Multi-CCD modelling of the point spread function

T. Liaudat, J. Bonnin, J.-L. Starck, M. A. Schmitz, A. Guinot, M. Kilbinger, S. D. J. Gwyn

Astronomy & Astrophysics

Multi-CCD modelling of the point spread function⋆

T. Liaudat[1], J. Bonnin[1], J.-L. Starck[1], M. A. Schmitz[2], A. Guinot[1], M. Kilbinger[1,3], and S. D. J. Gwyn[4]

[1] AIM, CEA, CNRS, Université Paris-Saclay, Université de Paris, 91191 Gif-sur-Yvette, France
    e-mail: tobias.liaudat@cea.fr
[2] Department of Astrophysical Sciences, Princeton University, 4 Ivy Ln., Princeton, NJ 08544, USA
[3] Institut d'Astrophysique de Paris, UMR7095 CNRS, Université Pierre & Marie Curie, 98 bis boulevard Arago, 75014 Paris, France
[4] NRC Herzberg Astronomy and Astrophysics, 5071 West Saanich Road, Victoria, BC V9E 2E7, Canada

ABSTRACT

Context. Galaxy imaging surveys observe a vast number of objects, which are ultimately affected by the instrument's point spread function (PSF). It is weak lensing missions in particular that are aimed at measuring the shape of galaxies and PSF effects represent an significant source of systematic errors that must be handled appropriately. This requires a high level of accuracy at the modelling stage as well as in the estimation of the PSF at galaxy positions.
Aims. The goal of this work is to estimate a PSF at galaxy positions, which is also referred to as a non-parametric PSF estimation and which starts from a set of noisy star image observations distributed over the focal plane. To accomplish this, we need our model to precisely capture the PSF field variations over the field of view and then to recover the PSF at the chosen positions.
Methods. In this paper, we propose a new method, coined Multi-CCD (MCCD) PSF modelling, which simultaneously creates a PSF field model over the entirety of the instrument's focal plane. It allows us to capture global as well as local PSF features through the use of two complementary models that enforce different spatial constraints. Most existing non-parametric models build one model per charge-coupled device, which can lead to difficulties in capturing global ellipticity patterns.
Results. We first tested our method on a realistic simulated dataset, comparing it with two state-of-the-art PSF modelling methods (PSFEx and RCA) and finding that our method outperforms both of them. Then we contrasted our approach with PSFEx based on real data from the Canada-France Imaging Survey, which uses the Canada-France-Hawaii Telescope. We show that our PSF model is less noisy and achieves a ~22% gain on the pixel's root mean square error with respect to PSFEx.
Conclusions. We present and share the code for a new PSF modelling algorithm that models the PSF field on all the focal plane that is mature enough to handle real data.

Key words. methods: data analysis – techniques: image processing – cosmology: observations – gravitational lensing: weak

1. Introduction

Current galaxy imaging surveys, such as DES (Jarvis et al. 2016), KIDS (Kuijken et al. 2015), and CFIS (Ibata et al. 2017), as well as future surveys, such as the *Vera C. Rubin* Observatory's LSST (Tyson et al. 2006), the *Euclid* mission (Laureijs et al. 2011), or the Roman Space Telescope, require an estimated determination of the instrument's point spread function (PSF). For some scientific applications, such as weak gravitational lensing (Kilbinger 2015), low-surface brightness studies (Infante-Sainz et al. 2019), or analyses of diffraction-limited images in crowded stellar fields (Beltramo-Martin et al. 2020), the PSF must be reconstructed with a high level of accuracy. A preliminary approach is to derive a PSF model using available information about the instrument, whereupon the model parameters are then chosen by fitting observed stars in the field to yield a PSF model. This has been widely used for the *Hubble* Space Telescope (HST) as in the case of TinyTim software (Krist et al. 1995), although it was later shown that a relatively simple PSF estimation based on the data, which does not assume a model for the instrument, provides better fits to stars with regard to both photometry and astrometry measurements (Hoffmann & Anderson 2017). Furthermore, such a solution cannot readily be applied to

ground-based observations, where the atmosphere plays an important role and adds a stochastic aspect to the PSF. Other methods, based on imaging-data only, use unresolved stars in the field as direct measurements of the PSF, reconstructing an accurate PSF from these observed stars. A very impressive range of methodologies have been proposed in the past to perform this task: Moffat modelling (Bendinelli et al. 1988), polynomial models (Piotrowski et al. 2013; Bertin 2011), principal component analysis (Jee et al. 2007; Schrabback et al. 2010; Gentile et al. 2013), sparsity (Ngolè et al. 2015), neural networks (Herbel et al. 2018; Jia et al. 2020a,b), and optimal transport (Ngolè & Starck 2017; Schmitz et al. 2018). The PSFEx software (Bertin 2011) is the most widely used. The resolved components analysis (RCA) method (Ngolè et al. 2016; Schmitz et al. 2020) was proposed within the framework of the *Euclid* space mission to deal with PSFs that are both undersampled and spatially varied in the field. Cameras are often mosaics of several charge-coupled devices (CCDs), but all the above-mentioned methods can only build one PSF model per CCD, with the exception of the approach used by Miller et al. (2013), and the recently proposed approach by Jarvis et al. (2021). Since the models they build within each detector are independent from each other, it is difficult to capture global patterns of variation in the PSF. For example, upon observing PSFEx's shape residuals maps from the DES Year 1 results (Fig. 8 in Zuntz et al. 2018), we can see global patterns.

⋆ Code available at https://github.com/CosmoStat/mccd

Table 1. Important variables used in this article.

Variable	Description
Observational model	
$\mathcal{H}$	PSF field
$\mathcal{F}$	Degradation operator
u_i^k	2-dimensional position of star i in CCD k
n_{star}^k	Number of observed stars in CCD k
N	Number of observed stars in all the CCDs
$n_{k,i}$	Noise realisation of star i in CCD k
$y_{k,i}$	Square star observation stamp i on CCD k
n_y	Number of pixels on one dimension of $y_{k,i}$
D	Downsampling factor
$\mathbf{y}_{k,i}$	1-dimensional column representation of $y_{k,i}$
Y_k	Matrix stacking all the star observations $\mathbf{y}_{k,i}$
PSF model	
$\hat{H}_k$	PSF model estimation of the observed stars Y_k
$r_k, \tilde{r}_k$	Local and global number of eigenPSFs
$S_k, \tilde{S}$	Local and global *eigenPSF* matrices
$A_k, \tilde{A}_k$	Local and global weight matrices
$\alpha_k, \tilde{\alpha}$	Local and global spatial constraint weights
$V_k^\top, \Pi_k$	Local and global spatial constraint dictionaries
$K_\sigma^{\text{Loc}}, K_\sigma^{\text{Glob}}$	Local and global denoising parameters
$(e_{k,i}, a_{k,i})$	RCA graph constraint parameters
$\boldsymbol{w}_{k,i}, \tilde{\boldsymbol{w}}_i$	Local and global weight vectors for
	The sparsity inducing term
Φ	Sparsity inducing transform
PSF recovery	
ϕ	Radial Basis Function (RBF) kernel
N_{RBF}	Number of elements used to estimate the
	RBF interpolant
$(\lambda_i)_{i=1}^{N_{\text{RBF}}}$	RBF interpolation weights
$A_k(N_{\text{RBF}})$	Weight matrix composed by the N_{RBF} closest
	Stars of a given target position
$A_{k,u}, \tilde{A}_{k,u}$	Local and global interpolated weight columns
	For a target position u
$\hat{H}(u)$	Recovered PSF at position u

In this paper, we present a new method based on RCA that can capture large patterns spreading across several or all CCDs. We compare the results with both RCA and PSFEx based on simulations and real data. Section 2 reviews these two existing methods, while the proposed MCCD methods are described in Sect. 3. Experiments on simulated images are shown in Sect. 4 and tests on real data are presented in Sect. 5. We give our conclusions in Sect. 7. In addition, Table 1 provides a glossary of variables used throughout this article.

2. PSFEx and RCA

PSFEx (Bertin 2011) is a standard and widely-used software[1]. RCA (Ngolè et al. 2016) is a more recent method that was developed with the *Euclid* Visible Imager's PSF in mind in order to deal with the undersampling of the observed star images. The software is also freely available[2]. It is important to remark that these two approaches rely solely on the observed data: they are blind with respect to the optical system involved in the image acquisition process.

---

[1] https://github.com/astromatic/psfex
[2] https://github.com/CosmoStat/rca

2.1. The observation model

Let us define $\mathcal{H}(u)$ as the PSF field involved in our problem. It is a continuous function of a two-dimensional position, $u = (x, y)$, which, in principle, could be image coordinates based on the camera's CCD pixels or could also be celestial coordinates such as right ascension and declination. Throughout this paper, we assume that this PSF field accounts for the contribution of all effects from optical aberrations and diffraction to atmospheric distortions.

Our observation model consists of images, I_k, the pixels in one CCD chip, k, which contains n_{star}^k noisy stars at positions, u_i^k. We define a 'stamp' as a square small image cutout centred on a single star. Each star observation stamp, i, on the CCD's k can be written as:

$$y_{k,i} = \mathcal{F}\left(\mathcal{H}(u_i^k)\right) + n_{k,i}, \tag{1}$$

where $n_{i,k}$ accounts for a noise image that we will consider to be white and Gaussian, and $\mathcal{F}$ is the degradation operator. Three main effects are taken into account in this operator: (i) the discrete sampling into a finite number of pixels, namely an image stamp of $n_y \times n_y$ pixels; (ii) a sub-pixel shift that depends on where the centroid of the image is placed with respect to the pixel grid; and (iii) a downsampling that affects the pixels in the stamp by a factor of D leaving a $D\,n_y \times D\,n_y$ stamp. For example, to handle the *Euclid* mission sampling rate (Cropper et al. 2013), a factor $D = 1/2$ is required to achieve Nyquist sampling rate. Henceforth, and throughout this article, we use a unitary value for D.

We write each of these stamps into a one-dimensional column vector and, therefore, $Y_k = [\mathbf{y}_{k,1} \cdots \mathbf{y}_{k,n_{\text{star}}^k}]$ is the matrix containing all the observed stamps in CCD's k. It contains n_{star}^k columns and $D\,n_y \times D\,n_y$ rows. Finally, we concatenate all CCD matrices and obtain $Y = (Y_1 \cdots Y_K)$.

2.2. PSFEx

For a given exposure, this method builds one independent model for each CCD. It was designed as a companion software for `SExtractor` (Bertin & Arnouts 1996), which builds catalogues of objects from astronomical images. Each object contains several measurements that `PSFEx` then uses to describe the variability of the PSF. Each selected attribute follows a polynomial law up to some user-defined maximum polynomial degree d. The model for CCD k can be written as:

$$\hat{H}_k^{PSFEx} = S_k A_k, \tag{2}$$

where A_k has m rows corresponding to the number of polynomials used, and n_{star}^k columns corresponding to the number of observed stars used to train the PSF model. The matrix, S_k, is learned during training and has n_y^2 rows (the number of pixels in each image), and m columns.

For example, if d is set to 2 and the attributes chosen are the pixel coordinates (x, y), each column i of the A_k matrix corresponding to the star i at location $u_i^k = (x_i^k, y_i^k)$ is $\boldsymbol{a}_i^k = [1, x_i^k, y_i^k, (x_i^k)^2, (y_i^k)^2, x_i^k y_i^k]^T$. The number of monomials m corresponding to a maximum degree d can be computed as $(d + 1)(d + 2)/2$.

The training of the model amounts to solving an optimisation problem that takes the form:

$$\min_{\Delta S_k} \left\{ \sum_{i=1}^{n_{\text{star}}^k} \left\| \frac{\mathbf{y}_{k,i} - \mathcal{F}\left((S_{0,k} + \Delta S_k)\boldsymbol{a}_i^k\right)}{\hat{\sigma}_i} \right\|_2^2 + \|T \Delta S_k\|_F^2 \right\}, \tag{3}$$

where $\hat{\sigma}_i$ represents the estimated per-pixel variances, and T is a scalar weighting. The matrix S_k is decomposed as $S_{0,k} + \Delta S_k$, where the first term corresponds to a first guess of the PSF. The optimisation is carried out on the difference between this first guess and the observations. The second term in Eq. (3) acts as a Tikhonov regularisation which, in this case, favours smoother PSF models.

Finally, the PSF recovery at one galaxy position, u_j, is straightforward and can be done by using the learned S_k matrix and directly calculating a vector, $\hat{a}_{k,j}$, corresponding to the monomials of the chosen attributes. The recovered PSF is then computed as

$$\hat{h}_{k,j}^{\text{PSFEx}} = S_k \hat{a}_{k,j}. \tag{4}$$

2.3. Resolved components analysis

The RCA method is based on a matrix factorisation scheme. It was first presented in Ngolè et al. (2016) and later evaluated on *Euclid* image simulations in Schmitz et al. (2020). As with PSFEx, this method also builds independent models for each CCD within an exposure and is able to handle under-sampled images. Any observed star i from CCD k is modelled as a linear combination of PSF features, called eigenPSFs in the following, as

$$\hat{h}_{k,i}^{\text{RCA}} = S_k a_{k,i}, \tag{5}$$

where S_k is the matrix composed of the eigenPSFs, $a_{k,i}$ a vector containing the set of linear weights, and $\hat{h}_{k,i}^{\text{RCA}}$ is the reconstructed PSF.

