstruction of the on-axis PSF from $C_n^2(h)$ non-scaled terms, i.e. telemetry and NCPA residual.
- Define a PSF model as the convolution of the on-axis PSF with a $C_n^2(h)$ -dependent filter, including anisoplanatism (angular, focal and tip-tilt), fitting error (PSF wings) and aliasing error. The two latest terms are only r_0 -dependent. Then we least-square minimize the residual produced by the observations/model difference in fitting the $C_n^2(h)$ profile.
- We derive the closest version of the model we have in injecting retrieved fitted parameters from the previous step.

Several points can be discussed for optimizing this procedure. The way the problem is posed needs to be inverted and requires specific method. For the sake of simplicity of make sure the concept is feasible, we've opted for a model-fitting procedure but alternative option exists. In particular it may be possible to derive the cost function gradient with respect to the $C_n^2(h)$ profile.

In any case this problem need to be regularized to achieve a proper estimation. The way is done in the actual version of the FPP is to concatenate several measured PSF into a single array. For PSF discretized on $N \times N$ samples, we finally dispose of a meta array sizing $N \times (N \times n_{\text{src}})$ with n_{src} the number of picked-off PSFs from the focal plane image. In particular, NCPA residual may cause PSF core elongation that could confuse the model-fitting procedure. Handling several PSF across the field is a way to mitigate this problem. Also, including the on-axis PSF, what is only seeing-dependent, observed through the focal-image would make the seeing estimation more robust.

2.2 PSF model

The PSF model we're working with is the usual one based on Optical transfer function (OTF) multiplication as:

$$\text{OTF}_\Delta(\mathbf{u}/\lambda, C_n^2(h)) = \text{OTF}_0(\mathbf{u}/\lambda) \times \text{OTF}_\perp(\mathbf{u}/\lambda, r_0) \times \text{OTF}_{\text{Alias}}(\mathbf{u}/\lambda, r_0) \times \text{ATF}_\Delta(\mathbf{u}/\lambda, C_n^2(h)), \quad (1)$$

where the Fried's parameter r_0 is derived from the integral of $C_n^2(h)$ along altitude :

$$r_0 = \left(0.423 \times \left(\frac{2\pi}{\lambda} \right)^2 \times \frac{1}{\cos(\gamma)} \times \int_{\mathcal{R}} C_n^2(h_l) dh \right)^{-3/5}, \quad (2)$$

with γ the zenithal angle. See the KASP note #03 more description. Our point is to get a fast-computational model depending on the $C_n^2(h)$ values. We detail each model component herein.

If assuming altitude layers are known, or discretized at a desired sampling along altitude, we get as many parameters as number of layer we consider to get the $C_n^2(h)$ profile. Also, we may include the outer scale profile, or at least an integrated value kept constant for each altitude. In the following we've considered the outer scale as infinite on Henos, meaning we assume outer scale effect are not impacting PSF. For further deployment of this method on NIRC2 wide-field observations, outer scale must be considered. The FPP algorithm embedded into KASP offers the possibility to fit an outer scale profile or integrated value and layers heights as well.

2.2.1 Fitting model

We base the computation of this term on a Fourier approach, consisting in modeling the perpendicular phase PSD as the atmosphere Von-Kármán PSD released from low spatial frequencies lower than k_c the DM-cut off frequency :

$$\text{OTF}_\perp = \exp(C_\perp(\boldsymbol{\rho}) - C_\perp(0)), \quad (3)$$

where $C_\perp(\boldsymbol{\rho})$ is the covariance map derived by the phase PSD using the Wiener-Khintchine theorem :

$$C_\perp(\boldsymbol{\rho}) = \mathcal{F} \left[\tilde{\Phi}_{\text{atm}}(\mathbf{k}) \times \Pi_{k_c} \right], \quad (4)$$

with Π_{k_c} the gate function in the frequency domain :

$$\Pi_{k_c}(\mathbf{k}) = \begin{cases} 0 & \text{if } k_x \leq k_c \mid k_y \leq k_c \\ 1 & \text{else} \end{cases} \quad (5)$$

2.2.2 Aliasing model

Aliasing error comes from the aliased phase that creates a signal and that is reconstructed onto the DM commands and affects the in-band correcting part of the AO PSF. We follow the same procedure as for the fitting error :

$$\text{OTF}_{\text{Alias}} = \exp(C_{\text{Alias}}(\boldsymbol{\rho}) - C_{\text{Alias}}(0)), \quad (6)$$

where the covariance map is derived from :

$$C_{\text{Alias}}(\boldsymbol{\rho}) = \mathcal{F} \left[\tilde{\Phi}_{\text{Alias}}(\mathbf{k}) \right] \quad (7)$$

