A&A 549, A1 (2013)
DOI: 10.1051/0004-6361/201219739

Astronomy & Astrophysics

Interpolating point spread function anisotropy

M. Gentile, F. Courbin, and G. Meylan

Laboratoire d'astrophysique, École Polytechnique Fédérale de Lausanne (EPFL), Observatoire de Sauverny, 1290 Versoix, Switzerland
e-mail: marc.gentile@epfl.ch

Received 3 June 2012 / Accepted 19 September 2012

ABSTRACT

Planned wide-field weak lensing surveys are expected to reduce the statistical errors on the shear field to unprecedented levels. In contrast, systematic errors like those induced by the convolution with the point spread function (PSF) will not benefit from that scaling effect and will require very accurate modeling and correction. While numerous methods have been devised to carry out the PSF correction itself, modeling of the PSF shape and its spatial variations across the instrument field of view has, so far, attracted much less attention. This step is nevertheless crucial because the PSF is only known at star positions while the correction has to be performed at any position on the sky. A reliable interpolation scheme is therefore mandatory and a popular approach has been to use low-order bivariate polynomials. In the present paper, we evaluate four other classical spatial interpolation methods based on splines (B-splines), inverse distance weighting (IDW), radial basis functions (RBF) and ordinary Kriging (OK). These methods are tested on the Star-challenge part of the GRavitational lEnsing Accuracy Testing 2010 (GREAT10) simulated data and are compared with the classical polynomial fitting (Polyfit). In all our methods we model the PSF using a single Moffat profile and we interpolate the fitted parameters at a set of required positions. This allowed us to win the Star-challenge of GREAT10, with the B-splines method. However, we also test all our interpolation methods independently of the way the PSF is modeled, by interpolating the GREAT10 star fields themselves (i.e., the PSF parameters are known exactly at star positions). We find in that case RBF to be the clear winner, closely followed by the other local methods, IDW and OK. The global methods, Polyfit and B-splines, are largely behind, especially in fields with (ground-based) turbulent PSFs. In fields with non-turbulent PSFs, all interpolators reach a variance on PSF systematics $\sigma^2_{\rm sys}$ better than the 1×10^{-7} upper bound expected by future space-based surveys, with the local interpolators performing better than the global ones.

Key words. gravitational lensing: weak – methods: data analysis

1. Introduction

The convolution of galaxy images with a point spread function (PSF) is among the primary sources of systematic error in weak lensing measurement. The isotropic part of the PSF kernel makes the galaxy shape appear rounder, while the anisotropic part introduces an artificial shear effect that may be confused with the genuine shear lensing signal.

To tackle these issues, various PSF correction methods have been proposed (Kaiser et al. 1995; Luppino & Kaiser 1997; Hoekstra et al. 1998; Kaiser 2000; Bernstein & Jarvis 2002; Hirata & Seljak 2003; Refregier & Bacon 2003) and some of them implemented as part of shear measurement pipelines (Heymans et al. 2006; Massey et al. 2007; Bridle et al. 2010). However, these correction schemes do not have built-in solutions for addressing another problem: the spatial variation of the PSF across the instrument field of view that may arise, for instance, from imperfect telescope guidance, optical aberrations or atmospheric distortions.

A non-constant PSF field implies the PSF is no longer accurately known at galaxy positions and must then be estimated for the accurate shape measurement of galaxies. Bivariate polynomials, typically used as interpolators for this purpose, have in several cases been found unable to reproduce sparse, multi-scale or quickly varying PSF anisotropy patterns (Hoekstra 2004; Jarvis & Jain 2004; Van Waerbeke et al. 2002, 2005; Jee & Tyson 2011).

This raises the question of whether there exists alternative PSF models and interpolation schemes better suited for PSF estimation than those used so far. Indeed, it seems important to improve this particular aspect of PSF modeling in the perspective of future space-based missions such as Euclid or advanced ground-based telescopes like the LSST (Jee & Tyson 2011).

Only recently has the PSF variation problem begun to be taken seriously with, notably, the advent of the GRavitational lEnsing Accuracy Testing 2010 (GREAT10) Star Challenge, one of the two GREAT10 challenges (Kitching et al., in prep.). The Star Challenge images have been designed to simulate a variety of typical position-varying PSF anisotropy patterns and competing PSF interpolation methods were judged on their ability to reconstruct the true PSF field at asked, non-star positions.

The Star Challenge gave us the opportunity to evaluate a number of alternative schemes suitable for the interpolation of realistic, spatially-varying PSF fields. The objective of this paper is twofold: (1) to describe our approach for tackling the problems raised by the Star Challenge and to discuss our results; (2) to perform a comparative analysis of the different interpolation methods after applying them on the Star Challenge simulations.

Our paper is thus organized as follows. We begin by reviewing the most commonly used PSF representation and interpolation schemes in Sect. 2 and continue with a overview of the interpolation schemes mentioned above in Sect. 3. We then describe our PSF estimation pipeline and analyze our results in Sects. 4 and 5. Lastly, in Sect. 6, we measure the respective accuracy of

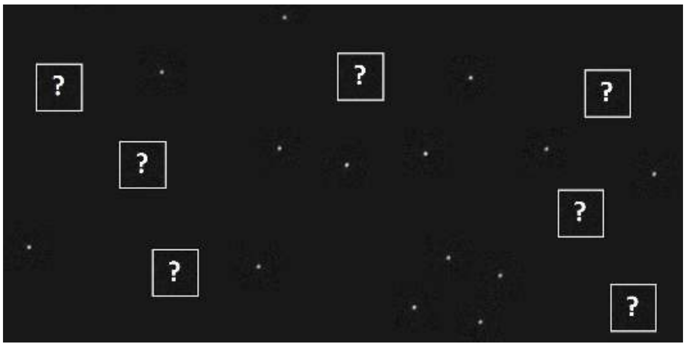

Fig. 1. Interpolating a spatially-varying PSF field. The illustrated field is a subset of an actual GREAT10 Star Challenge PSF field.

all methods based on the solutions made available after completion of the challenge and discuss the merits of each method. We conclude in Sect. 7.

2. An overview of existing PSF interpolation schemes

Before correcting galaxies in the image for a spatially-varying PSF field, every shear measurement pipeline has, in one way or another, to interpolate the PSF between the stars, as illustrated in Fig. 1. The way this is best achieved depends essentially on the PSF model used and on the PSF interpolation algorithm. The PSF model defines which features of the PSF are to be represented, which also determines on which quantities spatial interpolation is performed. The role of the interpolation scheme, on the other hand, is to apply a prediction algorithm to find the best estimates for those quantities.

In the KSB method Kaiser et al. (1995) and its KSB+ variant (Luppino & Kaiser 1997; Hoekstra et al. 1998), the relevant features of the PSF model are its ellipticity and size, which are estimated from the second-order geometrical moments of the PSF image. The main idea behind the PSF correction scheme is that the PSF distortion on a galaxy image can be well described by a small but highly anisotropic kernel q, convolved with a large, circular seeing disk. To find the appropriate q for galaxies, the values of q^* at star positions (and sometimes the so-called “smear” and “shear” polarization tensors $P^{\mathrm{sm}*}$ and $P^{\mathrm{sh}*}$) are interpolated across the image. For doing so, the typical procedure is to fit a second or third-order bivariate polynomial function.

Exactly which quantity is interpolated and which order is used for the polynomial depends on the KSB+ implementations. See e.g. Heymans et al. (2006, Appendix A), Massey et al. (2007) and recently published studies using KSB+ (Hoekstra et al. 1998; Clowe & Schneider 2002; Heymans et al. 2005; Hetterscheidt et al. 2007; Paulin-Henriksson et al. 2007; Fu et al. 2008; Umetsu et al. 2010).

A model representing a PSF as only a size and first-order deviation from circularity certainly appears quite restrictive. One can instead look for an extensive, but compact description of the PSF image, better suited to operations like noise filtering or deconvolution. A natural approach is to characterize the full PSF as a compact, complete set of orthogonal basis functions provided in analytical form, each basis being associated with a particular feature of the image (shape, frequency range, etc.). Ideally, this would not only simplify galaxy deconvolution from the PSF but also allow to better model the spatial variation of the PSF across the field of view.

Bernstein & Jarvis (2002) and Refregier (2003); Refregier & Bacon (2003); Massey & Refregier (2005) have proposed PSF expansions based on the eigenfunctions of the two-dimensional quantum harmonic oscillator, expressed in terms of Gauss-Laguerre orthogonal polynomials (Abramowitz & Stegun 1965). These functions can be interpreted as perturbations around a circular or elliptical Gaussian. The effect of a given operation (such as shear or convolution), on an image can then be traced through its contribution on each coefficient in the basis function expansion. For instance, the second-order $f_{2,2}$ coefficient of a Shapelet is the ellipticity estimator based on the Gaussian-weighted quadrupole moments used in KSB.

Modeling the PSF variation patterns with Shapelets typically involves the following steps: stars are expanded in terms of Shapelet basis functions and the expansion coefficients for each of the basis functions are fitted with a third or fourth-order polynomial. The interpolated values of the Shape let coefficients are then used to reconstruct the PSF at galaxy positions.

This scheme has been successfully applied to several weak lensing cluster studies (Jee et al. 2005a,b, 2006, 2007b; Bergé et al. 2008; Romano et al. 2010). However, it has been argued (Jee et al. 2007a; Melchior et al. 2010) that even a high-order Shapelet-based PSF model is unable to reproduce extended PSF features (such as its wings) and that the flexibility of the model makes it vulnerable to pixelation and noise. So, although the level of residual errors after Shapelets decomposition appears low enough for cluster analysis, it may prove too high for precision cosmic shear measurement.

Actually, it is not clear if there exists any set of basis functions expressed in analytical form that is capable of accurately describing all the signal frequencies contained in the PSF. An alternative approach is to decompose the PSF in terms of basis functions directly derived from the data through Principal Component Analysis (PCA), as pioneered by Lauer (2002), Lupton et al. (2001). This approach is supposed to yield a set of basis function, the so-called “Principal Components”, optimized for a particular data configuration and sorted according to how much they contribute to the description of the data.

In practice, two main procedures have been experimented that essentially depend on the type data where PCA is applied. Jarvis & Jain (2004) and Schrabback et al. (2010) fit selected components of the PSF (e.g. ellipticity or KSB anisotropy kernel) across all image exposures with a two-dimensional polynomial of order 3 or 4. PCA analysis is performed on the coefficients of the polynomial, which allows the large-scale variations of the PSF in each exposure to be expressed as a weighted sum of a small set of principal components. A further, higher-order polynomial fit is then conducted on each exposure to capture more detailed features of the PSF.

On the other hand, and more recently, Jee et al. (2007a), Nakajima et al. (2009) and Jee & Tyson (2011) experimented a different procedure for modeling the variation of the *Hubble* Space telescope (HST) ACS camera and a simulated Large Synoptic Survey Telescope (LSST) PSF. Instead of applying PCA on polynomial coefficients, they perform a PCA decomposition on the star images themselves into a basis made of the most discriminating star building blocks. Each star can then be expanded in terms of these “eigenPSFs” and the spatial variation of their coefficients in that basis is modeled with a bivariate polynomial.

Regardless of the procedure used, the PCA scheme proves superior to wavelets and Shapelet for reproducing smaller-scale features in the PSF variation pattern, thanks to improved PSF modeling and the use of higher-order polynomials. In the case of Jarvis & Jain (2004), applying PCA across all exposures allowed to compensate for the small number of stars available per exposure. Moreover, PCA is not tied to any specific PSF model.

It should be noted, however, that at least two factors may limit the performance of PCA in practical weak lensing applications: the first is that the PCA algorithm is only able to capture linear relationships in the data and thus may fail to reproduce some types of high-frequency variation patterns; the other is that PCA misses the components of the PSF pattern that are random and uncorrelated, such as those arising from atmospheric turbulence. How serious these limitations prove to be and how they can be overcome need to be investigated further (e.g. Jain et al. 2006; Schrabback et al. 2010).

All the above methods attempt to model PSF variation patterns in an empirical way by the application of some mathematical formalism. It may, on the contrary be more beneficial to understand which physical phenomena determine the structure of the PSF patterns and, once done, seek appropriates models for reproducing them (Jee et al. 2007a; Stabenau et al. 2007; Jee & Tyson 2011). The PSF of the HST ACS camera, for instance, has been studied extensively and in some cases, the physical origin of some of the patterns clearly identified. Jee et al. (2007a) and Jee & Tyson (2011) could relate the primary principal component to changes in telescope focus causes by constraints on the secondary mirror supporting structure and the "thermal breathing" of the telescope.