The modelling is recast into an optimisation problem where the S_k and A_k matrices are estimated simultaneously. The problem is ill-posed due to the undersampling and the noise, meaning that many PSF fields can reproduce the observed stars. In order to break this degeneracy, the RCA uses a series of regularisers during the optimisation procedure to enforce certain mild assumptions on the PSF field: (i) the low-rankness of the solution, enforced by setting the number of eigenPSFs learned, N, to be small; (ii) the positivity of the reconstructed PSFs; (iii) the sparsity of the PSF representation on an appropriate basis; and iv) the spatial constraints that account for imposing a certain structure within the A_k matrix. This last constraint is imposed by a further factorisation of A_k into $\alpha_k V_k^T$. The computation of the V_k^T matrix is addressed in Sect. 3.4. Finally, the PSF model reads:

$$\hat{H}_k^{\text{RCA}} = S_k \alpha_k V_k^T, \tag{6}$$

and the optimisation problem that the RCA method solves is:

$$\min_{S_k, \alpha_k} \left\{ \frac{1}{2} \left\| Y_k - \mathcal{F}\left(S_k \alpha_k V_k^T\right) \right\|_F^2 \right.$$
$$\left. + \sum_{i=1}^N \|w_{k,i} \odot \Phi s_{k,i}\|_1 + \iota_+(S_k \alpha_k V_k^\top) + \iota_\Omega(\alpha_k) \right\}, \tag{7}$$

where $w_{k,i}$ are weights, Φ represents a transformation allowing the eigenPSFs to have a sparse representation, $\odot$ denotes the Hadamard product, ι_+ is the indicator function of the positive orthant, and ι_Ω is the indicator function over a set Ω defined to enforce the spatial constraints.

The PSF recovery at a position u_j is carried out by a radial basis function (RBF) interpolation of the learned columns of the A_k matrix, issuing a vector, $a_{k,j}$. In this way, the spatial constraints encoded in the A_k matrix are preserved when estimating the PSF at galaxy positions. Finally, the reconstructed PSF is:

$$\hat{h}_{k,j}^{\text{RCA}} = S_k \hat{a}_{k,j}. \tag{8}$$

3. A new family of MCCD methods

The MCCD methods we propose in this study are aimed at exploiting all the information available in a single exposure, which requires the handling of all CCDs simultaneously. The main advantage of this approach is that we can build a more complex model since the number of stars available for training is much larger as compared to a model based on individual CCDs. We aim to construct a model that is capable of capturing PSF features following a global behaviour, in spite of the fact that the PSF field is discontinuous at CCD boundaries. The main reason behind this discontinuity effect is in the misalignment between various CCDs. Methods such as PSFEx or RCA, which process each CDD independently, avoid the discontinuity problem through construction, but have difficulties capturing global patterns of PSF variability that occur on scales larger than a single CCD. The main idea behind our MCCD approach is to include both a global model that provides a baseline estimation of the PSF and a local model that provides CCD-specific corrections.

3.1. The MCCD data model

In a typical wide-field setting, the PSF field, $\mathcal{H}$, exhibits a certain regularity that we translate into spatial correlations of the PSFs. The model we build for a specific CCD k is the matrix $\hat{H}_k \in \mathbb{R}^{n_y^2 \times n_{\text{star}}^k}$, which is composed by the concatenation of the estimations of the different stars encountered in that CCD. Each postage stamp column of length, n_y^2, corresponds to the model for a specific flattened star from the n_{star}^k stars present in CCD k.

The PSF field at star positions is reconstructed as a linear combination of PSF features, called *eigenPSFs*, which are learned from the observations. As previously stated, we want to have both a global and a local component for the model, so we need different eigenPSFs for each component. Hence, the model is based on a matrix factorisation scheme as follows:

$$\hat{H}_k = \underbrace{S_k A_k}_{\text{Local: } \hat{H}_k^{\text{Loc}}} + \underbrace{\tilde{S} \tilde{A}_k}_{\text{Global: } \hat{H}_k^{\text{Glob}}}, \tag{9}$$

where $S_k \in \mathbb{R}^{n_y^2 \times r_k}$ contains r_k local eigenPSFs and $\tilde{S} \in \mathbb{R}^{n_y^2 \times \tilde{r}}$ contains $\tilde{r}$ global eigenPSFs. The matrices, $A_k \in \mathbb{R}^{r_k \times n_{\text{star}}^k}$ and $\tilde{A}_k \in \mathbb{R}^{\tilde{r} \times n_{\text{star}}^k}$, correspond to the local and global weights of the linear combinations, respectively. We can see that for a given CCD k, the final model, $\hat{H}_k$, is made up of the sum of the contributions of the local, $\hat{H}_k^{\text{Loc}}$, and global, $\hat{H}_k^{\text{Glob}}$, models.

Now, let us build a single model for all the K CCDs in the focal plane. We start by building a single matrix containing all the PSF models by concatenating the model $\hat{H}_k$ for each CCD as follows:

$$\hat{H} = \begin{pmatrix} \hat{H}_1 & \cdots & \hat{H}_K \end{pmatrix}, \tag{10}$$

where $\hat{H} \in \mathbb{R}^{n_y^2 \times N}$ and $N = \sum_{k=0}^K n_{\text{star}}^k$ is the total number of stars in one camera exposure. Then we can concatenate the different eigenPSF matrices, k, into a single matrix:

$$S = \begin{pmatrix} S_1 & \cdots & S_K & \tilde{S} \end{pmatrix}, \tag{11}$$

where $S \in \mathbb{R}^{n_y^2 \times r}$ and we concatenated the global eigenPSF matrix, $\tilde{S}$, at the end. This leaves a total of $r = \sum_{k=1}^{N} r_k + \tilde{r}$ columns for the S matrix. We can follow a similar procedure to define A as a block matrix:

$$A = \begin{pmatrix} A_1 & 0 & \cdots & 0 \\ 0 & A_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_K \\ \tilde{A}_1 & \tilde{A}_2 & \cdots & \tilde{A}_K \end{pmatrix}, \tag{12}$$

where $A \in \mathbb{R}^{r \times N}$ and 0 is used for matrices made up of zeros. The last row of the A block matrix is composed by the global model weights $\tilde{A}_k$. Having already defined the Multi-CCD matrices, $\hat{H}$, S and A, we can write the final model as:

$$\hat{H} = S A, \tag{13}$$

where we include all the CCDs. Expanding it leads to a formula, as shown in Eq. (9), for each CCD.

3.2. Inverse problem and regularisation

The estimation of our model, summarised in the matrices S and A of Eq. (13), is posed as an inverse problem. Given the observation and MCCD data models presented above, this problem amounts to the minimisation of $\|Y - \mathcal{F}(SA)\|_F^2$, where $\|\cdot\|_F$ denotes the Frobenius matrix norm. This problem is ill-posed due to the noise in the observations and to the degradation operator, $\mathcal{F}$, meaning that there are many PSF models that would match the star observations. In order to break this degeneracy, we enforce several constraints, based on basic knowledge of the PSF field, which we use to regularise our inverse problem. Similarly to the ones exploited in the RCA method (Schmitz et al. 2020), we use the following constraints: (i) Low-rankness of the model; (ii) Positivity of the model; (iii) Sparsity in a given domain; (iv) Spatial variations. We give a more detailed description of these in Appendix A. These constraints are used by both parts of our model, namely, the global and the local components.

As mentioned above, the spatial constraint is enforced by further factorisation of the coefficient matrices, A. However, since we want to enforce different properties for the global and the local contributions, the factorisation used differs for each case.

3.3. Global model

We want the global component to provide a baseline estimation of the PSF and for that we propose that the coefficients follow a polynomial variation of the position. The global coefficient matrix, $\tilde{A}_k$, is factorised into $\tilde{A}_k = \tilde{\alpha}\Pi_k$, where $\tilde{\alpha} \in \mathbb{R}^{\tilde{r} \times \tilde{r}}$ is a weight matrix and $\Pi_k \in \mathbb{R}^{\tilde{r} \times n_{\text{star}}^k}$ contains each considered monomials evaluated at global star positions. The dimension, $\tilde{r}$, is determined by the maximum allowed degree in the polynomials: for all monomials of degree less than a given d, we have $\tilde{r} = \binom{d+2}{2} = \frac{(d+1)(d+2)}{2}$. For example, for $d = 2$ (i.e. $\tilde{r} = 6$), we have:

$$\Pi_k = \begin{pmatrix} 1 & \cdots & 1 \\ x_{k,1} & \cdots & x_{k,n_{\text{star}}^k} \\ y_{k,1} & \cdots & y_{k,n_{\text{star}}^k} \\ x_{k,1}^2 & \cdots & x_{k,n_{\text{star}}^k}^2 \\ x_{k,1}y_{k,1} & \cdots & x_{k,n_{\text{star}}^k}y_{k,n_{\text{star}}^k} \\ y_{k,1}^2 & \cdots & y_{k,n_{\text{star}}^k}^2 \end{pmatrix}, \tag{14}$$

where $(x_{k,i}, y_{k,i})_{1 \leq i \leq n_{\text{star}}^k}$ are the global pixel coordinates of the observed stars distributed in the kth CCD. The global component of the model for a specific CCD k are as follows:

$$\hat{H}_k^{\text{Glob}} = \tilde{S} \, \tilde{\alpha} \, \Pi_k. \tag{15}$$

It is important to mention that despite our choice, throughout this paper, to use position polynomials for building the global space constraint, the model is not necessarily restricted to that choice. The Π_k matrix could be constructed using other parameters of the observations in order to facilitate the capture of other dependencies and could also follow other types of functions.

3.4. Local model

It is possible to define different types of local models. In this paper, we discuss three options that depend on how we enforce the local spatial constraint. More specifically, they depend on how we factorise the local A_k matrix in the relation:

$$\hat{H}_k^{\text{Loc}} = S_k A_k. \tag{16}$$

Nevertheless, the MCCD framework does not restrict us to these three options and it is possible to define other local models. It is worth remarking that all the framework and optimisation procedures are maintained throughout the different flavours of the MCCD algorithms. The main difference is the way the spatial constraints are enforced in the local and global models.

3.4.1. MCCD-RCA

One motivation for the local model is to provide CCD-specific corrections and to do so, our first choice is RCA's spatial constraint strategy which leads to the MCCD-RCA algorithm. The motivation for this choice is the capability of the RCA spatial constraint to handle different types of PSF variations. On the one hand, it can capture smooth variations over the CCD and on the other hand, it can account for localised changes that affect a reduced number of PSFs. If the PSFs were sampled on a regular grid, this would mean capturing variations occurring at different spatial frequencies. Unfortunately, the PSF locations do not coincide with a regular grid but on what could be seen as a fully connected undirected weighted graph where the weights can be defined as a function of the distance between the different nodes (PSF locations)[3]. However, the RCA spatial constraint exploits the graph harmonics in order to capture the different PSF variations. These harmonics are represented by the eigenvectors of the graph's Laplacian matrix (Chung 1997), which depend on how we define the graph's weights. A parametric function of the PSF distances can serve as the graph's weights, as in Schmitz et al. (2020), and the selection of the function's parameters can be done following Ngolè et al. (2016). For each local model (i.e. each CCD in the mosaic), we define r_k graphs, each corresponding to one of the r_k local parameters. For each graph, we can extract the m_k most useful eigenvectors of its Laplacian matrix and present all of them as the columns of the matrix $V_k^{\text{RCA}} \in \mathbb{R}^{n_{\text{star}}^k \times r_k m_k}$. In this way, we can write:

$$A_k^{\text{RCA}} = \alpha_k^{\text{RCA}} V_k^{\top \, \text{RCA}}, \tag{17}$$

---

[3] A graph G can be defined as a pair (V, E), where V is the set of vertices and E the set of edges that connects the different vertices. In our case, each star position constitutes a vertex and there is one edge for each pair of vertices. The edges have no preferred direction and its value depends on the distance between the two vertices it connects.

where $\alpha_k^{\mathrm{RCA}} \in \mathbb{R}^{r_k \times r_k m_k}$ is a weight matrix that is used to enforce the spatial constraints. In other words, the sparsity of A_k^{RCA}'s rows in the dictionary $V_k^{\top\ \mathrm{RCA}}$. Full details are available in Ngolè et al. (2016) and Schmitz et al. (2020).

3.4.2. MCCD-POL

The second local model, referred to as MCCD-POL, follows a polynomial spatial constraint. Similar to PSFEx, we factorise the the local weights into two matrices as follows:

$$A_k^{\mathrm{POL}} = \alpha_k^{\mathrm{POL}} \Pi_k^{\mathrm{POL}}, \tag{18}$$

where Π_k^{POL} has the same form as the matrix in Eq. (14), with the difference that in this case, the positions are represented in local coordinates of its corresponding CCD k. As with d in the global model, a parameter is chosen to define the maximum order of the polynomial used.

3.4.3. MCCD-HYB

The third option consists of using the two local models we presented above, namely, the RCA and polynomial, to work together in an hybrid algorithm we will refer to as MCCD-HYB. The idea behind it is that the addition of the polynomial space constraint could help the original graph constraint to capture the different features found. In this case, we factorise the local weights with block matrices as:

$$A_k^{\mathrm{HYB}} = \alpha_k^{\mathrm{HYB}} V_k^{\top\ \mathrm{HYB}} = \begin{pmatrix} \alpha_k^{\mathrm{RCA}} & 0 \\ 0 & \alpha_k^{\mathrm{POL}} \end{pmatrix} \begin{pmatrix} V_k^{\top} \\ \Pi_k^{\mathrm{POL}} \end{pmatrix}, \tag{19}$$

where α_k^{POL} and Π_k^{POL} are the matrices defined in the polynomial version and α_k^{RCA} and $V_k^{\top}$ are the matrices defined in the original MCCD-RCA algorithm.