The aliased phase results from the spatial sampling of the phase every $1/2d$. The aliasing PSD is thus a sum of the initial spectrum shifted by an integer time $1/d$. The calculation includes the filtering through the WFS

reconstructor $\tilde{\mathbf{R}}$ and the AO system temporal transfer function $\tilde{\mathcal{H}}$. Taking into account the latter term is possible when associating each temporal frequency to a spatial one by $f = k \times v$ where v is the wind speed and for a Taylor-assumed turbulence. We end up with the final expression :

$$\tilde{\Phi}_{\text{Alias}}(\mathbf{k}) = |\tilde{\mathbf{R}}(k_x, k_y) \times \tilde{\mathcal{H}}(k_x \cdot v_x, k_y \cdot v_y)|^2 \times \left(\sum_{p=1}^{\infty} \sum_{q=1}^{\infty} \tilde{\Phi}_{\text{atm}}(k_x - p/d, k_y - q/d) \right). \quad (8)$$

2.2.3 Anisoplanatism model

A complete description of the anisoplanatism model is given in the KASP note#02. The Anisoplanatism transfer function (ATF) involved in Eq. 1 is derived in a 4D fashion from the covariance of the anisoplanatic phase as :

$$\text{ATF}_{\Delta}(\mathbf{u}/\lambda, C_n^2(h)) = \text{OTF}_{\text{DL}}^{-1}(\mathbf{u}/\lambda) \iint_{\mathcal{R}^2} \mathcal{P}(\mathbf{r}) \mathcal{P}(\mathbf{r} + \mathbf{u}) \times \exp \left[C_{\Delta}(\mathbf{r}, \mathbf{r} + \mathbf{u}, C_n^2(h)) - C_{\Delta}(0, 0, C_n^2(h)) \right] d^2 \mathbf{r}, \quad (9)$$

when OTF_{DL} is the diffraction-limit OTF derived from the pupil function \mathcal{P} :

$$\text{OTF}_{\text{DL}}(\mathbf{u}/\lambda) = \iint_{\mathcal{R}^2} \mathcal{P}(\mathbf{r}) \mathcal{P}(\mathbf{r} + \mathbf{u}) d^2 \mathbf{r}. \quad (10)$$

The anisoplanatic phase is defined as the turbulent phase, free from spatial frequencies not compensable by the DM, in the science direction subtracted from the residual phase in the guide star direction :

$$\phi_{\Delta}(\mathbf{r}) = \phi_1(\mathbf{r}) - \phi_2(\mathbf{r}), \quad (11)$$

what is supposed to be a stochastic process following a Gaussian distribution to write the ATF as it's done in Eq. 9, which is a reasonable assumption considering the Central limit theorem. The anisoplanatic covariance $C_{\Delta}(\mathbf{r}_1, \mathbf{r}_2)$ takes the following form :

$$\begin{aligned} C_{\Delta}(\mathbf{r}_1, \mathbf{r}_2) &= \langle \phi_{\Delta}(\mathbf{r}_1) \phi_{\Delta}^*(\mathbf{r}_2) \rangle \\ &= \langle \phi_1(\mathbf{r}_1) \phi_1^*(\mathbf{r}_2) \rangle + \langle \phi_2(\mathbf{r}_1) \phi_2^*(\mathbf{r}_2) \rangle - \langle \phi_1(\mathbf{r}_1) \phi_2^*(\mathbf{r}_2) \rangle - \langle \phi_2(\mathbf{r}_1) \phi_1^*(\mathbf{r}_2) \rangle \\ &= C_{11}(\mathbf{r}_1, \mathbf{r}_2) + C_{22}(\mathbf{r}_1, \mathbf{r}_2) - C_{12}(\mathbf{r}_1, \mathbf{r}_2) - C_{21}(\mathbf{r}_1, \mathbf{r}_2), \end{aligned} \quad (12)$$

where angular covariance matrix are derived using a point-wise description of the problem based on Von-K/'arm/'an statistics.