In fact, various combined effects make the PSF vary spatially or over time. Some patterns are linked to the behavior of the optical system of the telescope or the detectors. Others are related to mechanical or thermal effects that make the telescope move slightly during an observation. For ground-based instruments, refraction in the atmosphere and turbulence induce further PSF distortion.

Incorporating such a wide diversity of effects into a PSF variation model is not an easy task. However, according to Jarvis et al. (2008), models of low-order optical aberrations such as focus and astigmatism can reproduce 90% of the PSF anisotropy patterns found in real observation data. If so, physically-motivated models could provide an alternative or a complement to purely empirical methods such as PCA.

3. Looking for better PSF interpolation schemes

The analysis of commonly used PSF interpolation schemes in the previous section has shown that the range of PSF interpolation algorithms is actually quite restricted: almost always the quantities used to characterize the PSF are fitted using a bivariate polynomial function.

But it is important to acknowledge there may exist alternative interpolation schemes that would prove more effective for that purpose than polynomial fitting. Beyond this, it is essential to recognize the goal here is not to only interpolate changes in the PSF but also to perform a *spatial* interpolation of such changes.

Interpolation (e.g. Press et al. 2007) is commonly understood as the process of estimating of values at location where no sample is available, based on values measured at sample locations. *Spatial interpolation* differs from regular interpolation in that it can take into account and potentially exploit spatial relationships in the data. In particular, it is often the case that points close together in space are more likely to be similar than points further apart. In other words, points may be spatially autocorrelated, at least locally. Most spatial interpolation methods attempt to make use of such information to improve their estimates.

After a critical review of polynomial fitting, we consider and discuss alternative spatial interpolation schemes for modeling PSF variation patterns.

3.1. A critical view of polynomial fitting

In the context of spatial interpolation, fitting polynomial functions of the spatial coordinate $\boldsymbol{x} = (x_i, y_i)$ to the sampled $z(\boldsymbol{x})$ values of interest by ordinary least squares regression (OLS) is known as "Trend Surface Analysis" (TSA). The fitting process thus consists in minimizing the sum of squares for $(\hat{z}(\boldsymbol{x}) - z(\boldsymbol{x}))$, assuming the data can be modeled as a surface of the form

$$\hat{z}(\boldsymbol{x}) = \sum_{r+s \leq p} b_{rs}\, x^r\, y^s \tag{1}$$

The integer p is the order of the trend surface (and the order of the underlying polynomial). Finding the b_i coefficients is a standard problem in multiple regression and can be computed with standard statistical packages.

In the literature reviewed from the previous section, authors often justify their choice of polynomial fitting by arguing the PSF varies in a smooth manner over the image. Indeed trend surfaces are well suited to modeling broad features in the data with a smooth polynomial surface, commonly of order 2 or 3.

However, PSF variation patterns result from a variety of physical effects and even though polynomials may adequately reproduce the smoothest variations, there may exist several other types of patterns that a low-order polynomial function cannot capture. Polynomials are also quite poor at handling discontinuities or abrupt changes in the data. This concerns particularly sharp discontinuities across chip gaps and rapid changes often found near the corners of the image.

An illustrative example of the shortcomings just described was the detection of a suspicious non-zero B-mode cosmic shear signal in the VIRMOS-DESCART survey (Van Waerbeke et al. 2001, 2002). After investigation (Hoekstra 2004; Van Waerbeke et al. 2005), the small scale component of the signal was traced to the PSF interpolation scheme: the second-order polynomial function was unable to reproduce the rapid change in PSF anisotropy at the edges of the images. In fact, one of the main limitation of polynomials when used for interpolating PSF images in weak lensing studies lie in their inability to reproduce variations on scales smaller than the typical inter-stellar distance on the plane of the sky (often $\lesssim$1 arcmin at high galactic latitude).

Unfortunately there are no satisfactory solutions to these shortcomings. Increasing the order of the polynomial function does not help as it leads to oscillations while attempting to capture smaller-scale or rapidly-varying features. The $z(\boldsymbol{x})$ values may reach extremely (and unnaturally) small or large values near the edge or just outside the area covered by the data. Such extreme values can also create problems in calculations.

One way to alleviate such problems is to pack more densely the interpolating points closer to the boundaries, but this may not be easy to achieve in practice. Hoekstra (2004) and Van Waerbeke et al. (2005) also obtained good results with an interpolator made of a polynomial function to model large-scale changes combined with a rational function to deal with small-scale variations. It is not clear, however, if this scheme can be safely applied on different data and this may require a significant amount of fine tuning.

In addition to the issues just mentioned, local effects in one part of the image may influence the fit of the whole regression

Table 1. Least squares polynomial fitting/trend surface: Pros and cons.

Least squares polynomial fitting	
Pros	Simple and intuitive Fast to compute
Cons	Usually only able to capture broad features (underfitting) Increasing the order of polynomials does not improve and generally degrades accuracy (overfitting) High-order polynomials generate numerical issues (rounding errors, overflow, etc.) High sensitivity to outliers and fitting errors Local changes propagate to the whole polynomial surface No estimation of interpolation errors (deterministic)

Table 2. Spatial interpolation methods reviewed in this article.

Interpolation method	Scope	Exactness	Model
Polynomial fitting	global	approximate	deterministic
Basis splines	global	approximate[1]	deterministic
Inverse distance weighting	local	exact[2]	deterministic
Radial basis function	local	exact[3]	deterministic
Ordinary Kriging	local	exact[4]	stochastic

Notes. [1] Can be made exact by disabling smoothing. [2] Smoothing possible with specific algorithms. [3] Some Kriging algorithms are approximate.

surface, which makes trend surfaces very sensitive to outliers, clustering effects or observable errors in the $z(\boldsymbol{x})$. Finally, OLS regression implicitly assumes the $z(\boldsymbol{x})$ are normally distributed with uncorrelated residuals. These assumptions do not hold true when spatial dependence exists in the data.

Actually, the fact that trend surfaces tend to ignore any spatial correlation or small-scale variations can turn into an advantage to remove broad features of the data prior to using some other, finer-grained interpolator. Indeed, we saw in Sect. 1 that Jarvis & Jain (2004) took advantage of this feature in their PCA-based interpolation scheme.

Most of the aforementioned limitations are rooted in the use of standard polynomials. One possible way out is to abandon trend surfaces altogether and use piecewise polynomials instead (especially Chebyshev polynomials), splines (de Boor 1978; Dierckx 1995; Schumaker 2007; Prenter 2008) or alternative schemes that do not involve polynomials. Table 1 recalls the main advantages and disadvantages of polynomial fitting.

3.2. Toward alternative PSF interpolation methods

Having pointed out some important shortcomings of polynomial regression, it seems legitimate to look for alternative classes of interpolators. It is however, probably illusory to look for an ideal interpolation scheme that can describe equally well any kind of PSF variation structure. For instance the patterns of variation in a turbulent PSF are very different from those found in a diffraction-limited PSF. It is therefore probably more useful to identify which class of interpolators should be preferably used for a particular type of PSF pattern.

It is also key to realize that one does not need to reconstruct the entire PSF field: one only has to infer the PSF at specific galaxy positions based on its knowledge at sample star positions. This implies that the class of interpolation schemes applicable to the PSF variation problem is not restricted to surface fitting algorithms such as polynomial fitting, but also encompasses interpolation algorithms acting on scattered data.

Such data may also be considered as a partial realization of some stochastic process. In such case, it becomes possible to quantify the uncertainty associated with interpolated values and the corresponding interpolation method is referred to as a method for *spatial prediction*. In this article we will neglect this distinction and use the generic term “spatial interpolation”.

In fact, there are quite a few interpolation schemes that can be applied to model PSF changes. Over the years a large number of interpolation methods have been developed in many disciplines and with various objectives in mind. Spatial interpolators are usually classified according to their range (*local* versus *global*), the amount of smoothing (*exact* versus *approximate*) and whether they consider the data as a realization of some random process (*stochastic* versus *deterministic*).

A *global* method makes use of all available observations in the region of interest (e.g. the image of a whole portion of the sky) to derive the estimated value at the target point whereas a *local* method only considers observations found within some small neighborhood around the target point. Thus, global methods may be preferable to capture the general trend in the data, whereas local methods may better capture the local or short-range variations and exploit spatial relationships in the data (Burrough & McDonnell 1998). A trend surface is an example of global estimator.

An interpolation methods that produces an estimate that is the same as the observed value at a sampled point is called an *exact* method. On the contrary a method is *approximate* if its predicted value at the point differs from its known value: some amount of smoothing is involved for avoiding sharp peaks or troughs in the resulting fitted surface.

Lastly, a *stochastic* (also called *geostatistical*) interpolator incorporates the concept of randomness and yields both an estimated value (the deterministic part) and an associated error (the stochastic part, e.g. an estimated variance). On the other hand, a *deterministic* method does not provide any assessment of the error made on the interpolated value.

Table 2 contains the list of spatial interpolation methods covered in this article along with their classification.

Nearly all methods of spatial interpolation share the following general spatial prediction formula

$$\hat{z}(\boldsymbol{x}_0) = \sum_{i=1}^{N} \lambda_i \, z(\boldsymbol{x}_i) \tag{2}$$

where $\boldsymbol{x}_0$ is a target point where the value should be estimated, the $z(\boldsymbol{x}_i)$ are the locations where an observation is available and the λ_i are the weights assigned to individual observations. N represents the number of points involved in the estimation (see Fig. 2 for an illustration). Each interpolation method has its own algorithm for estimating the weights λ_i. All the interpolation methods evaluated in this article except splines, follow Eq. (2).

We now review several widely used interpolation schemes that can be applied to the PSF interpolation problem: polynomial splines, inverse distance weighting (IDW), radial basis functions (RBF) and Kriging. In the remaining sections, we test these interpolation methods using the GREAT10 Star Challenge simulated data.

3.3. Spline interpolation

A (polynomial) univariate *spline* or degree p (order $p+1$) is made of a set of polynomial pieces, joined together such that pieces

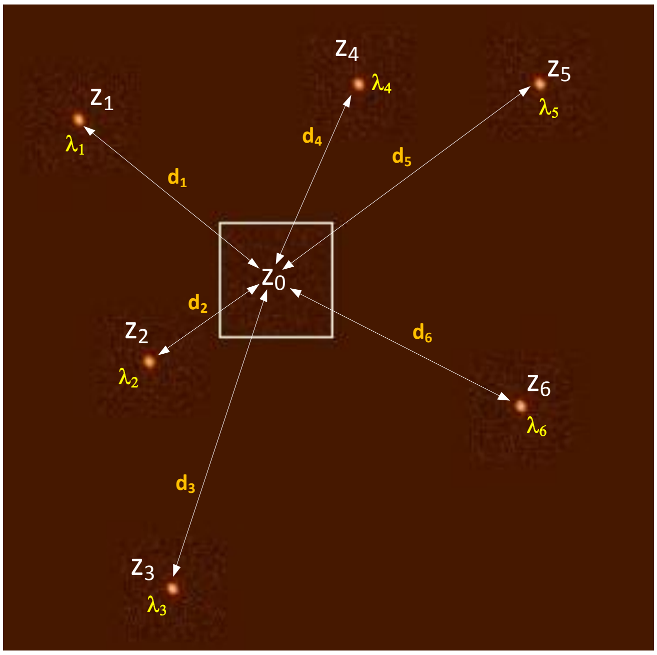

Fig. 2. An illustration of local interpolation between a set of neighboring observations $Z(\boldsymbol{x}_i)$ at distances d_i from a target location $\boldsymbol{x}_0$. In this example, a set of weights λ_i is assigned to each of the $Z(\boldsymbol{x}_i)$, as in Eq. (2).

and their derivatives at junction points (*knots*) are continuous up to degree $p-1$ (de Boor 1978; Dierckx 1995; Schumaker 2007; Prenter 2008).