Finally, generically including the spatial constraints in Eq. (9), we get the following description of our model for a specific CCD:

$$\hat{H}_k = S_k \alpha_k V_k^{\top} + \tilde{S} \tilde{\alpha} \Pi_k, \tag{20}$$

which we can also write in a global form, $\hat{H} = S \alpha V^{\top}$, where $\hat{H}$ and S have already been defined in Eqs. (10) and (11), and where α and $V^{\top}$ are the following matrices:

$$\alpha = \begin{pmatrix} \alpha_1 & 0 & \cdots & 0 \\ 0 & \ddots & & \vdots \\ \vdots & & \alpha_N & 0 \\ 0 & \cdots & 0 & \tilde{\alpha} \end{pmatrix}, \quad V^{\top} = \begin{pmatrix} V_1^{\top} & 0 & \cdots & 0 \\ 0 & V_2^{\top} & & \vdots \\ \vdots & & \ddots & 0 \\ 0 & \cdots & 0 & V_N^{\top} \\ \Pi_1 & \Pi_2 & \cdots & \Pi_N \end{pmatrix}. \tag{21}$$

3.5. Optimisation problem

Combining the regularisations enumerated in Sect. 3.2 and the data model described in Sect. 3.1, we can construct the optimisation problem in an elegant way by reformulating Eq. (7). However, we can split the optimisation problem into a more convenient form:

$$\min_{\substack{S_1,\ldots,S_N,\tilde{S} \\ \alpha_1,\ldots,\alpha_N,\tilde{\alpha}}} \Bigg\{ \sum_{k=1}^{N} \bigg(\frac{1}{2} \|Y_k - \mathcal{F}_k(S_k \alpha_k V_k^{\top} + \tilde{S} \tilde{\alpha} \Pi_k)\|_F^2 +$$
$$\sum_{i=1}^{r_k} \|\boldsymbol{w}_{k,i} \odot \Phi \boldsymbol{s}_{k,i}\|_1 + \iota_+(S_k \alpha_k V_k^{\top} + \tilde{S} \tilde{\alpha} \Pi_k) + \iota_{\Omega_k}(\alpha_k) \bigg)$$
$$+ \sum_{i=1}^{\tilde{r}} \|\tilde{\boldsymbol{w}}_i \odot \Phi \tilde{\boldsymbol{s}}_i\|_1 + \iota_{\tilde{\Omega}}(\tilde{\alpha}) \Bigg\}. \tag{22}$$

In the previous equation, the columns of $Y_k \in \mathbb{R}^{D^2 n_y^2 \times n_{\mathrm{star}}^k}$ are the stars distributed in the kth CCD sensor, $\mathcal{F}_k$ is the degradation operator, $\boldsymbol{w}_{k,i}$ and $\tilde{\boldsymbol{w}}_i$ are weight vectors, Φ is a transform that allows a sparse representation of our eigenPSFs, and Ω_k and $\tilde{\Omega}$ are sets to enforce sparsity and normalisation of the rows of α_k and $\tilde{\alpha}$, respectively. The indicator function of a set C is written as $\iota_C(\cdot)$, that is equal to 0 if the argument belongs to C and $+\infty$ otherwise. For example, ι_+ is the indicator function over the positive orthant. More explicitly, the sets Ω_k and $\tilde{\Omega}$ are defined the following way:

$$\Omega_k = \{\alpha_k \mid \forall i \in \{1,\ldots,r_k\}, \|(\alpha_k^{\top})_i\|_0 \le \eta_{k,i} \wedge \|(\alpha_k V_k^{\top})_i\|_2 = 1\}, \tag{23}$$

$$\tilde{\Omega} = \{\tilde{\alpha} \mid \forall i \in \{1,\ldots,\tilde{r}\}, \|(\tilde{\alpha}^{\top})_i\|_0 \le \tilde{\eta}_i \wedge \|(\tilde{\alpha} \Pi_k)_i\|_2 = 1\}, \tag{24}$$

where $(\eta_{k,i})_{1 \le i \le r_k}$ and $(\tilde{\eta}_i)_{1 \le i \le \tilde{r}}$ are appropriately chosen integers, and $\|\cdot\|_0$ is the pseudo-norm ℓ_0 that returns the number of non-zero elements of a vector. So we are enforcing, in the global case, the row $i \in \{1,\ldots,\tilde{r}\}$ of $\tilde{\alpha}$ to have at most $\tilde{\eta}_i$ non-zero elements. An interpretation could be that we are forcing each eigenPSF to follow a small number of positional polynomials as $\tilde{A}$'s rows will be sparsely represented over the Π_k matrices.

The Φ transform used throughout this paper is the starlet transform (Starck et al. 2011). We enforce the sparsity on the different decomposition levels excluding the coarse scale. The ℓ_1 term promotes the sparsity of the eigenPSFs with respect to Φ while the weights $\boldsymbol{w}_{k,i}$ and $\tilde{\boldsymbol{w}}_i$ regulate the sparsity penalisation against the other constraints and should be adapted throughout the optimisation algorithm depending on the noise level.

The second term in each of the $\tilde{\Omega}$ sets (e.g. $\|(\alpha_k V_k^{\top})_i\|_2 = 1$) was not mentioned in the regularisation Sect. 3.2, but they are needed to avoid a degenerated solution, for example $\|S_k\|_F \to \infty$ and $\|A_k\|_F \to 0$, due to the usual scale indeterminacy when doing a matrix factorisation. To avoid this, we normalise the A_k and $\tilde{A}$ columns. This translates to forcing the normalisation of the eigenPSF weights contributing to model each observed star. This does not mean that the eigenPSF weights will be the same for each star, but that the norms of the weight vectors are equal.

3.6. Algorithm

The optimisation in Eq. (22) is non-convex as we are facing a matrix factorisation problem. To overcome this situation we use an alternating minimisation scheme where we optimise one variable at a time, iterating over the variables as studied in Xu & Yin (2013) or Bolte et al. (2014). In consequence, we can at most expect them to converge towards a local minima. The main iteration is performed over the different variables occurring in Eq. (22), first over the global $\tilde{S}, \tilde{\alpha}$ and then over the local $S_1, \alpha_1, \ldots, S_K, \alpha_K$.

The method is shown in Algorithm 1, which contains the four main optimisation problems derived from the alternating

scheme. There exists a wide literature on minimisation schemes involving non-smooth terms, specifically proximal methods (Parikh & Boyd 2014), which we can exploit in order to handle the four cases. Notably, we use the algorithm proposed by Condat (2013) for problems (II), (III), and (IV). For problem (I), we use the method proposed by Liang et al. (2018) which is an extension of the well-known FISTA algorithm (Beck & Teboulle 2009). Even though the ℓ_0 pseudo-norm is non-convex and, therefore, not adapted to the general scenario of the aforementioned algorithms, we can alleviate this issue by combining the application of its proximal operator and a given heuristic.

With regard to the algorithm's initialisation, we start by a preprocessing where we reject stars that are strong outliers in terms of shape or size. We run the shape measurement algorithm mentioned in Sect. 4.4 on the training stars and discard those that are several sigmas away from nearby stars. At this moment, we can assign a specific weight for each training star. There are three available options: (i) to use a unitary weight for each training star; (ii) to use a weight provided by the user; (iii) to compute a weight ω_i as a function of the star's signal-to-noise-ratio (S/N) based on $\omega_i \propto S/N_i/(S/N_i+\text{median}(S/N))$ and bound to a specific interval to avoid bright stars from dominating the optimisation.

Next, we continue with all the local eigenPSFs set to zero, as seen in line 4 of Algorithm 1; and the $\tilde{\alpha}$ matrix set to the identity, favouring the specialisation of each global eigenPSF to one specific monomial. By following this procedure, we are training a global polynomial model that fits the stars as best as it can. Later on, the local models work with the residuals between the observed stars and the global model, trying to capture variations missed in the previous step.

There are four iteration loops in Algorithm 1. In line 8, this is the main iteration, and in line 15, the iteration over the CCDs for the training of the local model. The other two iterations on lines 9 and 14 correspond to a refinement of the estimation. Our objective is to correctly estimate the global and the local contributions for the model and to do this, we alternate the minimisation between the global and the local contributions, which we call outer minimisation. On top of that, each of these two contributions include an inner alternating minimisation scheme as we are performing a matrix factorisation for the local and for the global models. For example, we are simultaneously minimising over S_k, α_k for the local model and over $\tilde{S}, \tilde{\alpha}$ for the global model. We want to refine this inner minimisation, meaning that the optimisation of the two variables separately approaches the joint optimisation of both variables. To accomplish this, we need to go through a small number of iterations, which are described by the n superscript variables, before continuing the iteration of the next alternating scheme. The optimisation strategy can be seen as a compound alternating minimisation scheme considering the outer and the inner alternations. More information about the optimisation strategy can be found in Appendix D.

3.7. PSF recovery

Once the training of the model on the observed stars is complete, we can continue with the problem of estimating the PSF field at galaxy positions. We call this problem PSF recovery. Gentile et al. (2013) conducted a study on PSF interpolation techniques and Ngolè & Starck (2017) proposed a sophisticated approach based on optimal transport theory (Peyré & Cuturi 2018). We will follow a RBF (Radial Basis Function) interpolation scheme with a thin plate kernel[4], as in Schmitz et al. (2020),

---

[4] Where the kernel is defined as $\phi(r) = r^2 \ln(r)$.
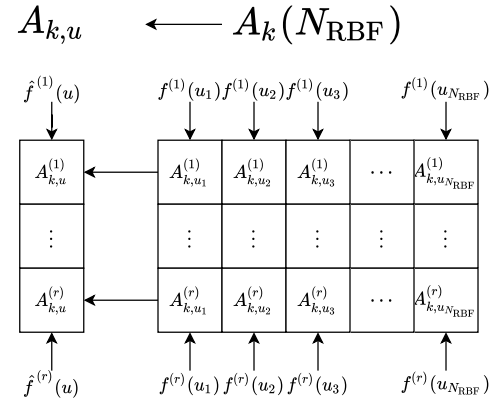
Fig. 1. Example of the interpolation procedure involved in the PSF recovery.

due to its simplicity and good performance. This choice comes with the assumption that the influence of each observation does not depend on the direction but only on the distance to the target which is well described by the RBF kernel.

The RBF interpolation of a function f on a position u works by building a weighted linear combination of RBF kernels ($\phi(\cdot)$) centred in each of the available training star positions u_i. The interpolation function reads

$$\hat{f}(u) = \sum_{i=1}^{N_{\text{RBF}}} \lambda_i \phi\left(\|u - u_i\|\right), \tag{25}$$

where $(\lambda_i)_{i=1}^{N_{\text{RBF}}}$ are the linear weights that need to be learnt and N_{RBF} is the number of elements used to estimate the interpolant. In order to learn the weights, we force the exact reconstruction of the interpolant on the known positions, that is $\hat{f}(u_i) = f(u_i) \, \forall i \in \{1, \ldots, N_{\text{RBF}}\}$. By fixing the aforementioned constraint we have a system of N_{RBF} equations with N_{RBF} unknown that are the λ_i weights. Once the system is solved, it is just a matter of evaluating the interpolant on the desired position u following Eq. (25).

At this point, we need to choose the function f over which we go on to interpolate. A straightforward choice would be to use the reconstructed PSFs at the training positions as the $f(u_i)$. Nevertheless, this would not take into account the specificities and structure of our model. Following the discussion in Sect. 4.2 of Schmitz et al. (2020), we use the learnt A_k and $\tilde{A}_k$ matrices. They encompass all the spatial distribution properties of the learned features, that is, our eigenPSFs; thus it is natural for our framework to use these values as the function to interpolate.

We continue with a brief explanation of the interpolation procedure. For one given target position u in CCD k, we consider the N_{RBF} closest observed stars to that position that also belong to the CCD k. We call $A_k(N_{\text{RBF}})$ to the A_k matrix composed only with the columns of the aforementioned N_{RBF} stars. We want to estimate the interpolated column vector $A_{k,u}$. For this, we use a RBF interpolation scheme for each row of the $A_k(N_{\text{RBF}})$ matrix. The elements of the row t represent the $(f^{(t)}(u_i))_{i=1}^{N_{\text{RBF}}}$ evaluations and the element $A_{k,u}^{(t)}$ represents the interpolated value $\hat{f}^{(t)}(u)$. The same procedure is repeated for each row of the $A_k(N_{\text{RBF}})$ matrix so as to obtain the column vector $A_{k,u}$. This is illustrated in Fig. 1. We repeat the procedure with the global component matrix, $\tilde{A}_k$, in order to obtain $\tilde{A}_{k,u}$, another column vector with the interpolated values. At this point, we note that we handle the global and the local contributions independently. Once we have calculated the two interpolated vectors, the reconstructed PSF is obtained

---

Algorithm 1 Multi-CCD Resolved Components Analysis

Initialisation:
1: Preprocessing()
2: **for** $k = 1$ to K **do**
3: $\quad$ Harmonic constraint parameters $(e_{k,i}, a_{k,i})_{1 \le i \le r_k} \rightarrow V_k^\top, \alpha_k^{(0,0)}$
4: $\quad 0_{n_y^2 \times r_k} \rightarrow S_k^{(0,0)}$
5: **end for**
6: Global coordinates $\rightarrow \Pi_k, \tilde{\alpha}^{(0,0)}$ $\hfill (\tilde{\alpha}^{(0,0)} = I)$
7: $0_{n_y^2 \times \tilde{r}} \rightarrow \tilde{S}^{(0,0)}$

Alternate minimisation:
8: **for** $l = 0$ to l_{max} **do** $\hfill$ Algorithm's main iterations

9: $\quad$ **for** $n = 0$ to n_G **do** $\hfill$ Global alternating iterations
10: $\quad\quad$ Noise level, $\tilde{\alpha}^{(l,n)} \rightarrow$ update $\tilde{W}^{(l,n)}$
11: $\quad\quad \tilde{S}^{(l+1,n+1)} = \arg\min_{\tilde{S}} \{\sum_{k=1}^{K} \frac{1}{2}\|Y_k - \mathcal{F}_k(S_k^{(l,0)}\alpha_k^{(l,0)}V_k^\top + \tilde{S}\tilde{\alpha}^{(l,n)}\Pi_k)\|_F^2 + \sum_i \|\tilde{\boldsymbol{w}}_i^{(l,n)} \odot \Phi\tilde{\mathbf{s}}_i\|_1\}$ $\hfill$ (I)
12: $\quad\quad \tilde{\alpha}^{(l+1,n+1)} = \arg\min_{\tilde{\alpha}} \{\sum_{k=1}^{K} \frac{1}{2}\|Y_k - \mathcal{F}_k(S_k^{(l,0)}\alpha_k^{(l,0)}V_k^\top + \tilde{S}^{(l+1,n+1)}\tilde{\alpha}\Pi_k)\|_F^2 + \iota_{\tilde{\Omega}}(\tilde{\alpha})\}$ $\hfill$ (II)
13: $\quad$ **end for**

14: $\quad$ **for** $n = 0$ to n_L **do** $\hfill$ Local alternating iterations
15: $\quad\quad$ **for** $k = 1$ to K **do** $\hfill$ CCD iterations
16: $\quad\quad\quad$ Noise level, $\alpha_k^{(l,n)} \rightarrow$ update $W_k^{(l,n)}$
17: $\quad\quad\quad S_k^{(l+1,n+1)} = \arg\min_{S_k} \{\frac{1}{2}\|Y_k - \mathcal{F}_k(S_k\alpha_k^{(l,n)}V_k^\top + \tilde{S}^{(l+1,n_G)}\tilde{\alpha}^{(l+1,n_G)}\Pi_k)\|_F^2$
$\quad\quad\quad\quad + \sum_i \|\boldsymbol{w}_{k,i}^{(l,n)} \odot \Phi\mathbf{s}_{k,i}\|_1 + \iota_+(S_k\alpha_k^{(l,n)}V_k^\top + \tilde{S}^{(l+1,n_G)}\tilde{\alpha}^{(l+1,n_G)}\Pi_k)\}$ $\hfill$ (III)
18: $\quad\quad\quad \alpha_k^{(l+1,n+1)} = \arg\min_{\alpha_k} \{\frac{1}{2}\|Y_k - \mathcal{F}_k(S_k^{(l+1,n+1)}\alpha_k V_k^\top + \tilde{S}^{(l+1,n_G)}\tilde{\alpha}^{(l+1,n_G)}\Pi_k)\|_F^2 + \iota_{\Omega_k}(\alpha_k)\}$ $\hfill$ (IV)
19: $\quad\quad$ **end for**
20: $\quad$ **end for**

21: **end for**

following the MCCD data model as can be seen in the next equation

$$\hat{H}(u) = \tilde{S}\tilde{A}_{k,u} + S_k A_{k,u}. \tag{26}$$

We found that restricting the N_{RBF} neighbours to a single CCD for the global components gives better results. This might be due to the fact that the global components are able to capture some of the discontinuities from one CCD to another and, therefore, the interpolation is degraded when using stars from different CCDs. The number of neighbours N_{RBF} should be chosen as a function of the available number of stars per CCD in the training set and as the RBF kernel chosen. Henceforth, and given the training set we handle in this study, N_{RBF} is set to 20.