We define $\varphi_1(\mathbf{r}, l)$ as the phase in direction 1 and at an altitude h_l normalized to an unitary $C_n^2(h_l)$ value. We have so :

$$\phi_{\Delta}(\mathbf{r}) = \sum_{l=1}^{n_l} C_n^2(h_l) \times (\varphi_1(\mathbf{r}, l) - \varphi_2(\mathbf{r}, l)). \quad (13)$$

Considring there's no correlation between phase screen at different heights, the anisoplanatic covariance can be rewritten as a sum of covariance given at each layer as :

$$\begin{aligned} C_{\Delta}(\mathbf{r}_1, \mathbf{r}_2, C_n^2(h)) &= \sum_{l=1}^{n_l} (C_n^2(h_l))^2 \times \underbrace{(\langle \varphi_1(\mathbf{r}_1, l) \varphi_1^*(\mathbf{r}_2, l) + \varphi_2(\mathbf{r}_1, l) \varphi_2^*(\mathbf{r}_2, l) - \varphi_1(\mathbf{r}_1, l) \varphi_2^*(\mathbf{r}_2, l) - \varphi_2(\mathbf{r}_1, l) \varphi_1^*(\mathbf{r}_2, l) \rangle)}_{f_{\Delta}(\mathbf{r}_1, \mathbf{r}_2, l)} \\ &= \sum_{l=1}^{n_l} (C_n^2(h_l))^2 \times f_{\Delta}(\mathbf{r}_1, \mathbf{r}_2, l), \end{aligned} \quad (14)$$

where $f_{\Delta}(\mathbf{r}_1, \mathbf{r}_2, l)$ is the anisoplanatic covariance given for the l^{th} layer at altitude h_l and normalized to $C_n^2(h_l) = 1$. Injecting Eq. 14 into Eq. 9 leads to make appear a product with the integral :

$$\text{ATF}_{\Delta}(\mathbf{u}/\lambda, C_n^2(h)) = \text{OTF}_{\text{DL}}^{-1}(\mathbf{u}/\lambda) \iint_{\mathcal{R}^2} \prod_{l=1}^{n_l} \mathcal{P}(\mathbf{r}) \mathcal{P}(\mathbf{r} + \mathbf{u}) \times \exp \left[(C_n^2(h_l))^2 \times F_{\Delta}(\mathbf{r}, \mathbf{r} + \mathbf{u}, l) \right] d^2 \mathbf{r}, \quad (15)$$

where $F_{\Delta}(\mathbf{r}_1, \mathbf{r}_2, l)$ is defined as :

$$F_{\Delta}(\mathbf{r}_1, \mathbf{r}_2, l) = f_{\Delta}(\mathbf{r}_1, \mathbf{r}_2, l) - f_{\Delta}(0, 0, l). \quad (16)$$

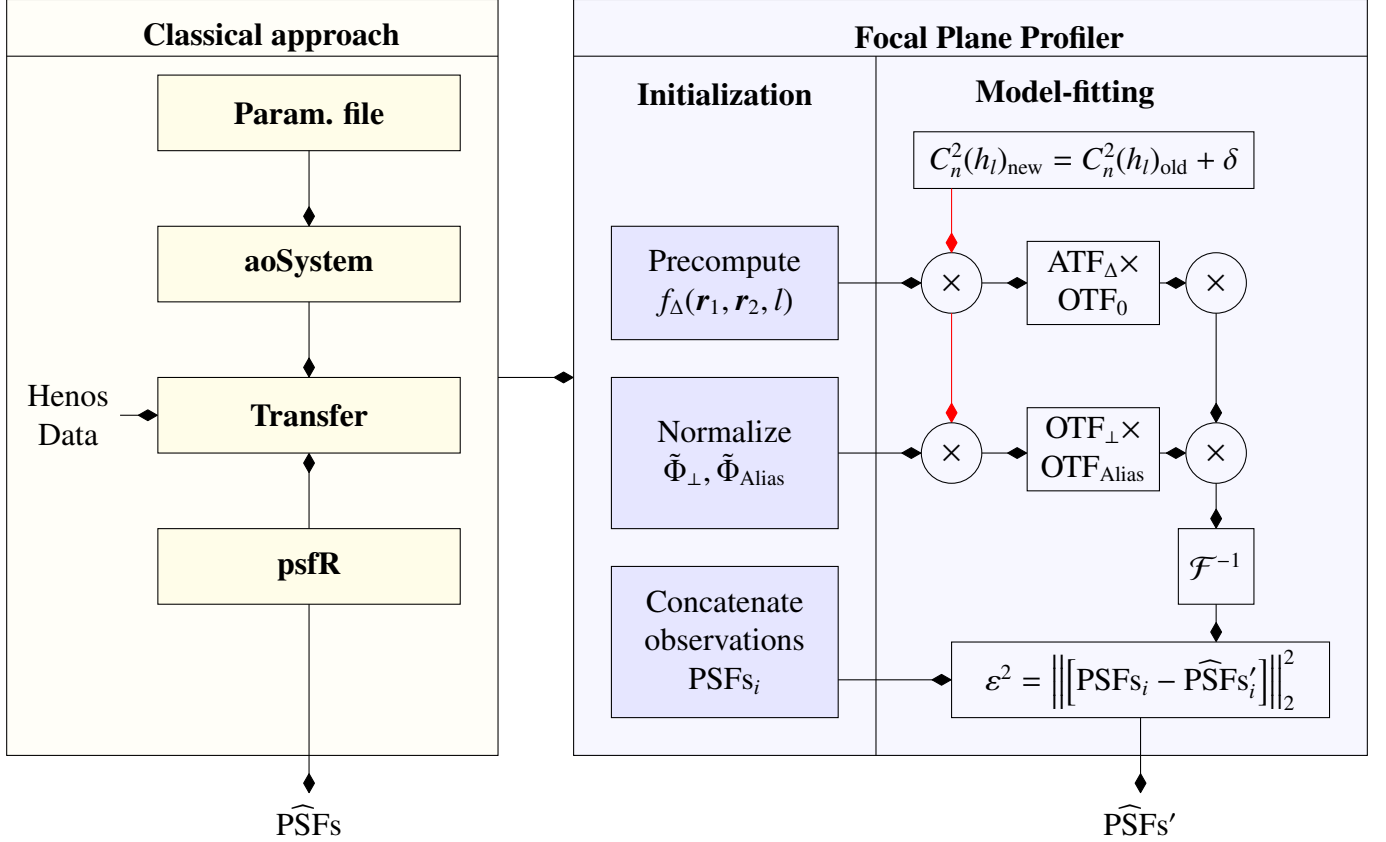
Now the ATF gradient as function of the l^{th} layers strength $C_n^2(h_l)$ takes the following shape :