When it comes to modeling two-dimensional spatially varying PSF attributes across an image, we are more specifically interested in bivariate polynomial splines. A function $s(x, y)$ defined on a domain $[a, b] \times [c, d]$ with respective, strictly increasing knot sequences λ_i, $i = 0, 1, ..., g+1$ ($\lambda_0 = a, \lambda_{g+1} = b$) in the x direction and μ_j, $j = 0, 1, ..., h+1$ ($\mu_0 = c, \mu_{h+1} = d$) in the y direction is called a bivariate (tensor product) spline function of degree $k > 0$ (order $k+1$) in x and $l > 0$ (order $l+1$) in y if the following two conditions are satisfied:

1. on each subregion $\mathcal{D}_{i,j} = [\lambda_i, \lambda_{i+1}] \times [\mu_j, \mu_{j+1}]$, $s(x, y)$ is given by a polynomial of degree k in x and l in y

   $$s(x, y) \in \mathcal{P}_k \otimes \mathcal{P}_l \quad i = 0, 1, ..., g;\ j = 0, 1, ..., h;$$

2. The function $s(x, y)$ and all its partial derivatives are continuous on $\mathcal{D}_{i,j}$

   $$\frac{\partial^{i+j} s(x, y)}{\partial x^i\, \partial y^j} \in C(\mathcal{D}_{i,j}) \quad i = 0, 1, ..., k-1;\ j = 0, 1, ..., l-1.$$

We saw earlier that polynomial fitting suffers in particular from two serious drawbacks. One of these is that individual observations can exert an influence, in unexpected ways, on different parts of the interpolating surface. The other is the tendency of the interpolation surface to wiggle without control as soon as one increases the degree of the polynomial to try to obtain a closer fit.

Polynomial splines solve these problems in two ways. First, a spline is not made of a single "global" polynomial but of a set of "local" polynomial pieces. This design confines the influence of individual observations within the area covered by the enclosing polynomial piece. In most applications, a specific type of spline is preferred, the so-called "Basis spline" (*B-spline*), built from as a linear combination of basis polynomial functions called *B-splines*

$$s(x, y) = \sum_{i=-k}^{g} \sum_{j=-l}^{h} c_{i,j}\, N_{i,k+1}(x)\, M_{j,l+1}(y)$$

where $N_{i,k+1}(x)$ and $M_{j,l+1}(y)$ are B-splines defined on the λ and μ knot sequences respectively. B-splines are popular for their computational efficiency, e.g. with the algorithms of Cox (Cox 1972) or de Boor (de Boor 1972). For a formal definition of the B-spline basis see e.g. de Boor (1978); Dierckx (1995); Prenter (2008).

The second issue is solved by the ability to control the smoothness of the spline. The example of polynomial fitting shows that a good fit to the data is not the one and only goal in surface fitting; another, and conflicting, goal is to obtain an estimate that does not display spurious fluctuations. A successful interpolation is, actually, a tradeoff between goodness of fit (fidelity to the data) and roughness (or "wiggleness") of fit: a good balance between these two criteria will allow the approximation to not pick up too much noise (overfitting) while avoiding signal loss (underfitting).

There is an extensive literature on spline interpolation and many algorithms and variants have been developed since their invention in the 1960s. Still, one can divide spline interpolation algorithms into two main families: those based on the so-called *constructive* approach, where the form of the spline function is specified in advance and the estimation problem is reduced to the determination of a discrete set of parameters; and those that follow a *variational* approach, where the approximation function is not known a priori, but follows from the solution of a variational problem, which can often be interpreted as the minimization of potential energy. We outline both approaches below.

Variational approach of spline interpolation

The variational approach (Wahba 1990; Green & Silverman 1994) consists in minimizing the functional

$$S(f, \alpha) = \sum_{i=1}^{N} \|z(\boldsymbol{s}_i) - f(\boldsymbol{s}_i)\|^2 + \alpha \int_{\mathcal{D}} \{f^{(m)}\}^2 \mathrm{d}\boldsymbol{s}_i \tag{3}$$

where the bivariate spline function f is fitted to the $z(\boldsymbol{s}_i), i = 0, ..., N$ set of points in the region $\mathcal{D}$ where the approximation is to be made. It can be shown (e.g. Green & Silverman 1994) that the solution is a natural spline, that is, a spline whose second and third derivatives are zero at the boundaries. splines obtained in such a way as known in the literature as *smoothing splines*. The parameter m represents the order of the derivative of f and $\alpha \geq 0$ is a smoothing parameter controlling the tradeoff between fidelity to the data and roughness of the spline approximation.

1. As $\alpha \longrightarrow 0$ (no smoothing), the left-hand side least squares estimate term dominates the roughness term on the right-hand side and the spline function attempts to match every single observation point (oscillating as required).
2. As $\alpha \longrightarrow \infty$ (infinite smoothing), the roughness penalty term on the right-hand side becomes paramount and the estimate converges to a least squares estimate at the risk of underfitting the data.

The most popular variational spline interpolation scheme is that based on the *thin-plate spline* (TPS; Duchon 1976; Meinguet 1979; Wahba & Wendelberger 1980; Wahba 1990; Hutchinson 1995). The TPS interpolating spline is obtained by minimizing an energy function of the form (3)

$$S(f, \alpha) = \sum_{i=1}^{N} \|z(\boldsymbol{s}_i) - f(\boldsymbol{s}_i)\|^2 + \alpha \, J_m(g) \, \mathrm{d}\boldsymbol{s}_i \tag{4}$$

The most common choice of m is 2 with J_2 of the form

$$J_2(g) = \int_{\mathcal{D}} \left\{ \left(\frac{\partial^2 g}{\partial x^2} \right)^2 + 2 \left(\frac{\partial^2 g}{\partial x \, \partial y} \right)^2 + \left(\frac{\partial^2 g}{\partial y^2} \right)^2 \right\} \mathrm{d}x \, \mathrm{d}y \tag{5}$$

where the roughness function $g(x, y)$ is given by

$$g(\boldsymbol{s}) = a_0 + a_1 x + a_2 y + \sum_{1}^{N} \lambda_i \, \phi(\boldsymbol{s} - \boldsymbol{s}_i) \tag{6}$$

ϕ being the RBF: $\Phi(x, y) = d_i^2 \ln(d_i)$ with Euclidean distance $d_i = \sqrt{(x - x_i)^2 + (y - y_i)^2}$. The λ_i are weighting factors.

Constructive approach of spline interpolation

Interpolating splines obtained through such a scheme are often referred to as *least squares splines* (Dierckx 1980, 1995; Hayes & Halliday 1994). For such splines, goodness of fit is measured through a least squares criterion, as in the variational approach, but smoothing is implemented in a different way: in the variational solution, the number and positions of knots are not varied, and the approximating spline is obtained by minimizing an energy function. On the other hand, in the constructive approach, one still tries to find the smoothest spline that is also closest to the observation points. But this is achieved by optimizing the number and placement of the knots and finding the corresponding coefficients $\boldsymbol{c}$ in the B-Spline basis. This is measured by a so-called *smoothing norm* $G(\boldsymbol{c})$. Thus, the approximating spline arises as the minimization of

$$S(f, \alpha) = \sum_{i=1}^{N} \|z(\boldsymbol{s}_i) - f(\boldsymbol{s}_i)\|^2 + \alpha \, G(\boldsymbol{c}) \tag{7}$$

using the same notation as in (3). An example of knot placement strategy is to increase the number of knots (i.e. reduce the inter-knot distance) in areas where the surface to fit varies faster or more abruptly. By the way, we note that minimization is not obtained by increasing the degree of the spline (which is kept low, typically 3).

Whatever the approach followed for obtaining a suitable interpolating spline, spline interpolation is essentially *global*, *approximate* and *deterministic*, as it involves all available observations points, makes use of smoothing and does not provide any estimation on interpolation errors. The interpolation can however be made exact by setting the smoothing coefficient to zero. Also, for smoothing splines (variational approach) a technique called *generalized cross-validation* (GCV; Craven & Wahba 1978; Wahba 1990; Hutchinson & Gessler 1994) allows to automatically choose, in expression (4), suitable parameters for α and m for minimizing cross-validation residuals. Otherwise, one can always use standard cross-validation or Jackknifing to optimize the choice of input parameters (see Sect. 4.3).

The most frequently used splines for interpolation are cubic splines, which are made of polynomials pieces of degree at most 3 that are twice continuously differentiable. Experience has shown that in most applications, using splines of higher degree seldom yields any advantage. As we saw earlier, splines avoid the pitfalls of polynomial fitting while being much more flexible, which allows them, despite their low degree, to capture finer-grained details. The method assumes the existence of measurement errors in the data and those can be handled by adjusting the amount of smoothing.

Table 3. Spline interpolation: Pros and cons.

Spline interpolation	
Pros	Able to capture both broad and detailed features The tradeoff between goodness and roughness of fit can be adjusted through smoothing
Cons	Overall smoothness may still be too high Keep a tendency to oscillate No estimation of interpolation errors in most algorithms Potentially less efficient to compute than local interpolation algorithms

On the minus side, cubic or higher degree splines are sometimes criticized for producing an interpolation that is "too smooth". They also keep a tendency to oscillate (although this can be controlled unlike with standard polynomials). In addition, the final spline estimate is influenced by the number and placement of knots, which confers some arbitrariness to the method, depending on the approach and algorithm used. This can be a problem since there is, in general, no built-in mechanism for quantifying interpolation errors. Lastly, spline interpolation is a global method and performance may suffer on large datasets. A summary of the main strengths and weaknesses of spline interpolation is given in Table 3.

3.4. Inverse distance weighting

Inverse distance weighting (IDW; Shepard 1968) is one of the oldest spatial interpolation method but also one of the most commonly used. The estimated value $\hat{z}(\boldsymbol{x}_0)$ at a target point $\boldsymbol{x}$ is given by Eq. (2) where the weights λ_i are of the form:

$$\lambda_i = \frac{1}{d^{\beta}(\boldsymbol{x}_0, \boldsymbol{x}_i)} \Big/ \sum_{i=1}^{N} \frac{1}{d^{\beta}(\boldsymbol{x}_0, \boldsymbol{x}_i)} \qquad \beta \geq 0 \qquad \sum_{i=1}^{N} \lambda_i = 1. \tag{8}$$

In the above expression, $d(\boldsymbol{x}_0, \boldsymbol{x}_i)$ is the distance between points $\boldsymbol{x}_0$ and $\boldsymbol{x}_i$, β is a power parameter and N is the number of points found in some neighborhood around the target point $\boldsymbol{x}_0$. Scaling the weights λ_i so that they sum to unity ensures the estimation is unbiased.

The rationale behind this formula is that data points near the target points carry a larger weight than those further away. The weighting power β determines how fast the weights tend to zero as the distance $d(\boldsymbol{x}_0, \boldsymbol{x}_i)$ increases. That is, as β is increased, the predictions become more similar to the closest observations and peaks in the interpolation surface becomes sharper. In this sense, the β parameter controls the degree of smoothing desired in the interpolation.

Power parameters between 1 and 4 are typically chosen and the most popular choice is $\beta = 2$, which gives the inverse-distance-squared interpolator. IDW is referred to as "moving average" when $\beta = 0$ and "linear interpolation" when $\beta = 1$.

For a more detailed discussion on the effect of the power parameter β, see e.g. Laslett et al. (1987); Burrough (1988); Brus et al. (1996); Collins & Bolstad (1996). Another way to control the smoothness of the interpolation is to vary the size of the neighborhood: increasing N yields greater smoothing.

Table 4. Inverse distance weighting: Pros and cons.

Inverse distance weighting	
Pros	Simple and intuitive Fast to compute
Cons	Choice of interpolation parameters empirical The interpolation is always exact (no smoothing) Sensitivity to outliers and sampling configuration (clustered and isolated points) No estimation of interpolation errors (deterministic)

IDW is a *local* interpolation technique because the estimation at $\boldsymbol{x}_0$ is based solely on observations points located in the neighboring region around $\boldsymbol{x}_0$ and because the influence of points further away decreases rapidly for $\beta > 0$. It is also forced to be *exact* by design since the expression for λ_i in Eq. (8) reaches the indeterminate form ∞/∞ when the estimation takes place at the point $\boldsymbol{x}_0$ itself. IDW is further labeled as *deterministic* because the estimation algorithm relies purely on geometry (distances) and does not provide any estimate on the error made.

IDW is popular for its simplicity, computational speed and ability to work on scattered data. The method also has a number of drawbacks. One is that the choice of the β parameter and the neighborhood size and shape are arbitrary, although techniques such as cross-validation or jackknifing can provide hints for tuning these parameters (see Sect. 4.3). Another is that there exists no underlying statistical model for measuring uncertainty in the predictions. Further, the results of the method method are sensitive to outliers and influenced by the way observations have been sampled. In particular, the presence of clustering can bias the estimation since in such cases clustered points and isolated points at similar but opposite distances will carry about the same weights. A common feature of IDW-generated interpolation surfaces is the presence of spikes or pits around observation points since isolated points have a marked influence on the prediction in their vicinity.