4. Numerical experiments

4.1. Data

The simulated data set we create to evaluate MCCD set is based on a Canada-France Imaging Survey (CFIS)[5] MegaCam[6] exposure from the Canada-France-Hawaii Telescope (CFHT). It contains 2401 stars distributed along 40 CCDs over a field of view of ~1 deg^2 as shown in Fig. 2. Each CCD consists of a matrix of 2048 by 4612 pixels with some given gaps between the different CCDs. The horizontal gap length consist of ~70 pixels while vertical gaps contain ~425 pixels.
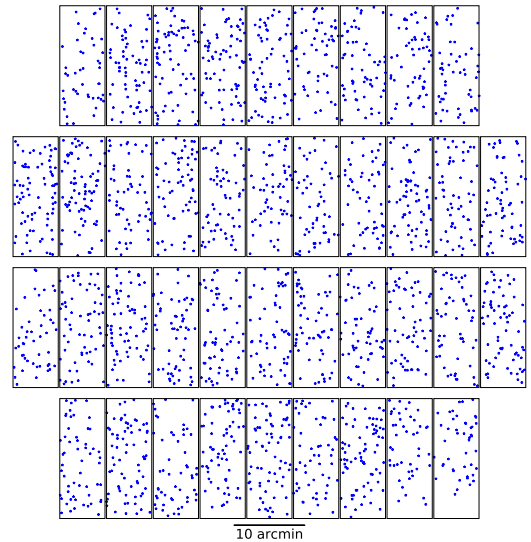
Fig. 2. Star positions in CFHT's MegaCam used for the simulated dataset. The positions were taken from a real CFIS exposure.
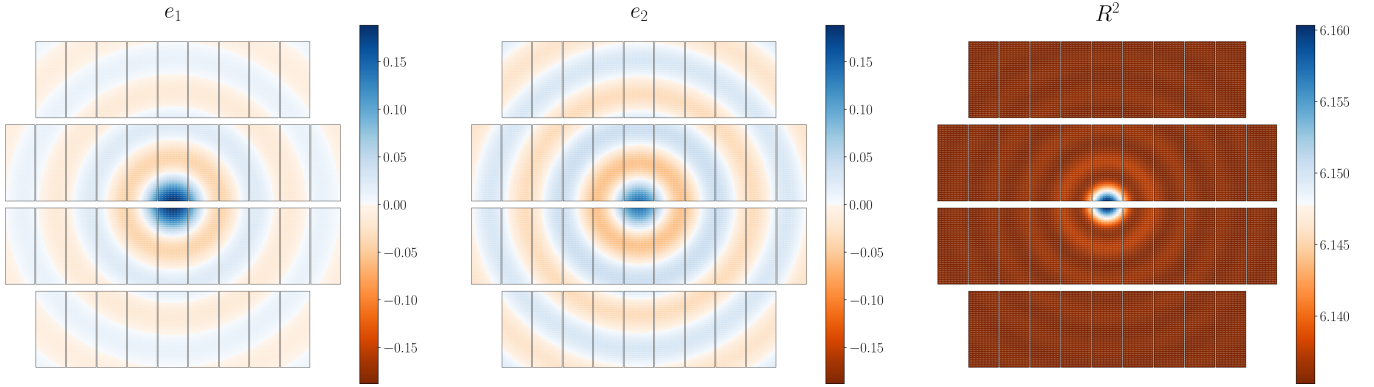
4.2. Training set

Our simulation pipeline considers a Moffat PSF profile with normalised flux drawn using the Galsim software[7] (Rowe et al. 2015) for each position in the exposure. To simulate the PSF shape variation, we used two radial analytic functions which

---

Fig. 3. Shape measurement results of the simulated test star catalogue following the analytical ellipticities.

define our ground truth shape ellipticities distortions. Shearing stars leads naturally to a size variation. Figure 3 shows the resulting e_1, e_2 and size maps. Our pipeline performs the following steps:

1. Simulate Moffat stars with a size fixed to the mean size measured in the real exposure.
2. Shear the simulated stars as a function of their position using the two analytical functions.
3. Apply a random sub-pixel shift following a uniform distribution centred on zero.
4. Apply a binning to get a 51×51 pixel image, with a pixel size equivalent to CFIS MegaCam's maps, that is, 0.187 arcsec.
5. Add a constant white Gaussian noise to the images, with standard deviation σ, derived from the desired S/N level

$$\text{S/N} = \frac{\|y\|_2^2}{\sigma^2 p^2}, \tag{27}$$

where y is the image postage stamp consisting of p^2 pixels. Each experience will consist of a constant S/N value, as we later see, which is drawn from the set {10, 30, 50, 70}.

Since `PSFEx` was designed as a companion software to `SExtractor`, we need to follow a different procedure to generate the simulated data. We first need to process our simulations with `SExtractor`, so that the catalogue produced can be used as inputs for `PSFEx`. To accomplish this, we mimic a complete CCD so that `SExtractor` is able to process it. We create star images as we already described for the MCCD method but without noise as it will be added later. Then we distribute them on a mock image of 2048 × 4612 pixels. The corresponding positions will be the pixel coordinates that are presented in Fig. 2. Once the mock image is created, we add the noise value according to the desired S/N to the whole image. When the mock image is created, we run `SExtractor` in order to have a star catalogue that `PSFEx` can use as input.

4.3. Testing data set

For the testing, we want to observe how well the different models capture the ellipticity maps when trained on real star positions. Therefore, the positions in each CCD are taken from a regular grid of 20 × 40 and considering that the total amount of CCDs is 40, we finally obtain a total of 32 000 stars with which to test our model. These stars are simulated following the same ellipticity maps (see Fig. 3), without any sub-pixel shift and without any noise. The goal is now to use the training data (i.e. simulated observed stars) to learn the model, and then to predict the PSFs

at positions of test stars. As we have the ground truth at these positions, without noise and sub-pixel shift, it is easy to get a robust evaluation of model predictions.

4.4. Quality criteria

In order to correctly assess the performance of our PSF modelling algorithm, we consider several criteria:

– Pixel root mean square error (RMSE): calculated between the pixel images of the recovered PSFs and the noiseless test stars. The expression of the pixel RMSE is the following:

$$Y_{\text{pix}}^{\text{RMS}} = \sqrt{\langle (Y^* - \hat{Y})^2 \rangle} = \left(\frac{1}{Nn_y^2} \sum_{i=1}^{N} \sum_{j=1}^{n_y^2} (Y_{i,j} - \hat{Y}_{i,j})^2 \right)^{\frac{1}{2}}, \tag{28}$$

where $Y_{i,j}$ is the pixel j of test star i that has a total of n_y^2 pixels, N is the total number of test stars, $\hat{Y}_{i,j}$ is the estimation of the test star's pixel and $\langle \cdot \rangle$ denotes the mean over all the elements in the array.

– Shape (ellipticity) error: We estimate the ellipticities of reconstructed stars using the adaptive moments' ellipticity estimator from `Galsim`'s HSM module (Hirata & Seljak 2003; Mandelbaum et al. 2005). The shape and size definitions can be found in Appendix B. For each of the ellipticity components, the RMSE is calculated as:

$$e^{\text{RMS}} = \sqrt{\langle (e^* - \hat{e})^2 \rangle} = \left(\frac{1}{N} \sum_{i=1}^{N} (e_i - \hat{e}_i)^2 \right)^{\frac{1}{2}}. \tag{29}$$

– Size error: We use the measurements from HSM and the definition in Appendix B to compute the following RMSE:

$$R^{2,\text{RMS}} = \sqrt{\langle (R^{2,*} - \hat{R}^2)^2 \rangle} = \left(\frac{1}{N} \sum_{i=1}^{N} (R_i^2 - \hat{R}_i^2)^2 \right)^{\frac{1}{2}}. \tag{30}$$

– Moment residual maps: To visualise the shape and size errors, we plot these quantities as a function of their position on the focal plane.

When comparing two methods, we define the relative gain with regard to the metric, m, of method 1 with respect to the method 2 as:

$$G_{1/2}(m) = \frac{m_2 - m_1}{m_2} \times 100\%. \tag{31}$$

4.5. Model parameters

Based on experiments with simulated and real data, we have chosen the following parameters

- PSFEx: we use the following configuration:

  PSF_SAMPLING   1.0
  PSF_SIZE   51,51
  PSFVAR_KEYS   XWIN_IMAGE,YWIN_IMAGE
  PSFVAR_GROUPS   1,1
  PSFVAR_DEGREES 2

  PSFVAR_DEGREES refers to the maximum polynomial degree, and, XWIN_IMAGE and YWIN_IMAGE, to the windowed centroid positions in pixel coordinates. The PSFEx software[8] does not include publicly an interpolation method, so we use an available PSFEx interpolation module[9].
- RCA: we set r equal to eight local components, the denoising parameters K_σ^{RCA} to 1, and the other parameters to their default value from its official repository[10].
- MCCD: we use the same parameters as RCA for the local component and a maximum polynomial degree of eight for the global components. The denoising parameters K_σ^{Loc} and K_σ^{Glob} are set to 1 for the local and the global contributions.

The MCCD parameters that most affect its behaviour are mentioned above. Their choice greatly relies on the training data set used. Depending on the number of stars available and the complexity of the instrument's PSF, it may be preferable to adopt a more complex model by augmenting the number of local components, r, and the maximum polynomial degree. However, if the stars are not enough to constrain the model, we may end with a model that overfits the training stars. A proper selection of the denoising parameters can control the bias-variance trade-off in the estimation. A high value of the denoising parameter, namely, 3, leads to an extremely denoised model. It will contain a high estimation bias that can be related with a model that cannot capture some spatial variations and fine details of the PSF. On the contrary, if the denoising parameter is close to zero, the only denoising performed by the MCCD is due to the low-rank constraint and therefore the estimations can be rather noisy.

4.6. Results

4.6.1. Comparison between PSFEX, RCA and MCCD-RCA

The first results can be seen in Figs. 4 and 5, where we compare the PSFEx, RCA, and MCCD-RCA algorithms. We observe that MCCD-RCA outperforms the other methods, with an average pixel RMS improvement over PSFEx of 51% and ellipticity RMS improvement ranging from 15% for stars with S/N 10 to 36% for a S/N of 70. The RCA is almost as good as MCCD-RCA for the pixel error, but does not provide good results for the other metrics. This behaviour can be explained by the fact that the model strongly deteriorates for some CCDs, giving extreme ellipticities and sizes values. These deteriorations of the model are not strong enough to produce a large pixel error but causes much more significant errors on the moments. We include in Appendix C RCA's R^2 residual map that shows the catastrophic failure in the modelling of some CCDs.

We can see on the right column of the residual maps in Fig. 5 that PSFEx's ellipticity residuals follow the global pattern from the dataset. This means that the ellipticity is not captured in the model, showing some of the difficulties found when modelling

a global ellipticity pattern using independent models for each CCD. The MCCD-RCA algorithm, which builds up a model for the whole focal plane, does a better job in capturing the global ellipticity pattern. The MCCD-RCA's residuals are smaller and less correlated with the pattern of the dataset. With regard to the third row of Fig. 5, where the size of the simulated PSFs is practically constant, we observe that the MCCD-RCA has slightly larger errors when the training star density is low, as in the bottom-right corner (see Fig. 2).

4.6.2. Comparison between MCCD-POL, MCCD-RCA, and MCCD-HYB

The comparison between the MCCD-POL, MCCD-RCA, and MCCD-HYB methods is shown in Fig. 6. First, we notice that MCCD-POL presents poor performance in most of the metrics. This indicates that the local polynomial model is not able to capture the PSF variations that are left over from the difference of the global model and the observed stars. Hence, even if MCCD-POL has a lower pixel error than PSFEx (see Fig. 4), it has greater ellipticity errors. Capturing these PSF variations properly is essential for obtaining good ellipticity performances. The MCCD-RCA and MCCD-HYB have similar behaviours, but MCCD-HYB uses a mixed approach of a polynomial and graph-based local model outperforms the original MCCD-RCA method in terms of ellipticity components. The average gain in both components of MCCD-HYB with respect to MCCD-RCA is around 18%, proving the utility of using the hybrid approach. This suggests that there are some features related to the PSF shape that can be captured by a simple polynomial model and not by the graph-based model alone. Examples of global and local eigenPSF from the MCCD-HYB model can be seen in Appendix C.