$$\frac{\partial \text{ATF}_{\Delta}(\mathbf{u}/\lambda, C_n^2(h))}{\partial C_n^2(h_l)} = \text{OTF}_{\text{DL}}^{-1}(\mathbf{u}/\lambda) \iint_{\mathcal{R}^2} \prod_{l=1}^{n_l} 2 \times C_n^2(h_l) \times \mathcal{P}(\mathbf{r}) \mathcal{P}(\mathbf{r} + \mathbf{u}) \times \exp \left[(C_n^2(h_l))^2 \times F_{\Delta}(\mathbf{r}, \mathbf{r} + \mathbf{u}, l) \right] d^2 \mathbf{r}. \quad (17)$$

So far this direct calculation is not yet implemented into the FPP algorithm, but will do in the future. In a first approach, we will speed up a lot the model-fitting procedure, but we would potentially be able to derive the PSF model gradient with respect to the $C_n^2(h)$ profile to perform direct minimization method, without getting through an iterative method.

2.3 Algorithm description

Algorithm description is given in Fig. 1 in a schema-blocks fashion. The right panel explicit the way the information stored into a parameter file is propagated through different Matlab class embedded into the KASP infrastructure to the outputs reconstructed PSF. the FPP algorithm comes on top of that, as a additional layer that produces



3 Results on the NRC-HIA Henos testbed

3.1 System description

All required parameters for PSF-reconstruction are introduced in [1]. Here is an exhaustive list of how the algorithm has been set up :

Note heights values are stretched by a factor eleven for reproducing the same anisoplanatism (in my understanding) on NFIRAOS@TMT considering the given LGS separation.

3.2 How to use the FocalPlaneProfiler class

The FPP algorithm is embedded into the KASP architecture from the focalPlaneProfiler class. This later needs to be passed the psfR class (see KASP notes#03) for initializing the FPP algorithm and perform the model-fitting step. We describe below the piece of code allowing to use the FPP class.

When instantiating, the FPP class can be provided by flag, *fitAlt* to identify altitude layers as well, as so as *fitL0* to include the integrated outer scale (fitL0=1) or outer scale profile (fitL0=2) in the fitting process. If we do so, the computation time drastically increase from a minute up to 5 mn-ish because the pre-computation of the normalized covariance is no longer available since it depends on altitude and outer scale.

Source locations in x	$11 \times [-4.5, -4.5, 4.5, 4.5, -4.5]/2$
Source locations in y	$11 \times [-4.5, 4.5, -4.5, 4.5]/2$
Guide star location	$11 \times [-4.5, -4/5]$
Sources wavelength	670 nm
r_0	0.584 @ 500 nm
L_0	∞
fractional r_0	74.3% ,17.4%,8.2%
altitude layer	(0.6, 5.2, 16.3) km
source height	98.5 km
Telescope diameter	8.13 m
DM actuator pitch	0.813 m
WFS sub-aperture spacing	0.271 m

Table 1: Summary of Henos set-up as inputs of the PSF reconstruction algorithm.