The original Shepard algorithm has been enhanced by several authors to address some of the shortcomings listed above. See in particular Renka (1988), Tomczak (1998) and Lukaszyk (2004). One frequent extension consists in explicitly introducing a smoothing factor s into Eq. (8), which then becomes

$$\lambda_i = \frac{1}{(d(\boldsymbol{x}_0, \boldsymbol{x}_i) + s)^\beta} \Bigg/ \sum_{i=1}^{N} \frac{1}{(d(\boldsymbol{x}_0, \boldsymbol{x}_i) + s)^\beta} \tag{9}$$

with values of s typically chosen between 1 and 5. Table 4 summarizes the main pros and cons of inverse distance weighting.

3.5. Interpolation with radial basis functions

We just described IDW, a simple form of interpolation on scattered data where the weighting power ascribed to a set neighboring point x_i from some point x only depends on an inverse squared distance function.

We now describe a similar, but more versatile form of interpolation where the distance function is more general and expressed in terms of a RBF (Buhmann 2003; Press et al. 2007). A RBF function, or *kernel* ϕ is a real-valued function where the value evaluated at some point $\boldsymbol{x}_0$ only depends on the radial distance between $\boldsymbol{x}_0$ and a set of points $\boldsymbol{x}_i$, so that $\phi(\boldsymbol{x}_0 - \boldsymbol{x}_i) = \phi(\|\boldsymbol{x}_0 - \boldsymbol{x}_i\|)$. The norm usually represents the Euclidean distance but other types of distance functions are also possible.

Table 5. Most popular RBF kernels.

RBF kernel $\phi(r)$	Expression
Multiquadric	$\sqrt{1 + (\epsilon\, r)^2}$
Inverse multiquadric	$1/[1 + (\epsilon\, r)^2]$
Gaussian	$\exp[-(\epsilon\, r)^2]$
Thin-plate	$r^2 \ln(r)$
Linear	r
Cubic	r^3

The idea behind RBF interpolation is to consider that the influence of each observation on its surrounding is the same in all direction and well described by a RBF kernel. The interpolated value at a point $\boldsymbol{x}_0$ is a weighted linear combination of RBF evaluated on points located within a given neighborhood of size N according to the expression

$$\hat{z}(\boldsymbol{x}_0) = \sum_{i=1}^{N} \lambda_i\, z(\boldsymbol{x}_i) = \sum_{i=1}^{N} \lambda_i\, \phi(\|(\boldsymbol{x}_0 - \boldsymbol{x}_i\|). \tag{10}$$

The weights are determined by imposing that the interpolation be exact at all neighboring points $\boldsymbol{x}_i$, which entails the resolution of a linear system of N equations with N unknown weighting factors λ_i. In some cases, it is necessary to add a low-degree polynomial $P_k(\boldsymbol{x})$ of degree k to account for a trend in $z(\boldsymbol{x})$ and ensure positive-definiteness of the solution. Expression (10) is then transformed into

$$\hat{z}(\boldsymbol{x}_0) = P_k(\boldsymbol{x}_0) + \sum_{i=1}^{N} \lambda_i\, \phi(\|(\boldsymbol{x}_0 - \boldsymbol{x}_i\|). \tag{11}$$

Sometimes, an interpolation scheme based on a *normalized RBF* (NRBF) of the form

$$\hat{z}(\boldsymbol{x}_0) = \sum_{i=1}^{N} \lambda_i\, \phi(\|(\boldsymbol{x}_0 - \boldsymbol{x}_i\|) \Bigg/ \sum_{i=1}^{N} \phi(\|(\boldsymbol{x}_0 - \boldsymbol{x}_i\|) \tag{12}$$

is preferred to (10), although no significant evidence for superior performance has been found.

The actual behavior and accuracy of RBF interpolation closely depends on how well the ϕ kernel matches the spatial distribution of the data. The most frequently used RBF kernels are listed in Table 5, where $r = \|\boldsymbol{x} - \boldsymbol{x}_i\|$ and the quantity ϵ is the so-called *shape parameter*. The required conditions for ϕ to be a suitable RBF kernel have been given by Micchelli (1986) but the choice of the most adequate kernel for a problem at hand is often empirical.

The shape parameter ϵ contained in the multiquadric, inverse multiquadric and Gaussian kernels influences the shape of the kernel function and controls the tradeoff between fitting accuracy and numerical stability. A small shape parameter produces the most accurate results, but is always associated with a poorly conditioned interpolation matrix. Despite the research work of e.g. Hardy (1990), Foley (1994) and Rippa (1999), finding the most suitable shape parameter is often a matter of trial and error. A rule of thumb is to set ϵ to approximately the mean distance to the nearest neighbor.

RBF interpolation based on the *multiquadric* (MQ) kernel $\sqrt{1 + (\epsilon\, r)^2}$ is the most common. It was first introduced by Hardy (1971) as a "superpositioning of quadric surfaces" for solving a problem in cartography. In its review of interpolation methods on

Table 6. Radial basis functions for interpolation: Pros and cons.

Radial basis functions	
Pros	Flexibility, thanks to various choice of kernel functions Relatively fast (local method), but computational speed depends on the kernel function
Cons	Choice of kernel functions and interpolation parameters empirical The interpolation is always exact (no smoothing) Sensitivity to outliers and sampling configuration (clustered and isolated points) No estimation of interpolation errors (deterministic)

scattered data, Franke (1982) highlighted the good performance of the MQ kernel, which has, since then proven highly successful in many disciplines (Hardy 1990).

RBF interpolation is fundamentally a *local*, *exact* and *deterministic* method. There are, however, algorithms that allow to introduce smoothing to better handle noise and measurement errors in the data. The method can prove highly accurate but this really depends on the affinity between the data and the kernel function used. Also, because predictions are exact, RBF functions can be locally sensitive to outliers. As for other deterministic methods like splines or IDW, the optimal set of parameters are most often determined by cross-validation or Jackknifing (see Sect. 4.3). Table 6 recapitulates the favorable and less favorable aspects of interpolation based on RBFs.

3.6. Kriging

Kriging is a spatial prediction technique initially created in the early 1950's by mining engineer Daniel G. Krige (Krige 1951) with the intent of improving ore reserve estimation in South Africa. But it was essentially the mathematician and geologist Georges Matheron who put Krige's work a firm theoretical basis and developed most of the modern Kriging formalism (Matheron 1962, 1963).

Following Matheron's work, the method has spread from mining to disciplines such as hydrology, meteorology or medicine, which triggered the creation of several Kriging variants. It is thus more accurate to refer to Kriging as a family of spatial prediction techniques instead of a single method. It is also essential to understand that Kriging constitutes a general method of interpolation that is in principle applicable to any discipline, such as astronomy.

The following textbooks provide a good introduction to the subject: Journel & Huijbregts (1978); Isaaks & Srivastava (1989); Cressie (1991); Deutsch & Journel (1997); Goovaerts (1997); Chilès & Delfiner (1999); Wackernagel (2003); Waller & Gotway (2004); Webster & Oliver (2007).

Like most of the local interpolation methods described so far in this article, Kriging makes use of the weighted sum (2) to estimate the value at a given location based on nearby observations. But instead of computing weights based on geometrical distances only, Kriging also takes into account the *spatial correlation* existing in the data. It does so by treating observed values $z(\boldsymbol{x})$ as *random variables* $Z(\boldsymbol{x})$ varying according to a spatial *random process*[1]. In fact, Kriging assumes the underlying process has a form of second-order stationarity called *intrinsic stationarity*. Second-order stationarity is traditionally defined as follows:

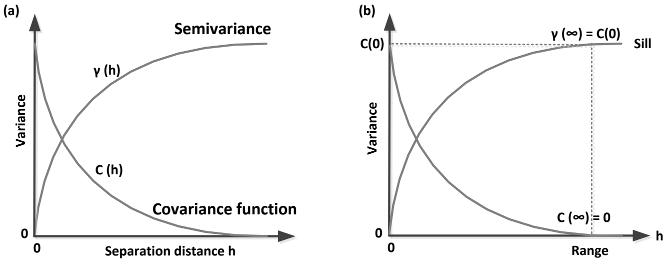

Fig. 3. a) Typical variogram $\gamma(\boldsymbol{h})$ and its equivalent covariance function $C(\boldsymbol{h})$: if the data has some sort of spatial autocorrelation, nearby (small h) $Z(\boldsymbol{x})$ observed values will be more similar than more distant $Z(\boldsymbol{x})$ values (larger h); **b)** as the separation distance h grows, the quantity $Z(\boldsymbol{x}+\boldsymbol{h}) - Z(\boldsymbol{x})$ in expression (16) will tend to increase on average, but less and less as the influence of $Z(\boldsymbol{h})$ on $Z(\boldsymbol{x}+\boldsymbol{h})$ weakens; at some threshold distance h, called the *range*, the increase in variance becomes negligible and the asymptotical variance value is known as the *sill*

1. The mathematical expectation $E(Z(\boldsymbol{x}))$ exists and does not depend on $\boldsymbol{x}$

$$E[Z(\boldsymbol{x})] = m, \quad \forall\, \boldsymbol{x}. \tag{13}$$

2. For each pair of random variable $\{Z(\boldsymbol{x}), Z(\boldsymbol{x}+\boldsymbol{h})\}$, the covariance exists and only depends on the separation vector $\boldsymbol{h} = \boldsymbol{x}_j - \boldsymbol{x}_i$,

$$C(\boldsymbol{h}) = E\{[Z(\boldsymbol{x}+\boldsymbol{h}) - m]\,[Z(\boldsymbol{x}) - m]\}, \quad \forall\, \boldsymbol{x}. \tag{14}$$

Kriging's *intrinsic stationarity* (Matheron 1963, 1965) is a slightly weaker form of second-order stationarity where the difference $Z(\boldsymbol{x}+\boldsymbol{h}) - Z(\boldsymbol{x})$ is treated as the stationary variable instead of $Z(\boldsymbol{x})$:

$$1.\ E[Z(\boldsymbol{x}+\boldsymbol{h}) - Z(\boldsymbol{x})] = 0, \quad \forall\, \boldsymbol{x} \tag{15}$$

$$2.\ Var[Z(\boldsymbol{x}+\boldsymbol{h}) - Z(\boldsymbol{x})] = E\{[Z(\boldsymbol{x}+\boldsymbol{h}) - Z(\boldsymbol{x})]^2\} = 2\gamma(\boldsymbol{h}). \tag{16}$$

The function $\gamma(\boldsymbol{h})$ is called *semivariance* and its graph *semivariogram* or simply *variogram*.

One reason for preferring intrinsic stationarity over secondary stationarity is that semivariance remains valid under a wider range of circumstances. When covariance exists, both stationarities are related through

$$\gamma(\boldsymbol{h}) = C(0) - C(\boldsymbol{h}), \qquad C(0) = Var[Z(\boldsymbol{x})], \tag{17}$$

Figure 3 shows a typical variogram along with its equivalent covariance function.

Over the years about a dozen Kriging variants have been developed. We will concentrate here on *ordinary Kriging* (OK), which is, by far, the most widely used. The description of other forms of Kriging can be found in the literature given at the beginning of this section.

ordinary Kriging is a *local*, *exact* and *stochastic* method. The set of $Z(\boldsymbol{x})$ is assumed to be an intrinsically stationary random process of the form

$$Z(\boldsymbol{x}) = m + \epsilon(\boldsymbol{x}). \tag{18}$$

The quantity $\epsilon(\boldsymbol{x})$ is a random component drawn from a probability distribution with mean zero and variogram $\gamma(\boldsymbol{h})$ given by (16). The mean $m = E[Z(\boldsymbol{x})]$ is assumed constant because

[1] Also called *random function* or *stochastic process*.

Table 7. Authorized Kriging theoretical variogram models.