4.7. Comparison of computing resources

The MCCD methods take ~2.9× more CPU-time than PSFEx when compared on the same machine. We evaluate it on the fitting and validation procedures, that is, the estimation of the PSF model and the recovery of PSF at test positions. A relevant detail is that the PSFEx package is coded in the C programming language, while the MCCD methods are completely coded in Python.

5. UNIONS/CFIS experiments

In this section, we compare the MCCD-HYB method with PSFEx using real data from the Ultra-violet Near-Infrared Optical Northern Sky (UNIONS) survey, which is a collaboration between the Panoramic Survey Telescope and Rapid Response System (Pan-STARRS) and CFIS. We use the r-band data from the latter.

5.1. Dataset

We analysed a subset of around $50\,\mathrm{deg}^2$ from the whole CFIS survey area that, in total, will eventually span $5000\,\mathrm{deg}^2$. It corresponds to the subset named W3 described in Erben et al. (2013), and includes 217 exposures. Each CCD from each exposure has been processed independently with SExtractor. The stars were selected in a size-magnitude diagram, in the magnitude range between 18 and 22, and a Full-Width Half Maximum (FWHM) range between 0.3 and 1.5 arcsec. In order to validate the PSF
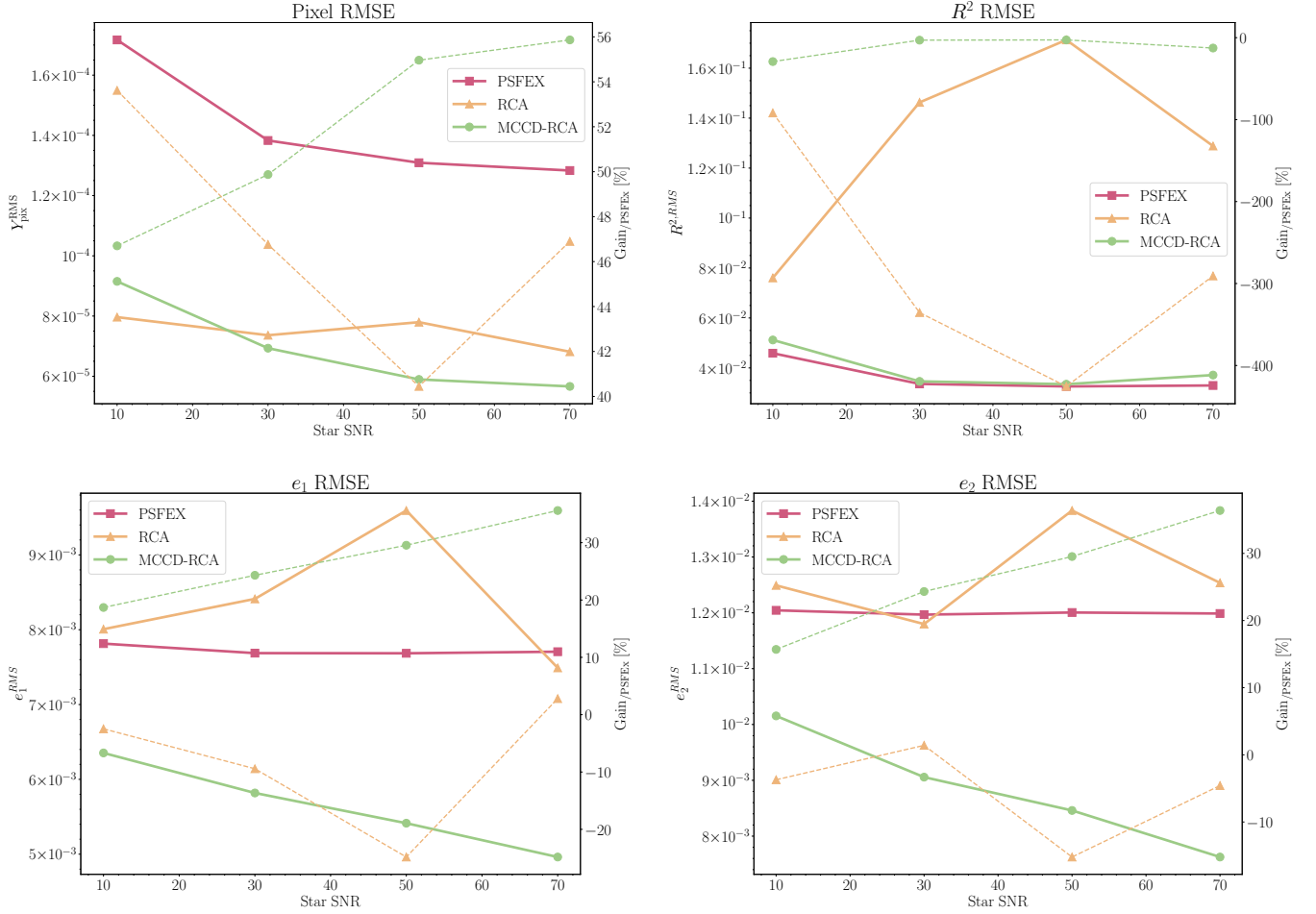
Fig. 4. RMSE on pixels, shape, and size metrics as a function of stars S/N for the three main methods. The RMSE are plotted with solid lines and the gain of the methods with respect to PSFEx are plotted with dashed lines.

models, we randomly split the stars into a testing and a training dataset, trying to estimate the first set of stars while constructing our model only with the second. The training dataset is composed of 80% of the detected stars and the test dataset of the remaining 20%. We consider a fixed threshold on the number of training stars per CCD, meaning that if the number of training stars in a given CCD is less than 30, we discard the CCD. The star density of the training dataset is presented in Fig. 7. The ellipticity and the size of the training stars can be seen in Fig. 8. Each bin represents the mean shape measurement over all the stars with a centroid located within the bin.

5.2. Model parameters

The setup of PSFEx for this experiment is similar to the one used for the simulated images, which can be found in Sect. 4.5. The MCCD-HYB method uses a maximum global polynomial degree of 3, with 16 local components and the denoising parameters K_σ set up to 0.1. In order to compare the star images with the different methods (PSFEx and MCCD-HYB), the models need to match the flux as well as the centre of the star. Hence, after estimating a PSF model at a given star location, the PSF is normalised and shifted to match the star. For this purpose, we use the same intra-pixel shift and flux estimation methods for both PSF models: (i) we estimate the star and the PSF centroids, (ii) we calculate the shift needed by the PSF to match the star and construct a shifting kernel, and (iii) the PSFs are convolved by

their corresponding shifting-kernel. To match the flux, we calculate an α parameter for each test star and PSF that corresponds to the argument that minimises the function, $f(\alpha) = \|I_1 - \alpha I_2\|_2$, where I_1 and I_2 are the star and the PSF, respectively.

5.3. Metric on real data: the Q_p criteria

Performing a comparison between two PSF models with real data is an arduous task since we do not know the shapes and pixel values of the observed stars. However, subtracting our estimated model from an observed star (i.e. pixel residual) should lead to a residual map containing only noise if the model is perfect. The probability of having our model correlated with the noise is extremely small. Therefore, from this point of view, the method with the smallest pixel RMS residual error can be considered as the best. Using all the test stars y_s and our estimates $\hat{y}_s$, we calculate the pixel RMS residual error: Err $= \sqrt{\frac{1}{N_i N_s} \sum_s \sum_i (y_{s,i} - \hat{y}_{s,i})^2}$, where N_s is the number of stars and N_i is the number of pixels we consider in a given image when we use a 10 pixel radius circle from the centre of the residual images. The noise standard deviation σ_{noise} is calculated from the stars only using the pixels outside the aforementioned circle. For a perfect modelling, we would have Err $\approx \sigma_{\text{noise}}$, and we define the Q_{p_1} metric as:

$$Q_{p_1} = \left(\text{Err}^2 - \sigma_{\text{noise}}^2 \right)^{1/2}. \tag{32}$$

MCCD-RCA e_1 residual

PSFEx e_1 residual

MCCD-RCA e_2 residual

PSFEx e_2 residual

MCCD-RCA $\Delta R^2/R^2$ residual

PSFEx $\Delta R^2/R^2$ residual
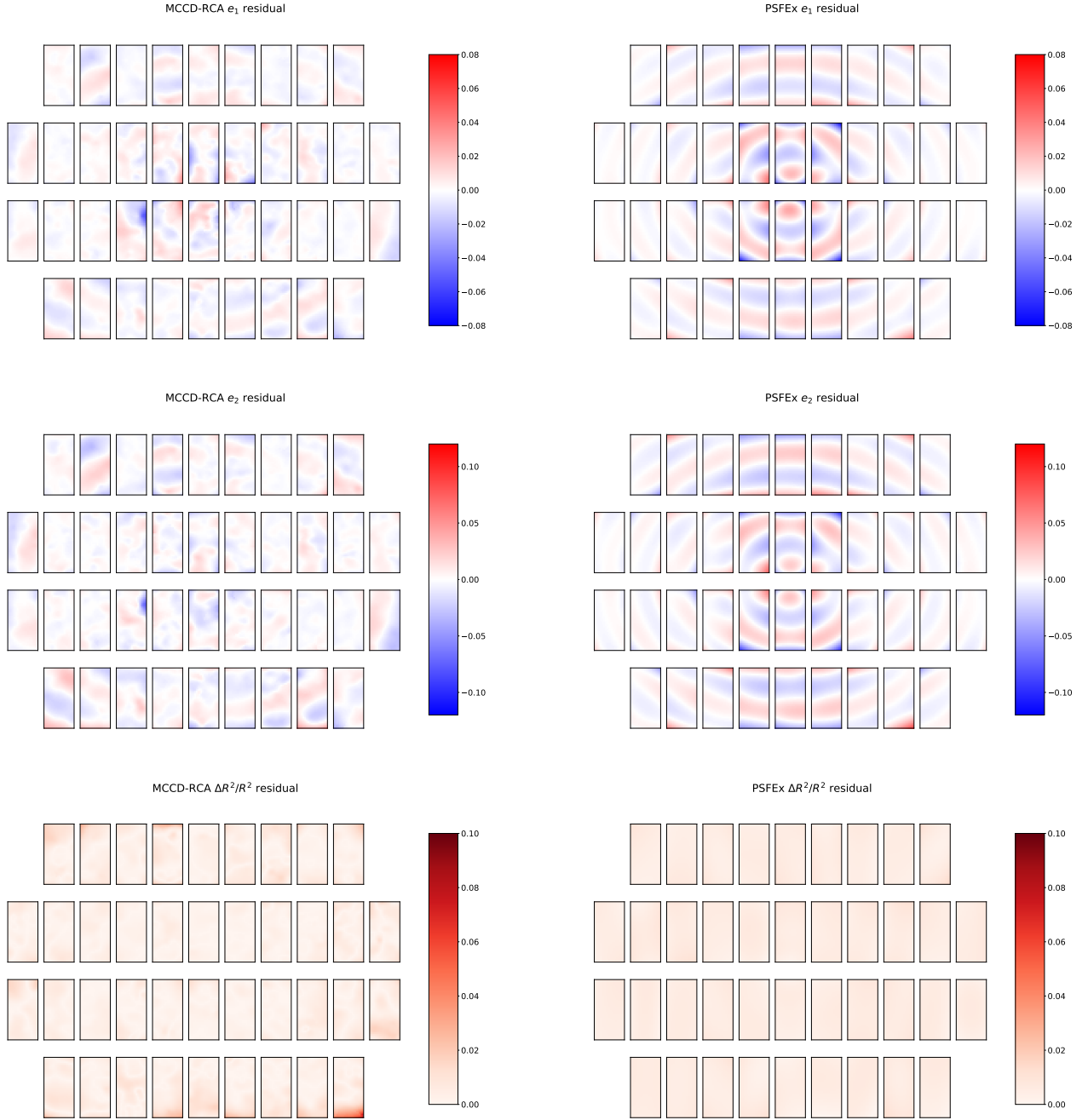
Fig. 5. Moment residual maps comparing the MCCD-RCA algorithm on the *left* and the PSFEx algorithm on the *right*. They are obtained by subtracting the model's and the test star's measured shape and size metrics and plotting them on their corresponding position over the focal plane. The S/N value of the star dataset is 50.

We next introduce two metrics to quantify how noisy the models are. The variance of the PSF model for the test stars s reads $\sigma_s^2 = [\text{Var}(y_s - \hat{y}_s) - \sigma_{\text{noise}}^2(y_s)]_+$, where $\text{Var}(\cdot)$ is a usual variance estimator, the operator $[\cdot]_+$ sets to zero negative values and $\sigma_{\text{noise}}^2(y_s)$ is the noise variance estimation for a single star. We present the Q_{p_2} and Q_{p_3} metrics in the following equations:

$$Q_{p_2} = \left(\frac{1}{N_s}\sum_s \sigma_s^2\right)^{1/2}, \qquad Q_{p_3} = \left(\frac{1}{N_s}\sum_s (\sigma_s^2 - Q_2^2)^2\right)^{1/4}. \quad (33)$$

The Q_{p_2} metric represents the modelling error expectation for a given star and the Q_{p_3} metric indicates the fluctuation of the

modelling error. A perfect PSF model would give values close to zero for the three metrics.

5.4. Results

The main results of the experiment are synthesised in Table 2, where the Q_p criteria are given. In the Q_{p_1} column, we can observe a 22% gain of the MCCD-HYB method with respect to PSFEx. From Q_{p_2} and Q_{p_3} metrics, we also conclude that the MCCD-HYB model is considerably less noisy than the one from PSFEx.