	LGS 1			LGS 2			LGS 3			LGS 4		
	Ref	PSF-R	FPP	Ref	PSF-R	FPP	Ref	PSF-R	FPP	Ref	PSF-R	FPP
SR [%]	43.7	42.6	42.6	4.1	3.8	3.0	6.0	5.2	3.0	5.0	5.3	4.2
FWHM [mas]	21.1	22.6	22.6	116.6	97	112	90	80	93	100	77	91
EE $10\lambda/D$ [%]	78.8	78.8	78.8	64.8	57	55	67	61	58.5	64	59	57
rms [counts]	0	251	250	0	3692	3700	0	1434	1614	0	1597	1481

% Instantiating the aoSystem class

```
sys = aoSystem('HenosSCAO','isSimu',false,'flagGaussian',true);
```

```
sys = instantiateHenosSystemClass(sys,pathData,object,pathCalib);
```

% Instantiating the psfR class and run the reconstruction process

```
psfr = psfR(sys,'flagGaussian',true);
```

```
pr = psfr.getPSF('approach','old','method','slopes-based','samplingDomain','zonal');
```

% Instantiating the focalPlaneProfiler class and run the profile estimation

```
fpp = focalPlaneProfiler(psfr,nLay,3,'fitL0h',0);
```

```
fpp = fpp.getProfile();
```

% Redo the psf reconstruction using retrieved $C_n^2(h)$ profile

```
psfr = psfR(sys,'flagGaussian',true);
```

```
pr = psfr.getPSF('approach','old','method','slopes-based','samplingDomain','zonal');
```

	r_0	$fr_0(1)$	$fr_0(2)$	$fr_0(3)$
Ref	82.97 cm	74.3%	17.4%	8.2%
Fitted	82.88 cm \pm 2.07	68.10% \pm 4.5	23.35% \pm 2.7	8.5% \pm 1.2