Model	Expression	
Pure nugget	$\gamma(h) = 0$ $\gamma(h) = c_0 \qquad c_0 \geq 0$	$h = 0$ $h > 0$
Spherical	$\gamma(h) = c_0 + c\{\frac{3h}{2a} - \frac{1}{2}(\frac{h}{a})^3\}$ $\gamma(h) = c_0 + c$	$h \leq a$ $h > a$
Exponential	$\gamma(h) = c_0 + c\,\{1 - \exp\,(-\frac{h}{a})\}$	
Gaussian	$\gamma(h) = c_0 + c\,\{1 - \exp\,(-\frac{h^2}{a^2})\}$	
Power	$\gamma(h) = c_0 + b\,h^p$	$b \geq 0,\ 0 \leq p < 2$

Notes. The models in this table correspond to purely isotropic Kriging. More elaborate formulas exist for correcting geometrical anisotropy through the rescaling or rotation of coordinate axes along the direction of major spatial continuity. In the above expressions, $c_0 = \lim_{h \to 0} \gamma(h)$ is the so-called *nugget* constant that represents measurement errors or indicates a spatially discontinuous process. The quantities $(c_0 + c)$ and a respectively represent the variogram *sill* and *range*. The *pure nugget* model corresponds to absence of spatial correlation.

Table 8. Kriging interpolation: Pros and cons.

Kriging	
Pros	Predictions based on a spatial statistical analysis of the data Best linear unbiased estimator (BLUE) Many forms of Kriging available, applicable to various data configurations Automatically accounts for clustering and screening effects; remains efficient in conditions of sparse data Can take into account variation bias toward specific directions (anisotropy) Able to quantify interpolation errors (Kriging variance)
Cons	Overall complexity Requires care when modeling spatial correlation structures Assumptions of intrinsic stationarity may not be valid (drift) and be handled though an appropriate Kriging variant Most Kriging variants are exact (no smoothing) Kriging is more computationally intensive than other local methods

of (15), but remains *unknown*. The ordinary Kriging predictor is given by the weighted sum

$$\hat{Z}(\boldsymbol{x}_0) = \sum_{i=1}^{N} \lambda_i \, Z(\boldsymbol{x}_i) \tag{19}$$

where the weights λ_i are obtained by minimizing the so-called *Kriging variance*

$$\sigma^2(\boldsymbol{x}_0) = Var\left[\hat{Z}(\boldsymbol{x}_0) - Z(\boldsymbol{x}_0)\right] = E\left\{\left[\hat{Z}(\boldsymbol{x}_0) - Z(\boldsymbol{x}_0)\right]^2\right\} \tag{20}$$

subject to the unbiaseness condition

$$E[\hat{Z}(\boldsymbol{x}_0) - Z(\boldsymbol{x}_0)] = 0 = \sum_{i=1}^{N} \lambda_i \, E[z(\boldsymbol{x}_i)] - m. \tag{21}$$

The resulting system of $N + 1$ equations in $N + 1$ unknowns λ_i is known as the *ordinary Kriging equations*. It is often expressed in matrix form as $\boldsymbol{A\lambda} = \boldsymbol{b}$ with

$$\boldsymbol{A} = \begin{bmatrix} \gamma(\boldsymbol{x}_1, \boldsymbol{x}_1) & \gamma(\boldsymbol{x}_1, \boldsymbol{x}_2) & \cdots & \gamma(\boldsymbol{x}_1, \boldsymbol{x}_N) & 1 \\ \gamma(\boldsymbol{x}_2, \boldsymbol{x}_1) & \gamma(\boldsymbol{x}_2, \boldsymbol{x}_2) & \cdots & \gamma(\boldsymbol{x}_2, \boldsymbol{x}_N) & 1 \\ \vdots & \vdots & & \vdots & \vdots \\ \gamma(\boldsymbol{x}_N, \boldsymbol{x}_1) & \gamma(\boldsymbol{x}_N, \boldsymbol{x}_2) & \cdots & \gamma(\boldsymbol{x}_N, \boldsymbol{x}_N) & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{bmatrix} \tag{22}$$

$$\boldsymbol{\lambda}^{\mathrm{T}} = \begin{bmatrix} \lambda_1 & \lambda_2 & \cdots & \lambda_N & \mu \end{bmatrix} \qquad \sum_{i=1}^{N} \lambda_i = 1$$

$$\boldsymbol{b}^{\mathrm{T}} = \begin{bmatrix} \gamma(\boldsymbol{x}_1, \boldsymbol{x}_0) & \gamma(\boldsymbol{x}_2, \boldsymbol{x}_0) & \cdots & \gamma(\boldsymbol{x}_N, \boldsymbol{x}_0) & 1 \end{bmatrix}.$$

The weights λ_i, along with the Lagrange multiplier μ, are obtained by inversing the $\boldsymbol{A}$ matrix

$$\boldsymbol{\lambda} = \boldsymbol{A}^{-1}\boldsymbol{b}. \tag{23}$$

The main interpolation steps with ordinary Kriging can now be articulated:

1. Construct an experimental variogram by computing the experimental semivariance $\hat{\gamma}(\boldsymbol{h})$ for a range of separation distances $\|\boldsymbol{h}\|$.
2. Fit the experimental variogram against an authorized variogram model. The mathematical expressions for the most common authorized *theoretical variogram* models are summarized in Table 7. After completion of this step, the $\gamma(\boldsymbol{x}_i, \boldsymbol{x}_j)$ value at any separation vector $\boldsymbol{h} = \boldsymbol{x}_j - \boldsymbol{x}_i$ can be calculated and used to compute the $\boldsymbol{A}$ matrix (22).
3. Calculate interpolated values: derive the Kriging weights λ_i for each point of interest $\boldsymbol{x}_0$ by solving Eq. (23) and obtain the Kriging estimate at $\boldsymbol{x}_0$ by substituting in (19).

Most of the strengths of Kriging interpolation stem from the use of semivariance instead of pure geometrical distances. This feature allows Kriging to remain efficient in condition of sparse data and to be less affected by clustering and screening effects than other methods.

In addition, as a true stochastic method, Kriging interpolation provides a way of directly quantifying the uncertainty in its predictions in the form of the Kriging variance specified in Eq. (20).

The sophistication of Kriging, on the other hand, may also be considered as one of its disadvantages. A thorough preliminary analysis of the data is required or at least strongly recommended prior to applying the technique (e.g. Tukey 1977). This can prove complex and time consuming.

One should also bear in mind that Kriging is more computationally intensive than the other local interpolation methods described in this article. The strong and weaker points of Kriging interpolation are highlighted in Table 8.

4. Applying spatial interpolation schemes on the GREAT10 Star Challenge data

In 2011, we participated in the GREAT10 Star Challenge competition (Kitching et al., in prep.), which allowed us to evaluate the performance of the interpolation schemes described above: those based on splines, IDW, RBF and ordinary Kriging. To our knowledge, the only reference to a similar work in the field of astronomy is that of Bergé et al. (2012).

The GREAT10 Star Challenge ran from December 2010 to September 2011 as an open, blind competition. As illustrated in Fig. 4, the data consisted in 26 datasets of 50 PSF fields, each field containing between 500 and 2000 simulated star images and featuring specific patterns of variation. The stars

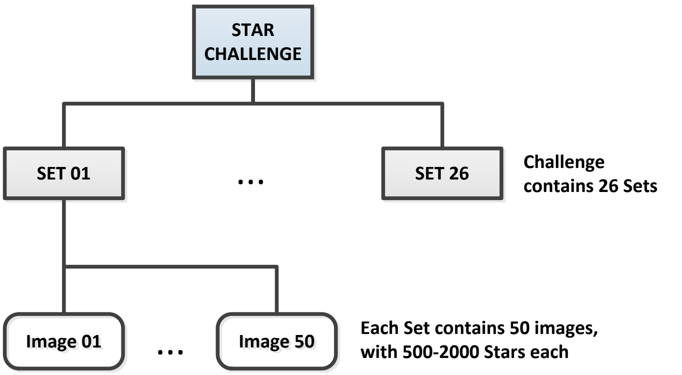
STAR CHALLENGE
SET 01
...
SET 26
Challenge contains 26 Sets
Image 01
...
Image 50
Each Set contains 50 images, with 500-2000 Stars each

Fig. 4. Star Challenge simulated data.

images were supplied as non-overlapping, randomly-scattered 48 × 48 pixels postage stamps, altered by Gaussian noise.

After completion of the challenge, it was revealed the stars had either a Moffat (Moffat 1969) or pseudo-Airy (Born & Wolf 1999; Kuijken 2008) profile, with a telescope component model from Jarvis et al. (2008). Depending on the sets, specific additional effects, such as Kolmogorov turbulence, were also incorporated.

The challenge itself was to predict the PSF at 1000 requested positions in each of the 1300 PSF fields (see Fig. 1).

4.1. Which model for the PSF?

The first important step to make was to choose an appropriate model for the PSF. Indeed, before selecting a particular PSF interpolator, one has to decide on which type of data that interpolator will operate.

Essentially three PSF modeling approaches have been explored in the literature:

1. PSF as a combination of basis functions;
2. PSF left in pixel form;
3. PSF expressed in functional form.

To help choosing the right model for the data at hand, useful guidance is provided by the notions of complexity and sparsity, recently put forward by (Paulin-Henriksson et al. 2008, 2009). The complexity of a model is characterized by the amount of information required to represent the underlying PSF image, which can be expressed as the number of degrees of freedom (DoF) present in the model. The more sophisticated the model the greater the number of its DoF. Sparsity, on the other hand, is meant to describe how efficiently a model can represent the actual PSF with a limited number of DoF, that is, with a simple model.

The simulated star images looked relatively simple and we decided that the right level of sparsity could be achieved with PSF in functional form (the third option). We then assumed that the most likely PSF profile used to create the stars was either Airy or Moffat. We opted for an elliptically symmetric Moffat function for its simplicity and because the stars did not show significant diffraction spikes. Each star was thus assumed to have a light intensity distribution of the form:

$$I(\xi) = I_0 \left[1 + \left(\frac{\xi}{\alpha}\right)^2\right]^{-\beta}, \quad \xi = \sqrt{(x' - x_c)^2 + \frac{(y' - y_c)^2}{q^2}}.$$

In the above expression, I_0 is the flux intensity at $\xi = 0$, ξ being the radius distance from the centroid (x_c, y_c) of the PSF to a spatial coordinate

$$\begin{bmatrix} x' - x_c \\ y' - y_c \end{bmatrix} = \begin{bmatrix} \cos\phi & \sin\phi \\ -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} x - x_c \\ y - y_c \end{bmatrix}, \tag{24}$$

obtained after counterclockwise rotation through an angle ϕ with respect to the $(0,\ x)$ axis. The quantity $\alpha = FWHM\,[2^{1/\beta} - 1]^{-1/2}$ is the Moffat scale factor expressed in terms of the full width at half maximum (FWHM) of the PSF and the Moffat shape parameter β. Lastly, q is the ratio of the semi-minor axis b to the semi-major axis a of the isophote ellipse, given by $q = b/a = (1 - |\boldsymbol{e}|)/(1 + |\boldsymbol{e}|)$, with $|\boldsymbol{e}| = \sqrt{{e_1}^2 + {e_2}^2}$, $e_1 = |\boldsymbol{e}| \cos 2\phi$ and $e_2 = |\boldsymbol{e}| \sin 2\phi$.

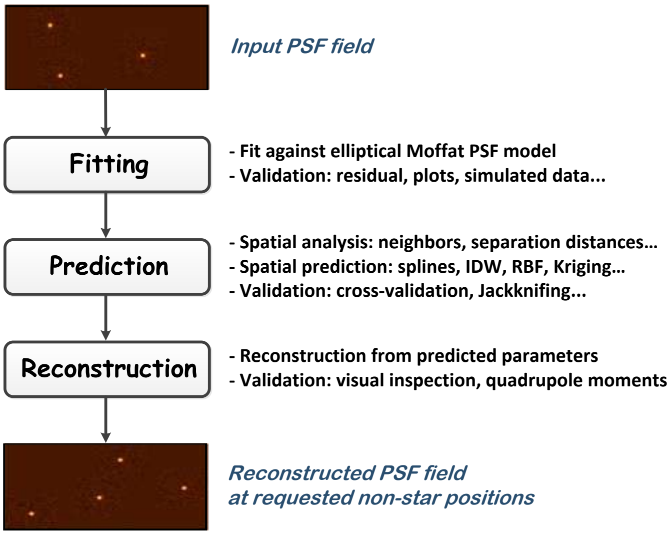
Input PSF field
Fitting
- Fit against elliptical Moffat PSF model
- Validation: residual, plots, simulated data...
Prediction
- Spatial analysis: neighbors, separation distances...
- Spatial prediction: splines, IDW, RBF, Kriging...
- Validation: cross-validation, Jackknifing...
Reconstruction
- Reconstruction from predicted parameters
- Validation: visual inspection, quadrupole moments
Reconstructed PSF field
at requested non-star positions

Fig. 5. The three-stage PSF prediction pipeline we used to compete in the Star Challenge. Elliptical Moffat profiles are fitted to the stars contained in the input Star Challenge PSF field; the model resulting parameters are then individually interpolated across the field at requested locations, using one of our PSF spatial interpolator. Lastly, the star images are reconstructed from the set of Moffat parameters predicted in the previous stage.