In order to explore potential remaining structure in the residuals, we stack together the residuals for all 534 test stars from a
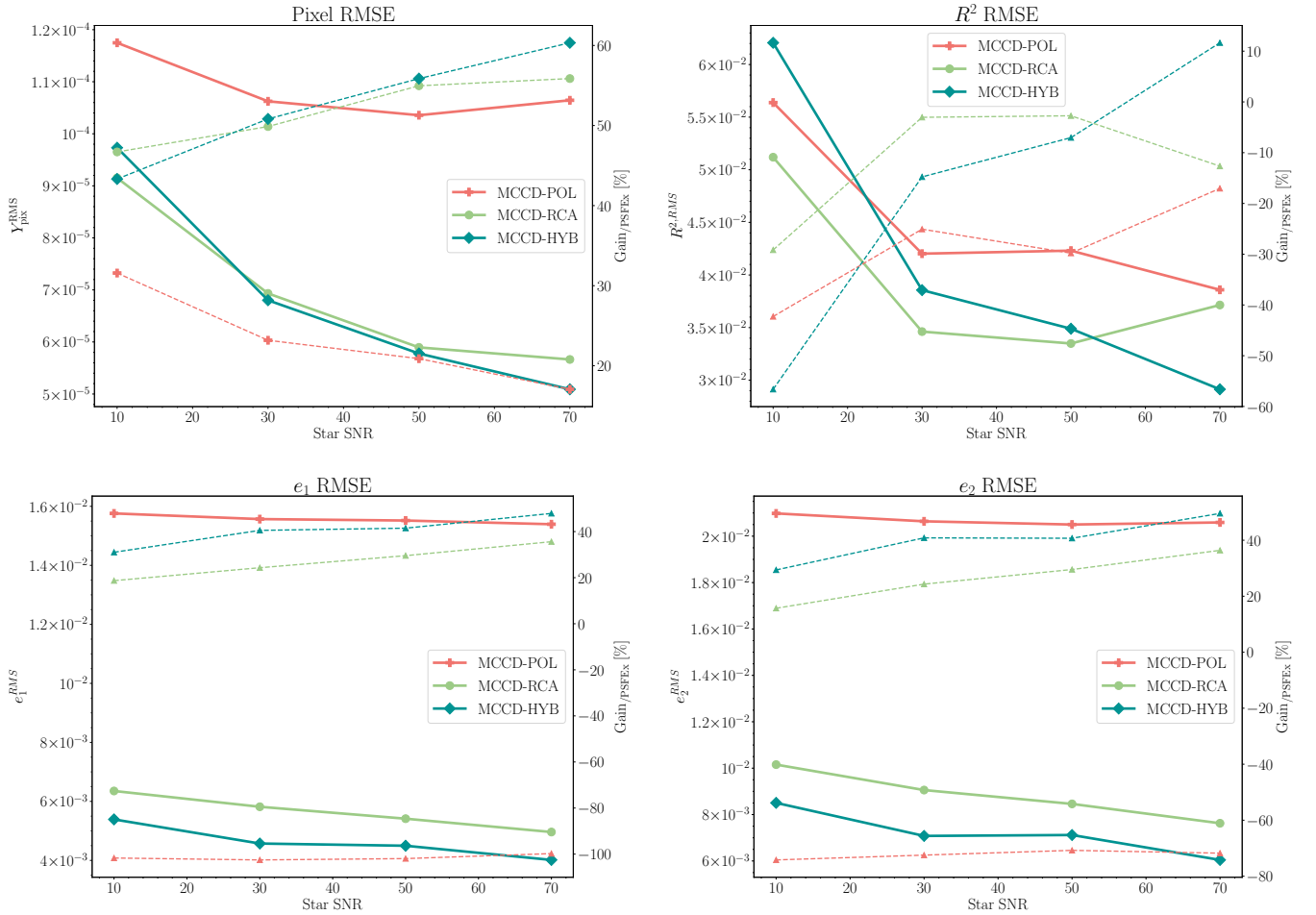
Fig. 6. Comparison of the performance of MCCD-POL, MCCD-RCA, and MCCD-HYB methods in terms of the RMSE on the pixels, the shape and the size metric of the star's S/N. The RMSE are plotted on a solid line and the gain of the method with respect to PSFEx is plotted on dashed lines.
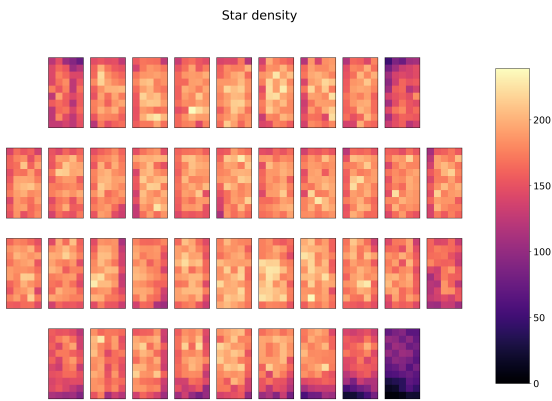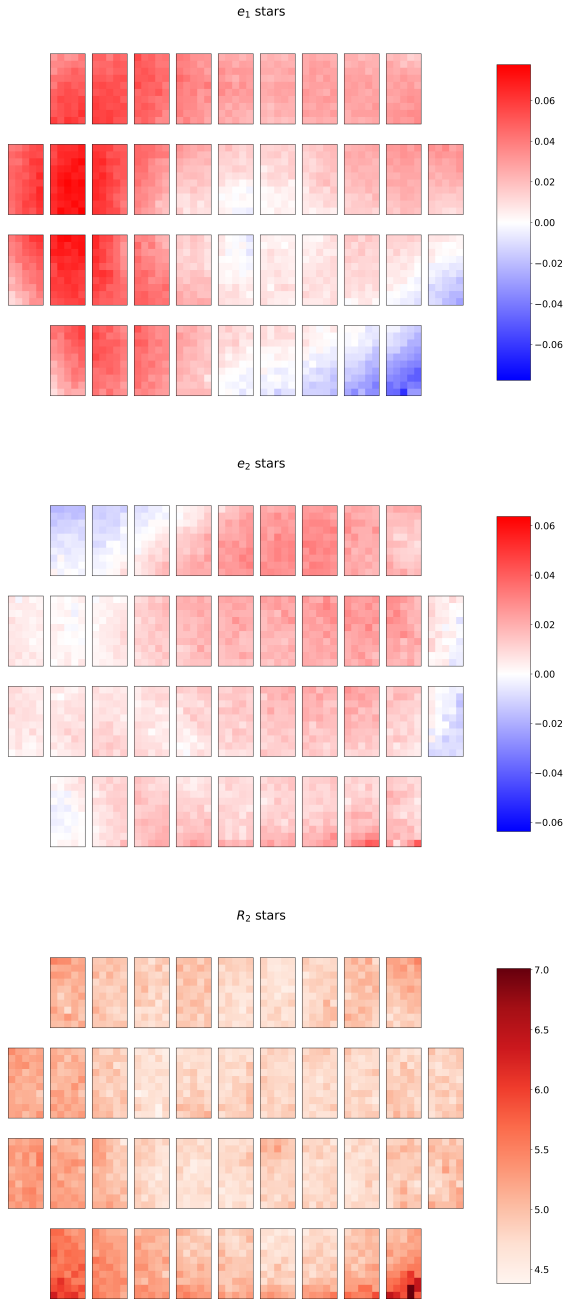

Fig. 7. Star density of all the training dataset with respect to their position on the MegaCam's focal plane. We have on average 1560 training stars per exposure.

random exposure. These are shown, along with the stacking of the test stars themselves, in Fig. 9. We can see that PSFEx has a sharper stacked error compared to MCCD-HYB. This could indicate that our algorithm is better at capturing the size of the PSF, as the peak of the residual is directly related to it. Considering that there is no trace of shifting errors and that we are calculating the flux optimally, a greater mismatch in the size of the PSF equals to a greater peak pixel error on the residual. The

third row presents the mean of the stacked absolute value of the residuals for both of the PSF models so that the residuals can not cancel themselves. We observe the same behaviour described above with the PSFEx pixel error distribution being sharper but more centred. It is also possible to notice the higher noise PSFEx has when compared to the MCCD-HYB model.

Figure 10 presents examples of star image reconstructions by the two different PSF models, PSFEx and MCCD-HYB, and their corresponding residuals. The proposed method yields a near noiseless model when compared to PSFEx, as can clearly be seen on the top-left and bottom-right stars of Fig. 10, where the stars have low S/N of 19.3 and 4.2, respectively. Both models share a good estimation of the bottom-right star, which comes a low-stellar-density region of the focal plane (the bottom-right corner, as can be seen in Fig. 7). On the bottom-left star of Fig. 10, we observe a similar type of error as that appearing in Fig. 9.

It is difficult to derive conclusions of different PSF model performances based on the shape measurement of noisy stars due to its high stochasticity. Nevertheless, driven by the comments from DES Y1 (Zuntz et al. 2018) on the residual mean size offset from the PSFEx model, we conducted a study with our data. We measured the size from the training stars and from both calculated PSF models, PSFEx and MCCD-HYB, and then computed the residual. The RMS residual size of the $\Delta R^2/R^2$ value gave 4.82×10^{-2} for PSFEx and 4.02×10^{-2} for MCCD-HYB. This represents a 16% gain of our proposed algorithm.

Fig. 8. Ellipticities and size maps measured on the training stars of our CFIS dataset.

Table 2. Q_p criterion using all test stars of the W3 dataset from CFIS.

Method	Q_{p_1}	Q_{p_2}	Q_{p_3}
PSFEx	15.56	8.13	14.31
MCCD-HYB	**12.14**	**6.68**	**10.86**
Gain$_{\text{PSFEx}}$	22%	18%	24%
Noise Std. Dev. (σ_{noise})		15.83	

Notes. The gain of the MCCD-HYB with respect to PSFEx and the noise standard deviation are also presented.
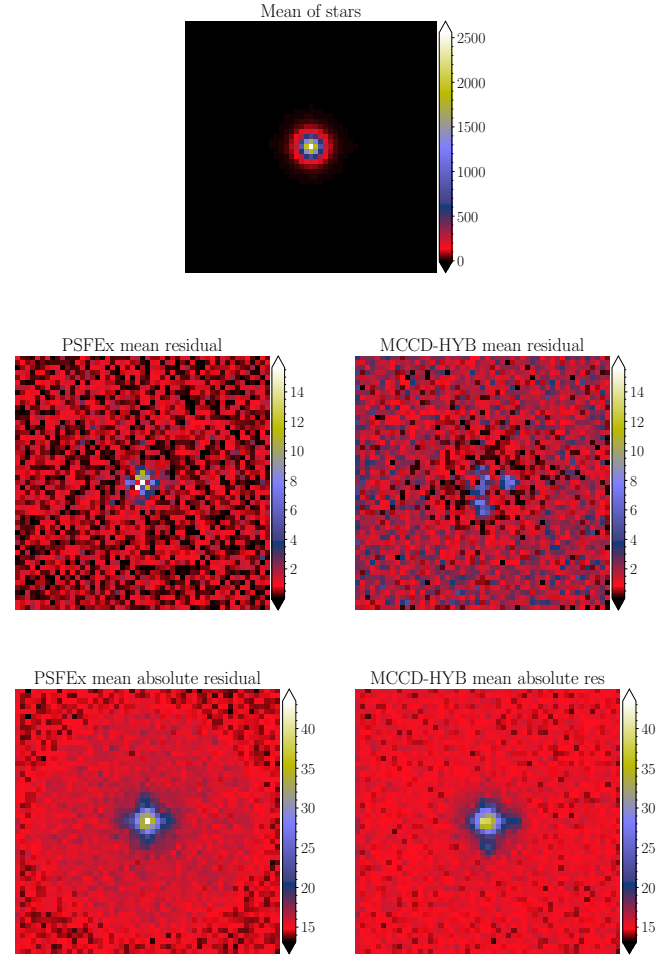
Fig. 9. Stacked star profile from all 534 test stars in a random CFIS exposure (id. 2099948) (*top*), corresponding stacked residuals after subtraction by the PSFEx (*middle left*) or MCCD-HYB (*middle right*) PSF models. The *bottom row* includes the stacked absolute value of the residuals.

Figure 11 presents in the left column the histogram of the residuals and in the right column the histograms of the size metrics. We notice that the MCCD-HYB algorithm has a sharper residual size around zero. The figure also includes the mean of the residuals for each PSF model. This shows that both models tend to overestimate the size of the PSF. However, the MCCD-HYB model presents a 30% gain in the mean residual size with respect to PSFEx, indicating a smaller bias in the shape.

6. Reproducible research

In the spirit of reproducible research, the MCCD-RCA algorithm is to be publicly available on the CosmoStat's Github[11],

---
[11] https://github.com/CosmoStat/mccd

including the material needed to reproduce the simulated experiences. The MCCD PSF modelling software will be included in the CFIS shape measurement pipeline Guinot et al. (in prep.).