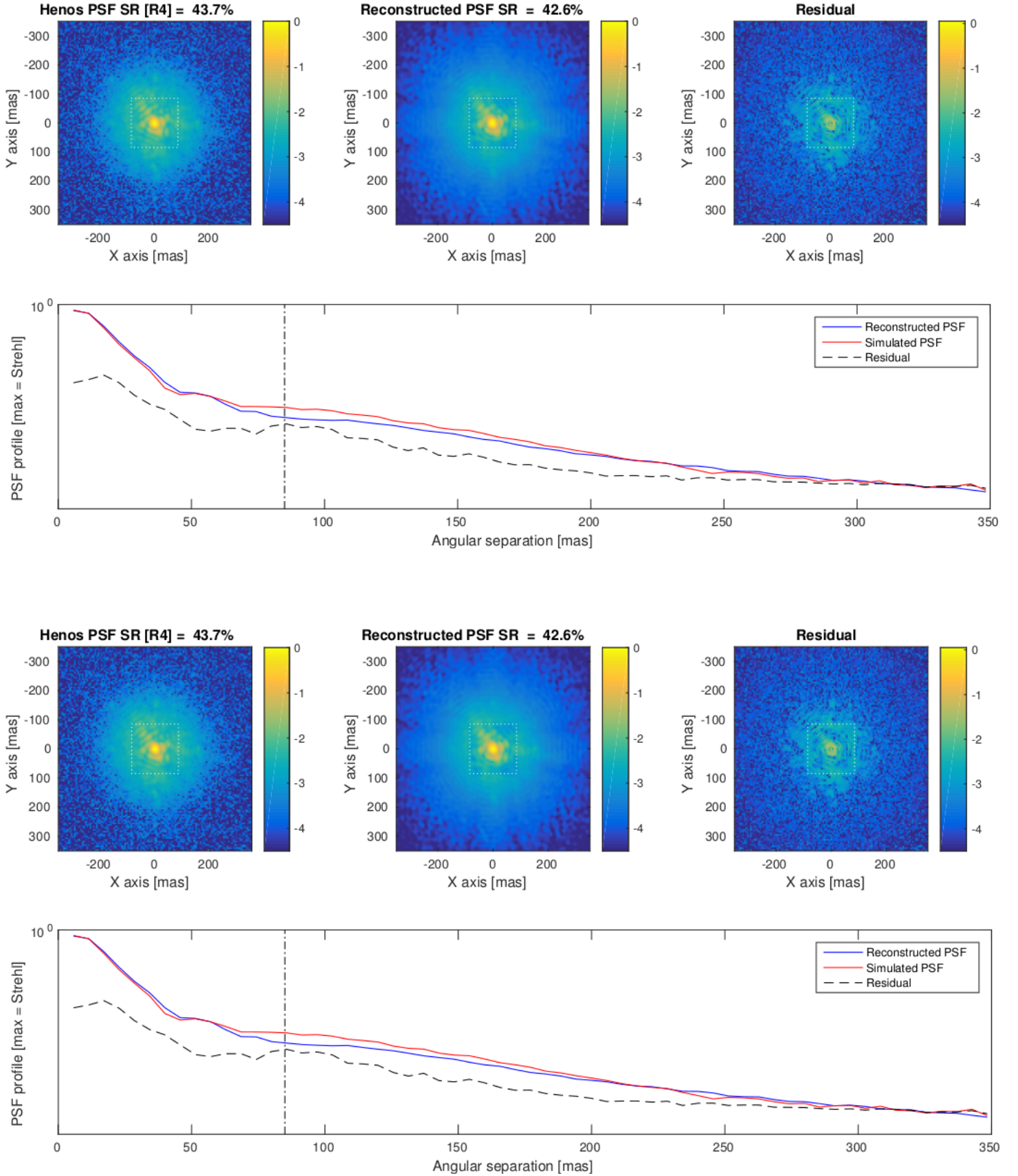


Figure 2: On-axis psf reconstructed before (**up**) and after (**down**) $C_n^2(h)$ calibration using FPP.