4.2. Our PSF prediction pipeline

The three-stage PSF prediction pipeline we used in the Star Challenge is sketched in Fig. 5. The purpose of the *fitting* stage is to produce a catalog of estimated FWHM and ellipticity values of the stars found at known spatial positions within the input Star Challenge PSF image.

In the *prediction* stage, that catalog is processed by an interpolation algorithm and a catalog is produced with estimated FWHM and ellipticities at new positions in the same image. Competitors were required to submit their results in the form of FITS Cube images (Kitching et al. 2011). In the *Reconstruction* stage, each star in a PSF field is thus reconstructed using that format from the interpolated quantities predicted in the *prediction* stage. A more detailed description of the pipeline is given in Appendix A.

4.3. Cross-validation and Jackknifing

The Star Challenge was a *blind* competition. The true answers being unknown, it was essential to find ways to evaluate how far the actual results were from the truth. To assess the fitting

Table 9. Common diagnostic statistics for use with cross-validation and Jackknifing.

Statistics	Expression
Mean error	$ME = \frac{1}{n}\sum_{i=1}^{n}\left[Z(\boldsymbol{x}_i) - Z(\boldsymbol{x}_{[i]})\right]$
Mean squared error	$MSE = \frac{1}{n}\sum_{i=1}^{n}\left[Z(\boldsymbol{x}_i) - Z(\boldsymbol{x}_{[i]})\right]^2$
Mean absolute error	$MAE = \frac{1}{n}\sum_{i=1}^{n}\left\lVert Z(\boldsymbol{x}_i) - Z(\boldsymbol{x}_{[i]})\right\rVert$
Mean squared deviation ratio	$MSDR = \frac{1}{n}\sum_{i=1}^{n}\left\{\left[Z(\boldsymbol{x}_i) - Z(\boldsymbol{x}_{[i]})\right]^2/\sigma^2_{[i]}\right\}$

accuracy in the first stage of the pipeline we could rely somewhat on the analysis of the residuals between observed and fitted star images. But when it came to evaluate prediction results, we had no such residuals to help us appraise the accuracy of the interpolation algorithm: we could only rely on the fitted observations $Z(\boldsymbol{x}_i)$. The use of cross-validation and Jackknifing provided a satisfactory solution to this problem.

Cross-validation

Cross-validation (CV) is a resampling technique frequently used in the fields of machine learning and data mining to evaluate and compare the performance of predictive models (Stone 1974; Geisser 1975; Isaaks & Srivastava 1989; Browne 2000).

In the context of the Star Challenge, we used CV to both evaluate the performance of an interpolation method and tune the free parameters of the underlying interpolation models.

As explained earlier, the deterministic interpolation methods (IDW, RBF, splines) we tested in the competition did not provide any quantification of residual errors. The first three diagnostic statistics mentioned in Table 9 provided a good indication of the level of accuracy reached. This technique was useful for Kriging as well because we could directly compare the mean error (ME) and mean squared error (MSE) provided by CV: Kriging being an unbiased estimator, we expected ME to be nearly zero, the MSE to be close to the Kriging variance provided by Eq. (20) and the mean squared deviation ratio (MSDR) to be around unity.

CV also proved useful for tuning the free parameters of the models behind the interpolation schemes, as mentioned in Appendix A.2. For instance, for RBF interpolation, we could rapidly try and discard the cubic, quintic, Gaussian and inverse multiquadric kernel functions. Another example was the ability to find the best search neighborhood size for local distance-based interpolation methods.

Jackknifing

The Jackknifing resampling technique was first proposed by Quenouille (1956) and further developed by Tukey (1958). A classical review on that subject is that of Miller (1974). See also Efron (1982); Efron & Gong (1983); Davis (1987); Tomczak (1998) for more general discussions on the use of CV in connection to Jackknifing.

To Jackknife a Star Challenge PSF field image, we would typically split the set of input coordinates into two equally-sized sets of star locations, i.e. 1000 randomly-selected star centroid positions from a set of 2000, one used for input and one used for prediction. We would then interpolate the PSF of the prediction set based on the PSF of the input set.

Table 10. Final results obtained by the B-SPLINE, IDW, Kriging and RBF methods in the Star Challenge, sorted by decreasing P-factors.

Rank	PSF interpolation method	P	$\sigma^2_{\rm sys}$
1	Basis spline (B-splines)	13.29	7.53×10^{-5}
2	Inverse distance weighting (IDW)	13.17	7.59×10^{-5}
3	Radial basis function (RBF)	12.72	7.86×10^{-5}
4	Radial basis function (RBF thin)	12.61	7.93×10^{-5}
5	Ordinary Kriging (OK)	7.23	1.38×10^{-4}

Notes. The B-splines method obtained the highest P-factor of the competition while the remaining four achieved the next highest scores.

5. Analyzing our GREAT10 Star Challenge results

5.1. Results on the Star Challenge data

The results obtained in the Star Challenge by the B-splines, IDW, Kriging, RBF and RBF-thin PSF interpolation schemes are shown in Table 10.

The B-splines method won the Star Challenge while the remaining four achieved the next highest scores of the competition.

The quantity P refers to the so-called *P-factor*, specified in Kitching et al. (in prep.). That P-factor is defined so as to measure the average variance over all images between the estimated and true values of two key PSF attributes: its size R and ellipticity modulus $e = |\boldsymbol{e}|$, estimated using second brightness moments computed over the reconstructed PSF images. Since the GREAT10 simulated star images have either Moffat or Airy profiles, R is actually an estimator of the FWHM of the stars.

The $\sigma^2_{\rm sys}$ quantity is related to the P-factor by $\sigma^2_{\rm sys} = 10^{-3}/P$ and represents a total residual variance in the measurement of the PSF. It approximates the corresponding metric specified in Amara & Réfrégier (2008); Paulin-Henriksson et al. (2008, 2009).

5.2. Performance metrics

In this article, we do not rely on the P-factor as a metric for assessing the performance of our methods, for the following reasons. Firstly, the P-factor is specific to the Star Challenge and is not mentioned anywhere else in the literature on PSF interpolation. Secondly, we are really interested in knowing the individual accuracy of ellipticity and size but P only appraises the combined performance of these quantities.

To assess the performance of an interpolator, we calculate instead the root mean squared error (RMSE) and standard error on the mean (SEM) of the residuals between true and calculated values of PSF ellipticity and size. As in Paulin-Henriksson et al. (2008); Kitching et al. (in prep.), we adopt the ellipticity modulus $e = (e_1^2 + e_2^2)^{1/2}$ and size squared R^2 as respective measures of ellipticity and size, and define the corresponding residuals as

$$\delta(e) = e_{\rm calc} - e_{\rm true}, \qquad \delta(R^2) = R^2_{\rm calc} - R^2_{\rm true}.$$

As regards PSF ellipticity, we adopt as performance metrics

$$E(e) = RMSE(\delta(e)/2), \quad \sigma(e) = SEM(\delta(e)/2)$$

while for PSF size, we evaluate

$$E(R^2) = RMSE(\delta(R^2))/\langle R^2_{\rm true}\rangle, \quad \sigma(R^2) = SEM(\delta(R^2))/\langle R^2_{\rm true}\rangle$$

Table 11. Average values of the performance metrics E and σ (see Sect. 5.2) over all sets, obtained by the B-SPLINE, IDW, Kriging and RBF methods in the Star Challenge.

Method	$E(e)$	$\sigma(e)$	$E(R^2)$	$\sigma(R^2)$
B-splines	2.03×10^{-2}	8.57×10^{-4}	1.90×10^{-1}	8.25×10^{-4}
IDW	2.04×10^{-2}	8.69×10^{-4}	1.92×10^{-1}	1.07×10^{-3}
RBF	2.26×10^{-2}	9.73×10^{-4}	1.98×10^{-1}	1.39×10^{-3}
Kriging	3.17×10^{-2}	1.26×10^{-3}	2.22×10^{-1}	2.18×10^{-3}

Table 12. Performance metrics used in this article.

PSF attribute	Metrics
PSF ellipticity	$E(e) = \sqrt{\langle(e_{\text{calc}} - e_{\text{true}})^2\rangle}/2$ $\sigma(e) = stdev\,(e_{\text{calc}} - e_{\text{true}})/\sqrt{2}/\sqrt{N}$
PSF size	$E(R^2) = \sqrt{\left\langle(R^2_{\text{calc}} - R^2_{\text{true}})^2\right\rangle}/\langle R^2_{\text{true}}\rangle$ $\sigma(R^2) = stdev\,(R^2_{\text{calc}} - R^2_{\text{true}})/\langle R^2_{\text{true}}\rangle/\sqrt{N}$

Notes. The angle brackets ⟨ and ⟩ denote averages and "*stdev*" the standard deviation. These statistics are calculated over the $N = 1000$ stars in each of the 50 images of each set.

where the angle brackets ⟨ and ⟩ denote averaging. The factor 2 in the expressions of $E(e)$ and $\sigma(e)$ arises because ellipticity has two components. We calculate these metrics over the $N = 1000$ stars in each of the 50 images of each set.

The quantity E provides a measure of the global accuracy of the interpolator (bias and precision combined) while σ provides insights into the variance of the residuals. The exact expressions for these performance metrics are given in Table 12.

5.3. Analysis of the star challenge results

The performance metrics of B-splines, IDW, RBF and Kriging are given in Table 11. The results of RBF and RBF-thin being very close, we no longer distinguish these two interpolators in the reminder of this paper and only mention them collectively as RBF.

Since a detailed analysis of the Star Challenge results of B-splines, IDW, RBF and Kriging as already been performed in Kitching et al. (in prep.), a similar analysis would be redundant here. We do have, however, a couple of observations to make, based on the metrics in Tables 10 and 11.

We observe that the global σ^2_{sys} variance of the most successful interpolation method is on the order of 10^{-4}. As demonstrated in Amara & Réfrégier (2008); Paulin-Henriksson et al. (2008) and confirmed by Kitching et al. (2009), future large surveys will need to constrain the total variance in the systematic errors to $\sigma^2_{\text{sys}} < 10^{-7}$, which corresponds to $E(e) \lesssim 10^{-3}$ and $E(R^2) \lesssim 10^{-3}$. The Star Challenge results thus tend to suggest that a ~10 improvement in $E(e)$ and a ~100 improvement in $E(R^2)$ are still required for achieving that goal.

Secondly, since we have been using a three-stage pipeline as described in Sect. 4.2, each stage, fitting, interpolation and reconstruction, can potentially contribute to the final error in size and ellipticity. Investigations following the publication of the true size and ellipticity values after the end of the Star Challenge, have led us to conclude fitting was actually the main performance limiting factor, not the interpolation or reconstruction process.

Table 13. Average values of the performance metrics E and σ (see Sect. 5.2) over all sets, based on the true input ellipticities and sizes.

Method	$E(e)$	$\sigma(e)$	$E(R^2)$	$\sigma(R^2)$
RBF	1.73×10^{-2}	7.18×10^{-4}	4.58×10^{-3}	1.44×10^{-4}
IDW	1.78×10^{-2}	7.24×10^{-4}	9.25×10^{-3}	2.91×10^{-4}
Kriging	1.82×10^{-2}	7.09×10^{-4}	6.47×10^{-3}	2.03×10^{-4}
Polyfit	2.29×10^{-2}	7.52×10^{-4}	5.16×10^{-3}	1.62×10^{-4}
B-splines	2.33×10^{-2}	7.39×10^{-4}	6.45×10^{-3}	2.04×10^{-4}

Also, the comparatively lower performance of Kriging is not related to the interpolation algorithm itself, but is actually due to an inadequate fitting setup, that was subsequently fixed for B-splines, IDW and RBF submissions.