7. Conclusion

In this work, we present a family of non-parametric PSF (Point Spread Function) modelling methods coined MCCD, including its best-performing extension MCCD-HYB, which are built upon the existing Resolved Component Analysis (RCA) method and are capable of constructing PSF models that span all the charge-coupled devices (CCDs) from an instrument's focal plane
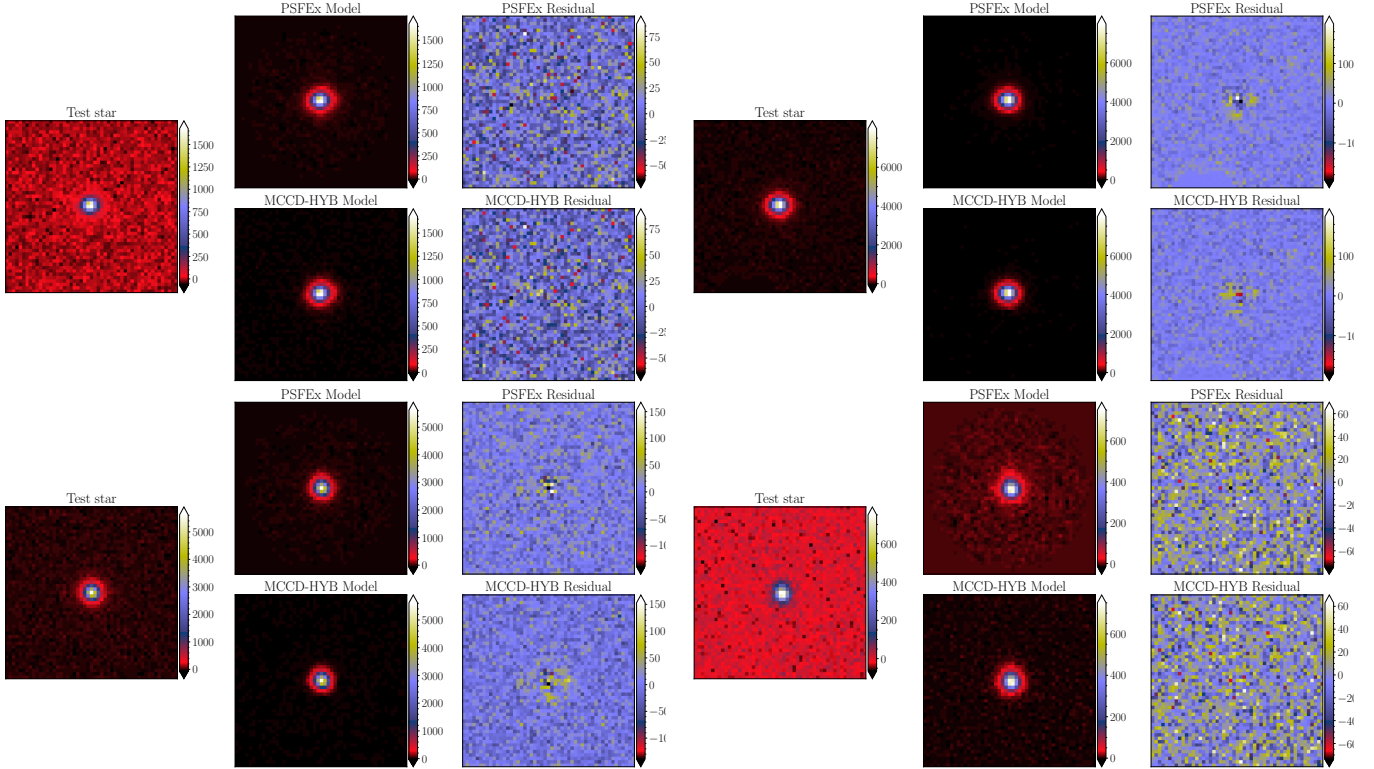
Fig. 10. Examples of real CFIS test stars, the estimations of both methods, and the pixel residuals. The exposures are the same as in Fig. 9. We present four test stars with the estimated PSF models and the corresponding residuals. *Top-left star*: a star extracted from the top-left corner of the focal plane with a S/N of 19. *Top-right star*: a star leading to a high error for both methods. *Bottom-left star*: a star located in the centre of the focal, with a relatively high S/N of 160. *Bottom-right star*: a star located in the bottom-right corner of the focal plane, with a low S/N of 4.
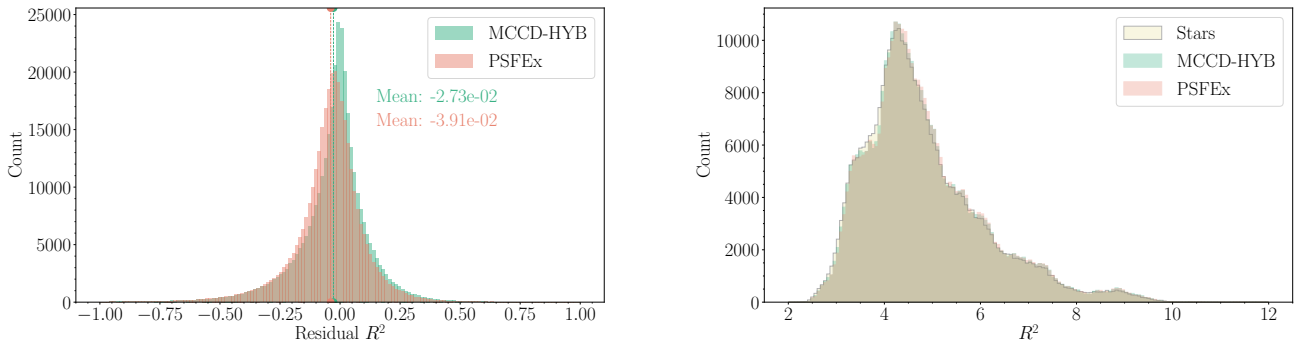
Fig. 11. Histograms showing the distribution of the size metric over the training stars and their residuals for both PSF models, `PSFEx` and MCCD-HYB.

at once. Naturally, the use of more stars for the training allows us to build more complex models that can capture evasive features. Our model is composed of global components, spanning all the CCDs, and local components that are CCD-specific. By using this structure, we can better capture global patterns and features that might be lost when using only a local model like in RCA or, the widely used algorithm, `PSFEx`.

The method was first tested with a set of simulated PSFs following a real star spatial distribution over MegaCam's focal plane, an instrument from the Canada-France-Hawaii Telescope(CFHT). Its use leads to better performance in all the evaluated metrics when compared to `PSFEx`. We then tested the method on a set of real CFIS images, an imaging survey based on CFHT, in order to confirm that it can handle real data. Our

method achieves a smaller pixel root mean square (RMS) residual than `PSFEx` and the estimated model is considerably less noisy.

The performance gain of the MCCD methods over `PSFEx` is higher when using our simulated dataset than when using the real dataset. This can be explained by the fact that our simulated dataset shows more intricate variations in the PSF than the real data does and MCCD is better at capturing such strong variations.

The proposed method can naturally handle more complex PSF profiles, such as those expected from space-based instruments. The RCA method was tested with *Euclid*-like simulated PSFs and has demonstrated a better performance than `PSFEx` (Ngolè et al. 2016; Schmitz et al. 2020). Therefore, we expect to

have an even superior performance in this scenario with MCCD. Thanks to its formulation, it can also handle super-resolution, making it suitable for undersampled data.

Despite the good performance of the method, there is still room for improvement. A natural straightforward extension for the MCCD algorithms would be to replace the denoising strategy by one more suited for the specificities of the PSFs we work with. This could be accomplished by using a deep neural network as the denoiser (Ronneberger et al. 2015; Ye et al. 2018).

References

Aliprantis, C., & Border, K. 2007, Infinite Dimensional Analysis A Hitchhiker's Guide (Springer)
Astropy Collaboration (Robitaille, T. P., et al.) 2013, A&A, 558, A33
Astropy Collaboration (Price-Whelan, A. M., et al.) 2018, AJ, 156, 123
Beck, A. 2017, First-Order Methods in Optimization, MOS-SIAM Series on Optimization (Society for Industrial and Applied Mathematics)
Beck, A., & Teboulle, M. 2009, SIAM J. Img. Sci., 2, 183
Beltramo-Martin, O., Marasco, A., Fusco, T., et al. 2020, MNRAS, 494, 775
Bendinelli, O., Parmeggiani, G., & Zavatti, F. 1988, J. Astrophys. Astron., 9, 17
Bertin, E. 2011, in Astronomical Data Analysis Software and Systems XX, eds. I. N. Evans, A. Accomazzi, D. J. Mink, & A. H. Rots, ASP Conf. Ser., 442, 435
Bertin, E., & Arnouts, S. 1996, A&AS, 117, 393
Bolte, J., Sabach, S., & Teboulle, M. 2014, Math. Program., 146, 459
Chung, F. R. K. 1997, Spectral Graph Theory (American Mathematical Society)
Condat, L. 2013, J. Optim. Theor. Appl., 158, 460
Cropper, M., Hoekstra, H., Kitching, T., et al. 2013, MNRAS, 431, 3103
Erben, T., Hildebrandt, H., Miller, L., et al. 2013, MNRAS, 433, 2545
Farrens, S., Grigis, A., El Gueddari, L., et al. 2020, Astron. Comput., 32, 100402
Gentile, M., Courbin, F., & Meylan, G. 2013, A&A, 549, A1

Golub, G. H., & Van Loan, C. F. 1996, Matrix Computations, 3rd edn. (USA: Johns Hopkins University Press)
Herbel, J., Kacprzak, T., Amara, A., Refregier, A., & Lucchi, A. 2018, JCAP, 07, 054
Hirata, C., & Seljak, U. 2003, MNRAS, 343, 459
Hoffmann, S. L., & Anderson, J. 2017, A Study of PSF Models for ACS/WFC, Instrument Science Report ACS, 2017-8
Hunter, J. D. 2007, Comput. Sci. Eng., 9, 90
Ibata, R. A., McConnachie, A., Cuillandre, J. C., et al. 2017, ApJ, 848, 128
Infante-Sainz, R., Trujillo, I., & Román, J. 2019, MNRAS, 491, 5317
Jarvis, M., Sheldon, E., Zuntz, J., et al. 2016, MNRAS, 460, 2245
Jarvis, M., Bernstein, G. M., Amon, A., et al. 2021, MNRAS, 501, 1282
Jee, M. J., Blakeslee, J. P., Sirianni, M., et al. 2007, PASP, 119, 1403
Jia, P., Wu, X., Yi, H., Cai, B., & Cai, D. 2020a, AJ, 159, 183
Jia, P., Li, X., Li, Z., Wang, W., & Cai, D. 2020b, MNRAS, 493, 651
Kilbinger, M. 2015, Rep. Prog. Phys., 78, 086901
Kluyver, T., Ragan-Kelley, B., Pérez, F., et al. 2016, ELPUB
Krist, J. 1995, in Astronomical Data Analysis Software and Systems IV, eds. R. A. Shaw, H. E. Payne, & J. J. E. Hayes, ASP Conf. Ser., 77, 349
Kuijken, K., Heymans, C., Hildebrandt, H., et al. 2015, MNRAS, 454, 3500
Laureijs, R., Amiaux, J., Arduini, S., et al. 2011, ArXiv e-prints [arXiv:1110.3193]
Liang, J., Luo, T., & Schönlieb, C. B. 2018, ArXiv e-prints [arXiv:1811.01430]
Mandelbaum, R., Hirata, C. M., Seljak, U., et al. 2005, MNRAS, 361, 1287
Miller, L., Heymans, C., Kitching, T., et al. 2013, MNRAS, 429, 2858
Ngolè, F., & Starck, J.-L. 2017, SIAM J. Imaging Sci., 10, 1549
Ngolè, F., Starck, J.-L., Ronayette, S., Okumura, K., & Amiaux, J. 2015, A&A, 575, A86
Ngolè, F., Starck, J.-L., Okumura, K., Amiaux, J., & Hudelot, P. 2016, Inverse Prob., 32, 124001
Parikh, N., & Boyd, S. 2014, Found. Trends Optim., 1, 127
Perez, F., & Granger, B. E. 2007, Comput. Sci. Eng., 9, 21
Peyré, G., & Cuturi, M. 2018, Comput. Trans. Optim.
Piotrowski, L. W., Batsch, T., Czyrkowski, H., et al. 2013, A&A, 551, A119
Ronneberger, O., Fischer, P., & Brox, T. 2015, International Conference on Medical Image Computing and Computer-assisted Intervention (Springer), 234
Rowe, B., Jarvis, M., Mandelbaum, R., et al. 2015, Astron. Comput., 10, 121
Schmitz, M. A., Heitz, M., Bonneel, N., et al. 2018, SIAM J. Imaging Sci., 11, 643
Schmitz, M. A., Starck, J. L., Mboula, F. N., et al. 2020, A&A, 636, A78
Schrabback, T., Hartlap, J., Joachimi, B., et al. 2010, A&A, 516, A63
Starck, J.-L., Murtagh, F., & Bertero, M. 2011, Handbook of Mathematical Methods in Imaging (Springer), 1489
Starck, J.-L., Murtagh, F., & Fadili, J. 2015, Sparse Image and Signal Processing: Wavelets and Related Geometric Multiscale Analysis, 2nd edn. (Cambridge University Press)
Tyson, J. A., Zhan, H., Knox, L., & LSST Collaboration 2006, Am. Astron. Soc. Meeting Abstracts, 209, 86.08
van der Walt, S., Colbert, S. C., & Varoquaux, G. 2011, Comput. Sci. Eng., 13, 22
Virtanen, P., Gommers, R., Oliphant, T. E., et al. 2020, Nat. Meth., 17, 261
Xu, Y., & Yin, W. 2013, SIAM J. Imaging Sci., 6, 1758
Ye, J. C., Han, Y., & Cha, E. 2018, SIAM J. Imag. Sci., 11, 991
Zuntz, J., Sheldon, E., Samuroff, S., et al. 2018, MNRAS, 481, 1149

Appendix A: RCA regularisations

In this section, we provide more detailed descriptions for each regularisation that we use in our local RCA model:

1. *Low rank*: PSF variations can be explained by a small number of eigenPSFs. This constraint can be enforced by the proper choice of two parameters, the number of local, r_k, and global, $\tilde{r}$, eigenPSFs. These parameters are directly linked with the complexity of the model we are addressing for each given case and its selection naturally depends on the PSF field we will be facing. It is important to allow the model a certain complexity so that it can correctly capture the PSF field's variations but it should not be much more complex as the model will tend to overfit the noisy observations and, therefore, lose its generalising power for estimating the PSF in galaxy positions.
2. *Positivity*: the reconstructed PSFs $\hat{H}$ should only contain non-negative pixel values.
3. *Sparsity*: the observed PSFs are structured images; a way to promote our model to follow this structured behaviour is to enforce the sparsity of the eigenPSFs in an appropriate basis.
4. *Spatial constraints*: the regularity of the PSF field $\mathcal{H}$ means that the smaller the distance between two PSFs positions u_i, u_j the smaller the difference between their representations should be $\mathcal{H}(u_i), \mathcal{H}(u_j)$. This regularity can be achieved by enforcing constraint in the coefficient matrices $A_k, \tilde{A}_k$; for example, the line l of A_k corresponds to the contribution of eigenPSF l to the n_{star}^k stars in CCD k located in positions $(u_i)_{i=1}^{n_{star}^k}$. The closer the positions, the closer the coefficient values should be.