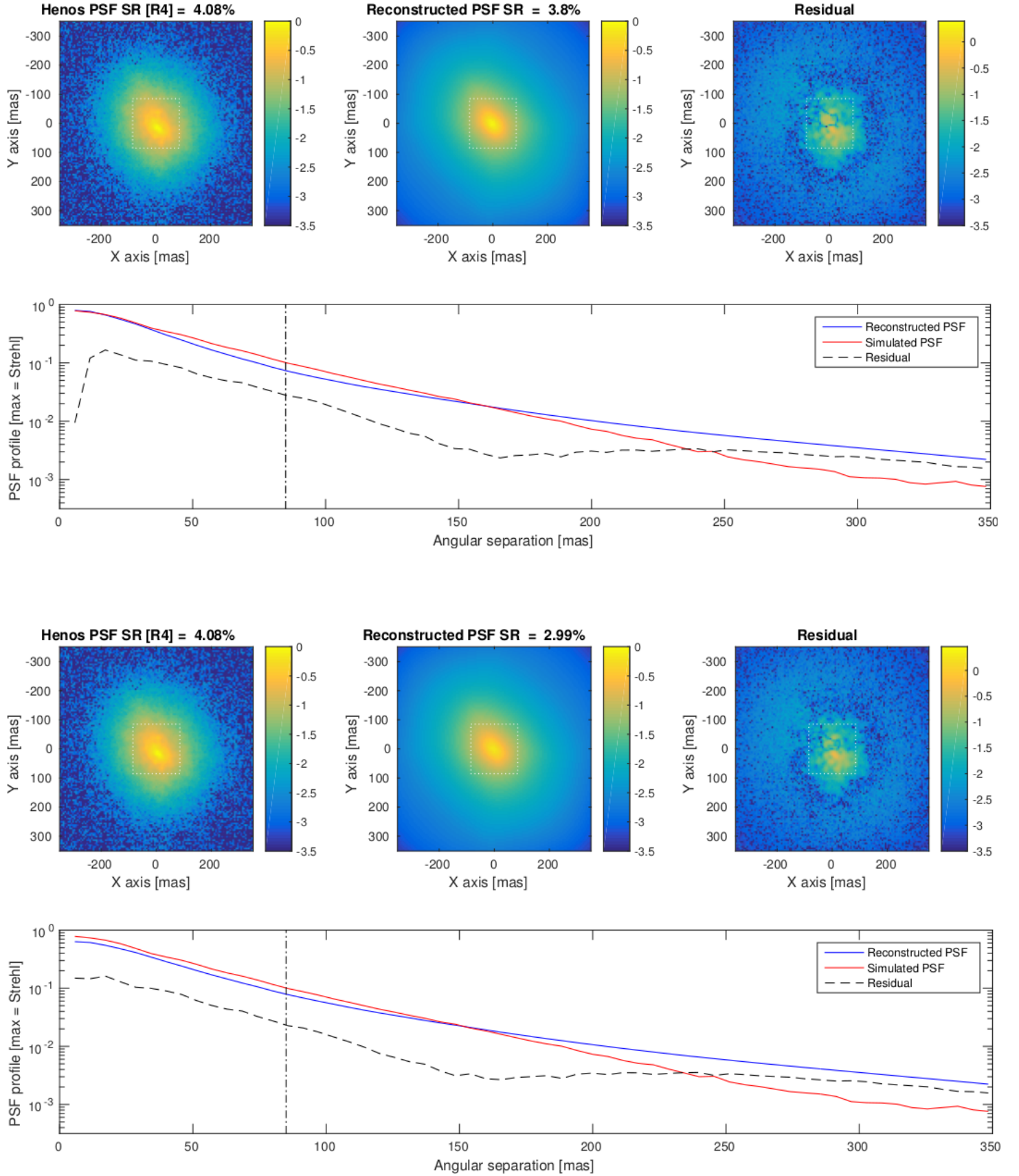


Figure 3: Off-axis LGS 2 psf reconstructed before (**up**) and after (**down**) $C_n^2(h)$ calibration using FPP.

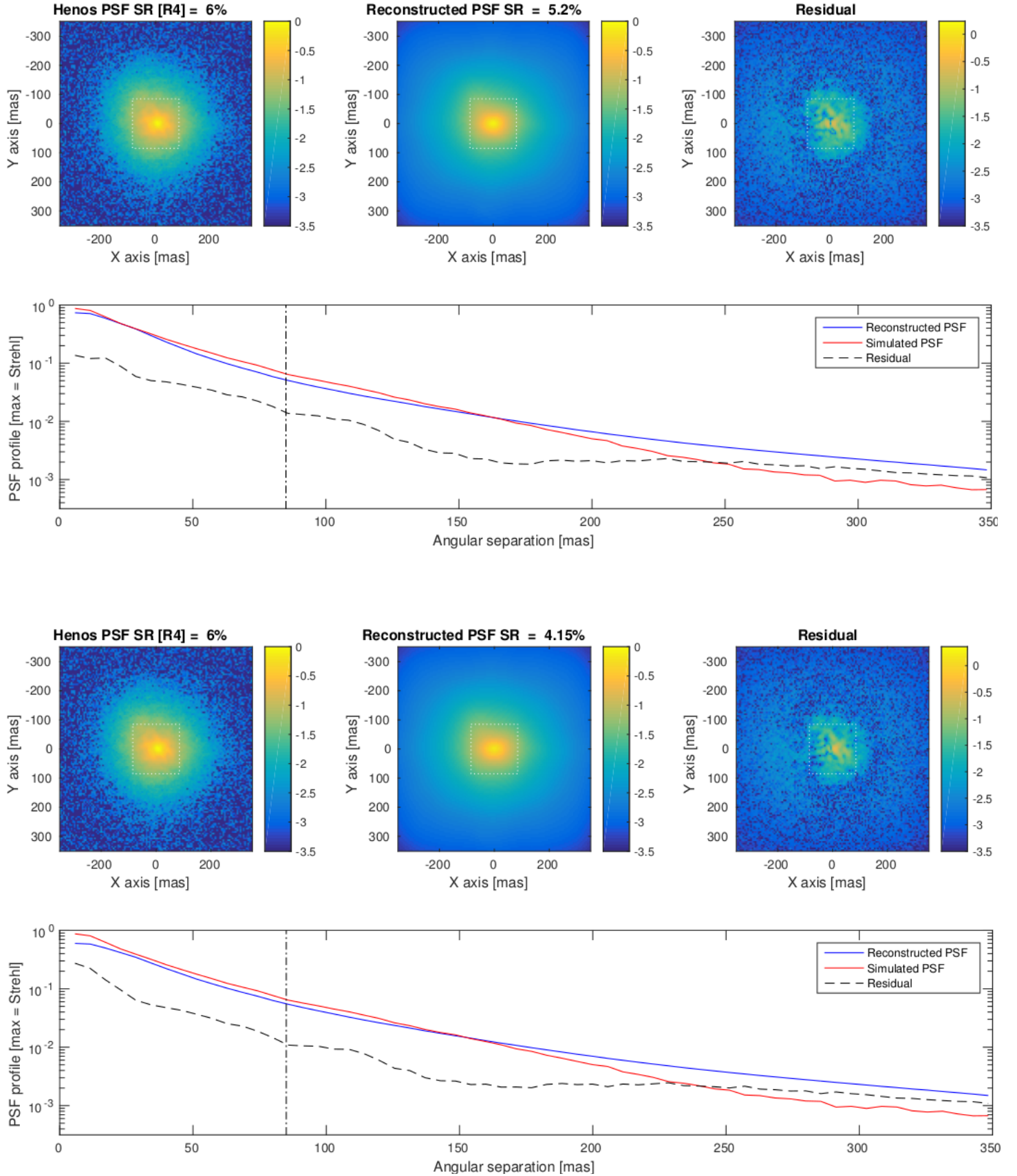


Figure 4: Off-axis LGS 3 psf reconstructed before (**up**) and after (**down**) $C_n^2(h)$ calibration using FPP.