As the main goal of this article is to assess the respective merits of the interpolation methods, we wish to eliminate all inaccuracies related to fitting. To achieve this, we use instead of our fitted ellipticity and FWHM estimates at known positions, the *true* input values, kindly supplied to us by the GREAT10 team. We interpolate these true input values at the expected target positions and then measure the error made by the interpolators. We present and analyze the corresponding results in the next section.

6. Comparing PSF spatial interpolation schemes

The results presented in this section are based on true FWHM and ellipticity values at known positions in the Star Challenge PSF images. We are thus confident that error statistics we obtained truly reflect the performance of the PSF interpolation methods and are not influenced in any way by inaccuracies due to the fitting of our PSF model or to the image reconstruction processes.

We compare below the respective performance of five PSF spatial interpolation schemes:

- The four interpolation schemes introduced in Sect. 3 that competed in the Star Challenge: B-splines, IDW, RBF and ordinary Kriging.
- An additional scheme, labeled *Polyfit*, which corresponds to a least-squares bivariate polynomial fit of the PSF, similar to that typically used in weak lensing studies (see Sect. 3.1).

The metric values reflecting the average accuracy E and error on the mean σ for these five interpolation schemes are given in Table 13.

6.1. Overall performance

The E and σ metrics on ellipticity and size after interpolation with all five methods are given in Table 13. These results lead to the following observations:

- If we compare Tables 13 and 12 we observe a ~100-fold decrease of $E(R^2)$ for all interpolators. This confirms that the fitting of PSF sizes was the main limitation that prevented us from reaching better results in the Star Challenge. In comparison, the fitting of ellipticities was quite good.
- If we now concentrate on Table 13, we find that the RBF interpolation scheme based on the use of radial basis functions,

has the highest accuracy and smallest error of the mean, both on size and ellipticity. We also observe that $E(e) \sim 10^{-2}$ whereas $E(R^2) \sim 10^{-3}$. This is because these statistics are averages over 26 image sets with different characteristics (see Sect. 6.2). In reality, $E(e)$ varies between $\sim 10^{-2}$ and $\sim 10^{-4}$, whereas $E(R^2) \sim 10^{-3}$ regardless of the sets.
- If we consider $E(e)$ in particular, two groups emerge. The first one contains RBF, IDW and Kriging, with $E(e) \lesssim 1.8 \times 10^{-2}$. The interpolators of the second group, B-splines and Polyfit with $E(e) \gtrsim 2.3 \times 10^{-2}$. We will see below that this is essentially due to the better accuracy of local interpolators on turbulent sets as regards ellipticity. If we focus on $E(R^2)$, the distinction between local and global interpolation schemes disappears. RBF and Polyfit stand out from the others with $E(R^2) \simeq 5 \times 10^{-3}$. We also note that the accuracy of IDW on size is worse by several order of magnitude.
- The errors on the mean $\sigma(e)$ and $\sigma(R^2)$ are on the order of 10^{-4} for all five schemes. As was observed for $E(e)$, we find that the local interpolators RBF, IDW and Kriging reach better $\sigma(e)$ values compared to global ones, B-splines and Polyfit. As for $\sigma(R^2)$, the best values are reached by RBF and Polyfit, similarly to what was found for $E(R^2)$.

6.2. Influence of PSF features simulated in the images

As explained in the Star Challenge result paper (Kitching et al., in prep.), the image sets were designed to simulate typical PSF features found in real astronomical images. Each set implements a unique combination of characteristics against which a method can be evaluated. All 50 images within a set share the same broad features, but differ in the way star positions, sizes and ellipticities are spatially distributed across the field.

The various PSF features tracked in the images are outlined below:

- *PSF model*: the fiducial PSF model includes a static and a dynamic component. The static component is based on a pseudo-Airy (Born & Wolf 1999; Kuijken 2008) or Moffat (Moffat 1969) functional form, depending on the set. The dynamic component made the ellipticity and size of individual stars vary spatially across the image of the PSF field.
- *Star size*: the images from most of the sets share the same "fiducial" 3-pixel FWHM, except sets 6, 14, 26 and sets 7, 15 whose images have respectively a FWHM of 1.5 and 6 pixels.
- *Masking*: sets 2, 10, 22 have a 4-fold symmetric mask denoted as "+" and sets 3, 11, 23 have a 6-fold mask symbolized by a "∗". Images from all other sets are unmasked.
- *Number of stars*: the majority of images contain 1000 stars. Sets 4, 12, 24 are denser, with 2000 stars, whereas sets 5, 13, 25 are sparser, with only 1500 stars.
- *Kolmogorov turbulence (KM)*: an attempt was made on sets 9 to 15, 17, 19 and 21 to simulate the effect of atmospheric turbulence by including a Kolmogorov spectrum in PSF ellipticity. See Heymans et al. (2012); Kitching et al. (in prep.) for the details. Figure 6 shows side by side a non-turbulent and a turbulent PSF.
- *Telescope effect*: a deterministic component was included in sets 17, 19 and 21 to reproduce effects from the telescope optics on the PSF ellipticity and size, essentially primary astigmatism, primary defocus and coma (Born & Wolf 1999), based on the model of Jarvis & Jain (2004).

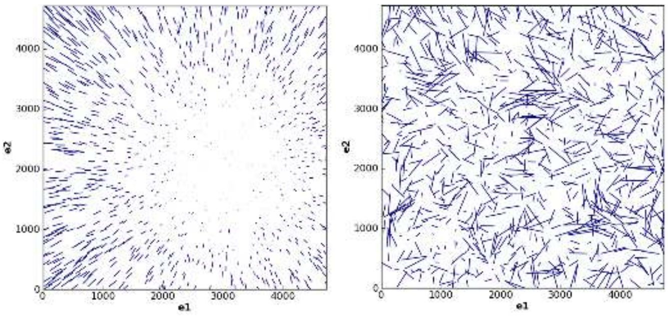

Fig. 6. A Star Challenge non-turbulent PSF (*left*) compared with a turbulent PSF (*right*). Each "whisker" represents the amplitude $|\boldsymbol{e}|$ of the ellipticity of stars. The largest whisker in the left hand side image corresponds to an ellipticity of 0.16. The right hand side image has a maximum ellipticity of 0.37. The ellipticity plots have respectively been made from the first PSF field image of sets 8 and 14.

In order to determine how interpolation schemes are affected by the aforementioned PSF characteristics, we have computed for each of them the performance metrics per individual image sets. We have plotted the metrics $E(e)$ and $E(R^2)$ in Figs. 7 and 8. We analyze the results below.

- *Influence of turbulence*: the PSF feature that affects the interpolation methods the most is the presence of a Kolmogorov (KM) turbulence in ellipticity. Figure 6 illustrates how erratic the spatial variation pattern of ellipticity can become in the presence of KM turbulence. It is clear that a prediction algorithm faces a much more challenging task on turbulent images than on images with more regular PSF patterns. To highlight this, we have averaged in Tables 14 and 15 the metrics E and σ separately over turbulent and non-turbulent sets. Comparing these two tables shows that $E(e) \sim 10^{-4}$ and $\sigma(e) \sim 10^{-5}$ on non-turbulent sets, whereas $E(e) \sim 10^{-2}$ and $\sigma(e) \sim 10^{-3}$ on turbulent sets. This represents a ~100-fold decrease in accuracy and error on the mean. This effect can also be seen on the plots of $E(e)$ in Figs. 7 for and 8.
  We also observe that, on sets without a KM spectrum, all interpolators evaluated in this paper typically reach $\sigma^2_{\text{sys}} \sim 10^{-8}$ already beyond the $\sim 10^{-7}$ goal of next-generation space-based weak lensing surveys. In contrast, sets with turbulent PSF do not match that requirement, with $\sigma^2_{\text{sys}} \sim 10^{-6}$.
  The similarities between $E(e)$ and $\sigma(e)$ values for RBF, IDW and Kriging in Table 15 suggest these methods behave more or less the same when confronted with turbulent ellipticities. To check this, we have have compiled in Fig. 9 the true ellipticity pattern of turbulent set 9 along with the actual predictions of the same pattern by all five interpolators. The same metrics for Polyfit and B-splines show that these global methods are even more handicapped by the presence of a KM spectrum.
  Turbulence also makes the spatial distribution of the FWHM less predictable and the methods are affected to various degrees: RBF, IDW, Polyfit and B-Spline are little influenced with similar $E(R^2)$ and $\sigma(R^2)$ values in Tables 14 and 15 and on the corresponding plots in Figs. 7 and 8. The only one really impacted is Kriging.
- *Influence of star density*: following the discussion of Sect. 3.2, we expect the local interpolation methods to be more accurate than global ones on images with higher star density but see their performance degrade on sparser star fields. Such local interpolators base their predictions on

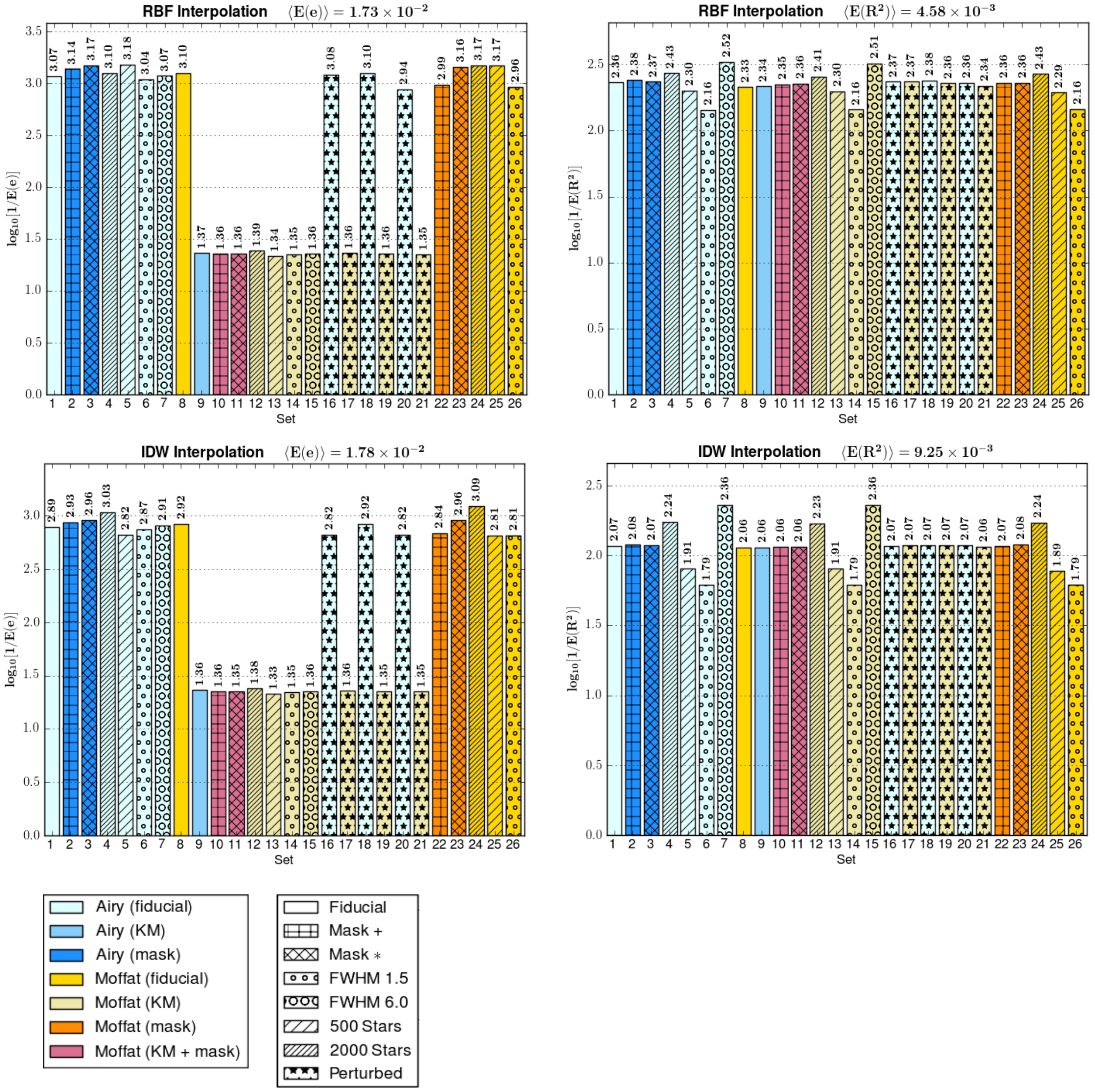

Fig. 7. Accuracy per set for the RBF and IDW interpolation methods. Sets with pseudo-Airy and Moffat are respectively colored in different shades of blue and orange, as specified in the legend at the bottom left of the figure. The various patterns contained in the left hand-side legend indicate the types of physical PSF features simulated in the images. The values on the bars correspond to $\log_{10}(1/E(e))$ and $\log_{10}(1/E(R^2))$ depending on the quantity plotted, so the taller the bar the greater the corresponding accuracy.

observations found in local neighborhoods and should therefore be in position to take advantage of any additional available. On the other hand, they should suffer comparatively more from insufficient sampling when the data is too sparse. This is indeed what we observe in the IDW plot Fig. 7, but the conclusion is less clear regarding RBF and Kriging: these schemes are indeed more accurate on denser sets when it comes to estimate the FWHM but the reverse is seen concerning ellipticities (plots Figs. 7 and 8). This is mostly noticeable on non-turbulent sets and may be caused by some overfitting taking place on denser ellipticity fields. This phenomenon does not occur on FWHM possibly because the FWHM spatial distribution is generally smoother than that of ellipticities in the Star Challenge dataset.