Appendix B: Shape and size definitions

The ellipticity parameters and the size are defined in terms of the moments of the surface brightness profile $I(x, y)$ following (Hirata & Seljak 2003):

$$\bar{\mu} = \frac{\int \mu \, I(x, y) \, w(x, y) \, dxdy}{\int I(x, y) \, w(x, y) \, dxdy}, \tag{B.1}$$

$$M_{\mu\nu} = \frac{\int I(x, y) \, (\mu - \bar{\mu}) \, (\nu - \bar{\nu}) \, w(x, y) \, dxdy}{\int I(x, y) w(x, y) \, dxdy}, \tag{B.2}$$

where $\mu, \nu \in \{x, y\}$ and $w(x, y)$ is weight window to avoid noise related issues. The size is defined as:
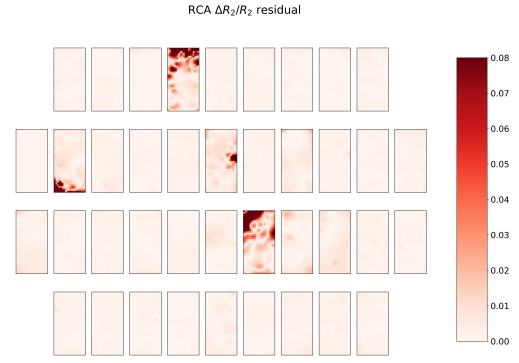
$$T = R^2 = M_{xx} + M_{yy}, \tag{B.3}$$

and the ellipticities are defined as:

$$e = e_1 + ie_2 = \frac{(M_{xx} - M_{yy}) + i \, 2M_{xy}}{T}. \tag{B.4}$$

The adaptive moment measurement from HSM gives σ as output which relates to our size metric as $R^2 = 2\sigma^2$.

Appendix C: Additional figures

In this appendix, we include the additional figures, Figs. C.1 and C.2.

Fig. C.1. Residual $\Delta R_2/R_2$ map of the RCA algorithm of stars with a S/N of 50. The CCDs where the RCA model is having degeneracies that can be clearly spotted on the map.
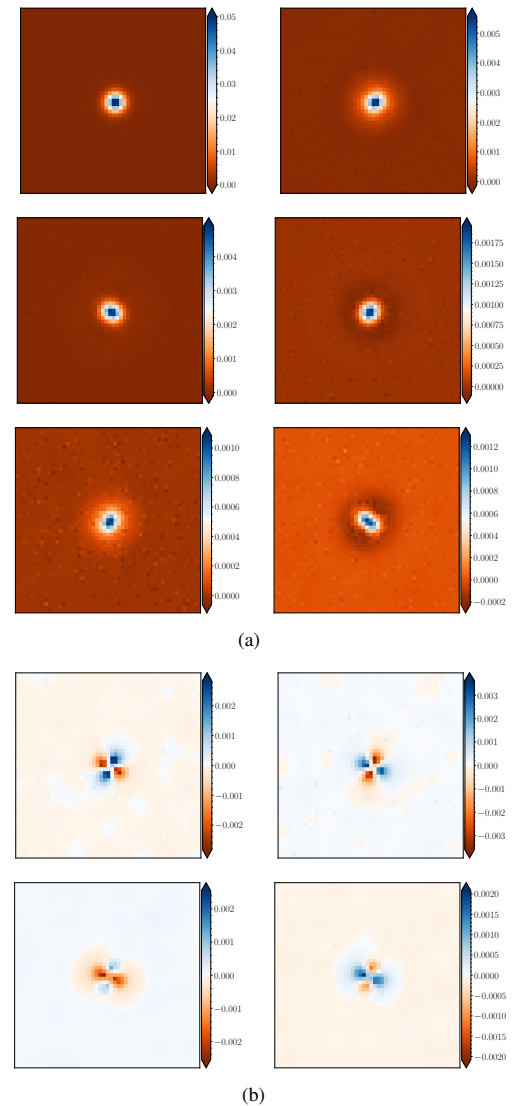
Fig. C.2. Example eigenPSFs extracted from the MCCD-HYB PSF model trained on the simulated dataset with a S/N of 70. The local eigenPSFs were extracted from the graph's spatial constraint of a central CCD. It can be seen from the eigenPSFs that the global model is specialising on the shape of the PSF while the local model specialises on capturing its ellipticity. It is also worth to mention that the first global eigenPSF found on the first row provides the baseline isotropic PSF the model uses. (*a*) Global eigenPSFs. (*b*) Local eigenPSFs.

Appendix D: Optimisation methods

In this appendix, we include details on the practical resolution of the four optimisation problems seen in Algorithm 1. For more information about proximal operators and proximal algorithms, we refer to Parikh & Boyd (2014) and Beck (2017).

D.1. Problem (III)

As in most of the optimisation problems, the algorithm used depends on the objective function we work with. In this case, we use the primal-dual algorithm 3.1 in Condat (2013)[12]. The main motivation resides in the nature of the constraints we use when optimising over S_k, as we face one smooth and two non-smooth terms, and a linear operator. The optimisation algorithm aims at solving the following problem:

$$\text{Find } \hat{x} \in \underset{x \in X}{\arg\min} \left[F(x) + G(x) + H(L(x)) \right], \quad (D.1)$$

where: (i) F is convex, differentiable and its gradient is L-Lipschitz continuous; (ii) G and H are proximable functions that should have closed form proximal operators; (iii) L is a bounded linear operator; and (iv) the set of minimisers of the aforementioned optimisation problem is nonempty. It is straightforward to identify the different functions in the optimisation of the local S_k matrix which match the formulation of Eq. (D.1). Following the notation we use throughout the paper, let $F_k(x = (S_1, \ldots, S_N, \tilde{S}, \alpha_1, \ldots, \alpha_N, \tilde{\alpha})) = \frac{1}{2}\|Y_k - \mathcal{F}_k(\hat{H}_k)\|_F^2$, with $\hat{H}_k = S_k \alpha_k V_k^\top + \tilde{S} \tilde{\alpha} \Pi_k$, and $G(S_k) = \sum_i \|\boldsymbol{w}_{k,i} \odot \Phi \mathbf{s}_{k,i}\|_1$. Let $H(S_k) = \iota_+(S_k)$ and the linear operator L be $L(S_k) = S_k \alpha_k^{(l)} V_k^\top + \tilde{S}^{(l)} \tilde{\alpha}^{(l)} \Pi_k$. For the moment, we consider Φ to be the identity.

To solve the algorithm, we need the proximal operator of H^*, the adjoint function of H, the proximal operator of G, and the gradient of F with its Lipschitz constant.

Starting with H, the proximal operator of H^* can be calculated directly using the proximal operator of the function H itself by means of the Moreau decomposition (Beck 2017, Theorem 6.44). The proximal operator of an indicator function over a set C is the orthogonal projection over that set. Therefore, we note $[X]_+$ the projection of $X \in \mathbb{R}^{n \times m}$ onto the positive orthant, that is,

$$\text{prox}_{\iota_+(\cdot)}(X) = [X]_+ \rightarrow [X_{i,j}]_+ = \begin{cases} X_{i,j} & \text{if } X_{i,j} \geq 0, \\ 0 & \text{otherwise.} \end{cases} \quad (D.2)$$

Continuing with G, the proximal operator of the ℓ_1 norm is the soft thresholding operator which can be defined component-wise, for $x, \lambda \in \mathbb{R}$, as

$$\text{SoftThresh}_\lambda(x) = (|x| - \lambda)_+ \text{sign}(x) = \begin{cases} x - \lambda, & x \geq \lambda, \\ 0, & |x| < \lambda, \\ x + \lambda, & x \leq -\lambda. \end{cases} \quad (D.3)$$

We name $L_{\nabla_{S_k} F(\cdot)}$ the Lipschitz constant of F's gradient. The next equations resume what we need to use the chosen optimisation algorithm:

$$\nabla_{S_k} F(S_k) = -\mathcal{F}_k^*(Y_k - \mathcal{F}_k(\hat{H}_k))V_k \alpha_k^\top, \quad (D.4)$$

$$L_{\nabla_{S_k} F(\cdot)} = \rho(\mathcal{F}_k^* \circ \mathcal{F}_k)\rho(\alpha_k V_k^\top (\alpha_k V_k^\top)^\top), \quad (D.5)$$

$$\text{prox}_{\tau G(\cdot)}([\mathbf{s}_{k,i}]_j) = \text{SoftThresh}_{\tau[\boldsymbol{w}_{k,i}]_j}([\mathbf{s}_{k,i}]_j), \quad (D.6)$$

$$\text{prox}_{\sigma H^*(\cdot)}(X) = X - (X)_+, \quad (D.7)$$

where the proximal operator of G is defined component-wise, the notation $[\mathbf{s}_{k,i}]_j$ represents the element j of the i column vector of matrix S_k, $\mathcal{F}_k^*$ is the adjoint operator of $\mathcal{F}_k$, and $\rho(\cdot)$ is the spectral radius[13] that we calculate using the power method (Golub & Van Loan 1996). For the algorithm's parameters τ and σ, based on Theorem 3.1 from Condat (2013), we use:

$$\tau = \frac{1}{\alpha L_{\nabla_{S_k} F(\cdot)}}, \qquad \sigma = \frac{\alpha L_{\nabla_{S_k} F(\cdot)}}{2\|L\|_{\text{op}}^2}, \quad (D.8)$$

where $\| \cdot \|_{\text{op}}$ is the operator norm (Aliprantis & Border 2007) and α is a parameter we set to $3/2$. Being L a bounded linear operator we can calculate $\|L\|_{\text{op}}$ as $\sqrt{\rho(L^*L)}$ being L^* its adjoint operator.

We now consider the case where Φ is not the identity, but it is orthonormal, $\Phi^T \Phi = I$. We can adapt the soft thresholding operator in order to cope with the G term. This would be $\mathbf{s}_k \rightarrow \Phi^T \text{SoftThresh}_{\tau \boldsymbol{w}_k}(\Phi \mathbf{s}_k)$. When using undecimated wavelets as the starlets, the orthonormal condition is not met. Nevertheless, they are tight frames whose Gram matrix is close to the identity which means that the presented formulation will be a good approximation. We refer to Starck et al. (2015) for more information on wavelets.

D.2. The remaining optimisation problems

We deal in a similar way with the problems (II) and (IV) from Algorithm 1 using the same optimisation method proposed in Condat (2013). On the other hand, for problem (I), we use the optimisation algorithm in Liang et al. (2018). This is due to the fact that we are neglecting the positivity constraint as we account for it when optimising over the other variables. In order to use these algorithms we need to compute the gradients of the differentiable term of each problem as follows:

$$\nabla_{S_k} F_k(x) = -\mathcal{F}_k^*(Y_k - \mathcal{F}_k(\hat{H}_k))(\alpha_k V_k^\top)^\top, \quad (D.9)$$

$$\nabla_{\alpha_k} F_k(x) = -S_k^\top \mathcal{F}_k^*(Y_k - \mathcal{F}_k(\hat{H}_k))V_k, \quad (D.10)$$

$$\nabla_{\tilde{S}} F(x) = \sum_{k=1}^N \nabla_{\tilde{S}} F_k(x) = \sum_{k=1}^N -\mathcal{F}_k^*(Y_k - \mathcal{F}_k(\hat{H}_k))(\tilde{\alpha} \Pi_k)^\top, \quad (D.11)$$

$$\nabla_{\tilde{\alpha}} F(x) = \sum_{k=1}^N \nabla_{\tilde{\alpha}} F_k(x) = \sum_{k=1}^N -\tilde{S}^\top \mathcal{F}_k^*(Y_k - \mathcal{F}_k(\hat{H}_k))\Pi_k^\top, \quad (D.12)$$

where $F = \sum_{k=1}^N F_k = \frac{1}{2}\|Y - \mathcal{F}(H + \tilde{S}\tilde{\alpha}\Pi)\|_F^2$. Concerning the global optimisation over $\tilde{S}$ and $\tilde{A}$, we need to consider all the CCDs when computing the gradient. So we can reformulate the global formulas as:

$$\nabla_{\tilde{S}} F(x) = -\mathcal{F}^*(Y - \mathcal{F}(\hat{H}))(\tilde{\alpha}\Pi)^\top, \quad (D.13)$$

$$\nabla_{\tilde{\alpha}} F(x) = -\tilde{S}^\top \mathcal{F}^*(Y - \mathcal{F}(\hat{H}))\Pi^\top. \quad (D.14)$$

An approximation for the Lipschitz constants of the different gradients can be calculated as:

$$L_{S_k} = \rho(\mathcal{F}_k^* \circ \mathcal{F}_k)\rho(\alpha_k V_k^\top (\alpha_k V_k^\top)^\top), \tag{D.15}$$

$$L_{\alpha_k} = \rho(\mathcal{F}_k^* \circ \mathcal{F}_k)\rho(S_k^\top S_k)\rho(V_k^\top V_k), \tag{D.16}$$

$$L_{\tilde{S}} = \rho(\mathcal{F}^* \circ \mathcal{F})\rho(\tilde{\alpha}\Pi(\tilde{\alpha}\Pi)^\top), \tag{D.17}$$

$$L_{\tilde{\alpha}} = \rho(\mathcal{F}^* \circ \mathcal{F})\rho(\tilde{S}^\top \tilde{S})\rho(\Pi \Pi^\top), \tag{D.18}$$

where $\rho(\cdot)$ is the spectral radius.

Finally, we also need the proximal operator of the indicator function over the unit-ball $\iota_\mathcal{B}(\cdot)$, where $\mathcal{B} = \{x \in \mathbb{R}^n \mid \|x\|_2 = 1\}$. It can be computed as:

$$\text{prox}_{\iota_\mathcal{B}(\cdot)}(x) = \frac{x}{\|x\|_2}. \tag{D.19}$$

D.3. Sparsity enforcement parameters

There are two moments when we enforce sparsity during the optimisation. First, when we denoise the eigenPSFs by the use of the ℓ_1 norm as in Eq. (22). The $\boldsymbol{w}$ weights are set depending on a noise estimation of the observed images, and the parameters K_σ^{Loc} and K_σ^{Glob}. The noise standard deviation is estimated using the median absolute deviation. The higher the K_σ parameters are set, the higher the thresholding and the denoising will be. Second, when we enforce the spatial constraints through α sparsity. In this case, we follow the sparsity enforcement proposed in Ngolè et al. (2016).