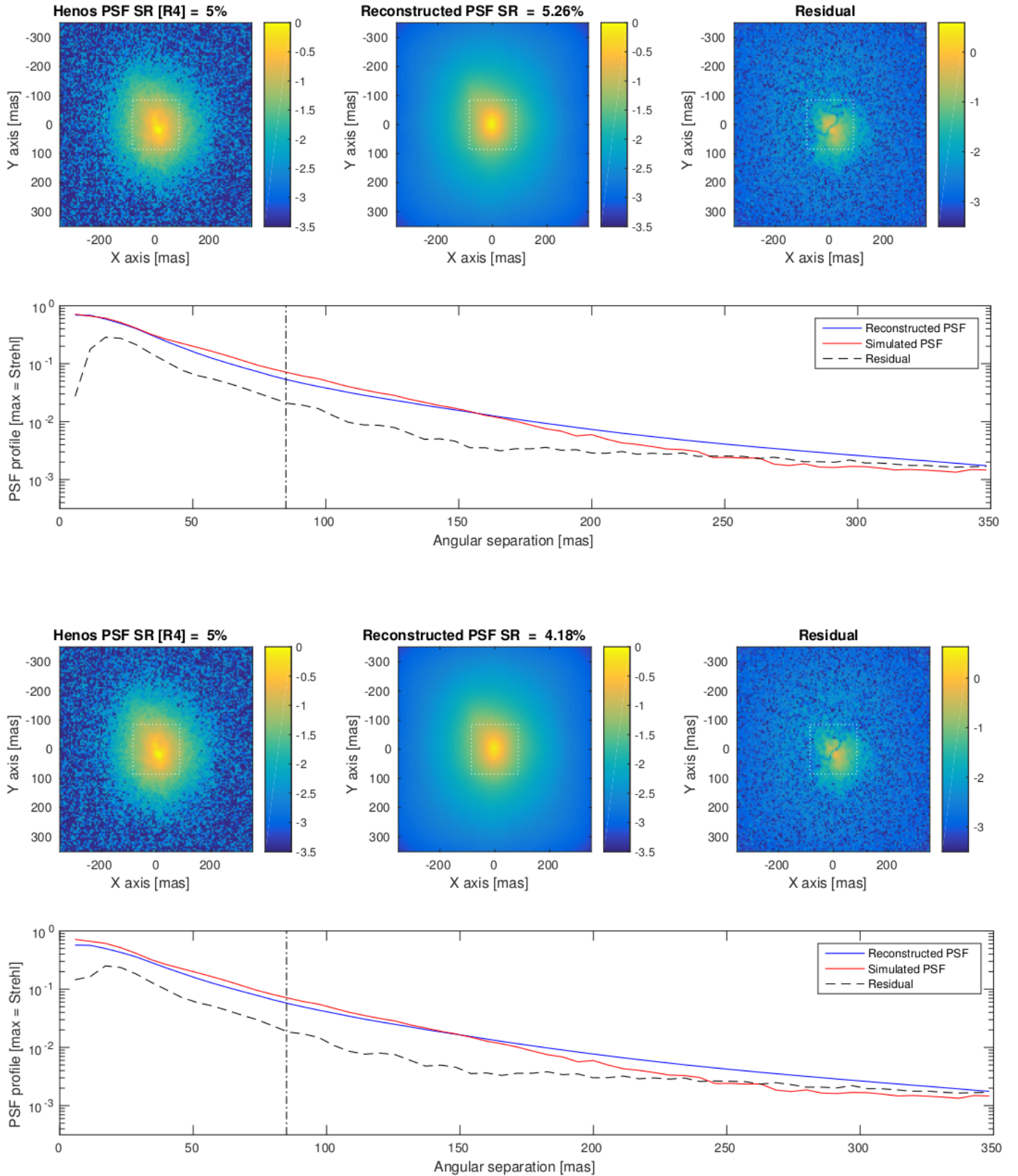


Figure 5: Off-axis LGS 4 psf reconstructed before (**up**) and after (**down**) $C_n^2(h)$ calibration using FPP.

3.3 FPP results

4 Conclusions

References

- [1] M. Rosensteiner, P. Turri, E. Mieda, J.-P. Véran, D. R. Andersen, and G. Herriot. On the verification of NFIRAOS algorithms and performance on the HeNOS bench. In *Adaptive Optics Systems V*, volume 9909 of *Proc. SPIE*, page 990949, July 2016.