We also expect the global interpolators B-splines and Polyfit to be little affected by difference in star density, since such schemes attempt to find a regression surface that takes all available data into account but at the same time minimize the overall bias through the least squares criterion. Such a surface tends to smooth out small-scale variations, mostly capturing broad features in the image. The corresponding predictions may become less accurate but, on the other hand, remain little influenced by sampling differences. This is exactly what we find in the plots of Polyfit and B-splines Fig. 8.

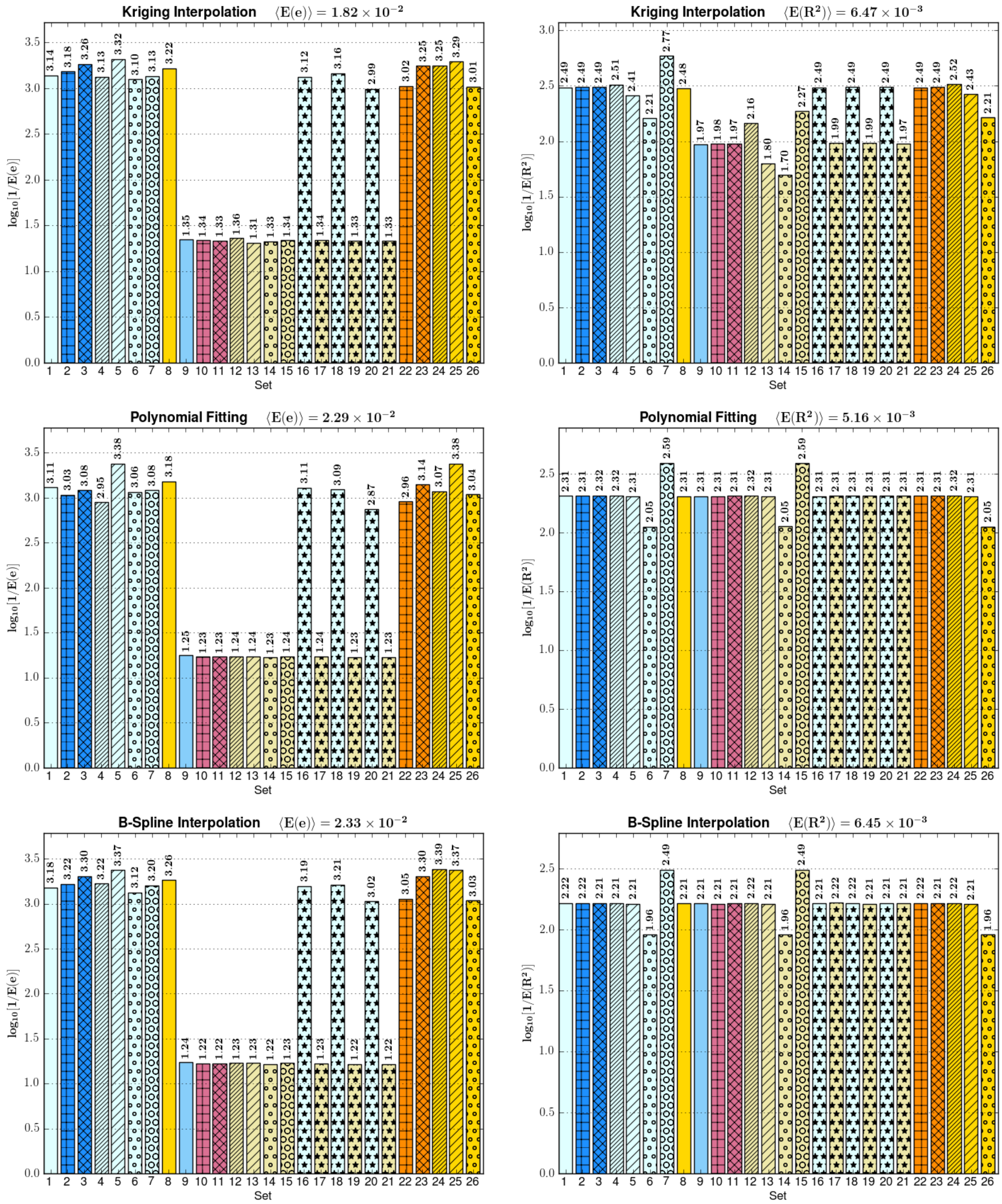

Fig. 8. Accuracy per set for Kriging, polynomial fitting and B-splines. The legend is the same as that used in Fig. 7. The values on the bars correspond to $\log_{10}(1/E(e))$ and $\log_{10}(1/E(R^2))$ depending on the quantity plotted, so the taller the bar the greater the corresponding accuracy.

The smoothness of the prediction surfaces of Polyfit and B-splines compared to that of local interpolators is clearly noticeable in Fig. 9.

- *Influence of the PSF model and size, masking and telescope effects*: although some interpolators do better than others on a particular PSF models, each individual scheme perform equally well on Moffat and Airy images. This can be seen, for example, on fiducial sets 1 and 8 where the error statistics on Moffat or Airy sets are almost identical for a given method. The same can be said of the influence

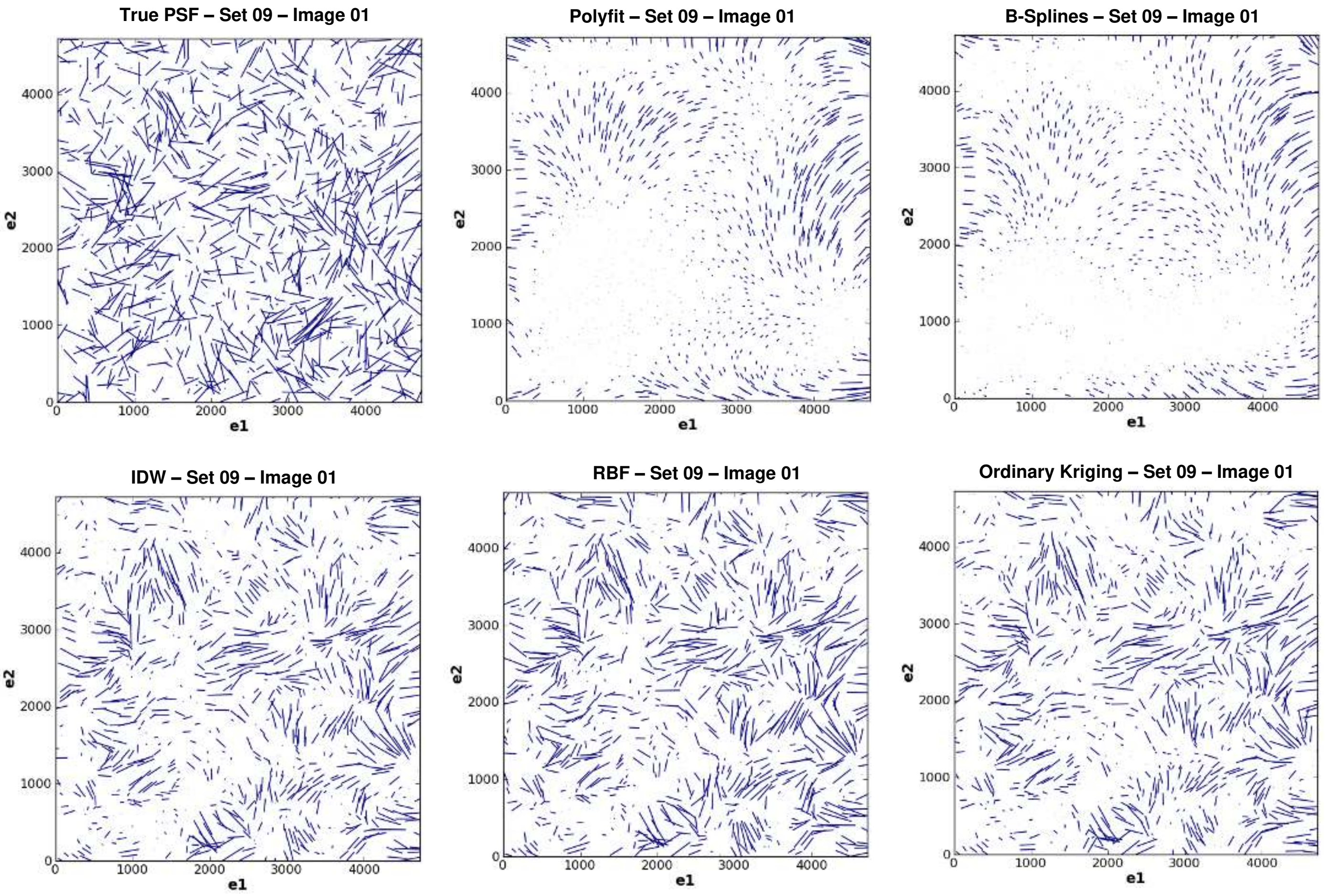

Fig. 9. An illustration of how the various interpolation methods studied in this article handled a turbulent PSF, which in this case is the first image of set 9. The true ellipticities are plotted on the upper-left corner of the figure and the remaining plots show the predictions of each methods. The largest whisker in the upper-left corner plot corresponds to an ellipticity of 0.38.

Table 14. Non-turbulent sets: average values of E and σ.

Method	$E(e)$	$\sigma(e)$	$E(R^2)$	$\sigma(R^2)$
RBF	8.26×10^{-4}	3.60×10^{-5}	4.59×10^{-3}	1.45×10^{-4}
IDW	1.28×10^{-3}	5.67×10^{-5}	9.37×10^{-3}	2.95×10^{-4}
Kriging	7.06×10^{-4}	3.16×10^{-5}	3.57×10^{-3}	1.13×10^{-4}
Polyfit	8.37×10^{-4}	3.73×10^{-5}	5.23×10^{-3}	1.64×10^{-4}
B-splines	6.28×10^{-4}	2.80×10^{-5}	6.53×10^{-3}	2.06×10^{-4}

Table 15. Turbulent sets: average values of E and σ.

Method	$E(e)$	$\sigma(e)$	$E(R^2)$	$\sigma(R^2)$
RBF	4.36×10^{-2}	1.81×10^{-3}	4.57×10^{-3}	1.44×10^{-4}
IDW	4.42×10^{-2}	1.79×10^{-3}	9.05×10^{-3}	2.85×10^{-4}
Kriging	4.61×10^{-2}	1.79×10^{-3}	1.11×10^{-2}	3.49×10^{-4}
Polyfit	5.82×10^{-2}	1.89×10^{-3}	5.04×10^{-3}	1.58×10^{-4}
B-splines	5.97×10^{-2}	1.88×10^{-3}	6.31×10^{-3}	1.99×10^{-4}

of FWHM, masking and telescope effects. We also observe star size to have a negligible impact on $E(e)$ for all methods, but we clearly see that $E(R^2)$ significantly increases (resp. decreases) for star fields with smaller (resp. larger) FWHM. Finally, all methods reach a slightly higher accuracy on masked images, especially with 6-fold masks.

6.3. Results from individual interpolation methods

- *Interpolation with radial basis functions (RBF)*: as shown in our previous discussion, the RBF interpolation scheme is the overall winner of our evaluation. According to our benchmarks, ellipticity patterns were best estimated by a *linear kernel* function, whereas a *thin-plate kernel* was more effective on FWHM values. A neighborhood size between 30 and 40 stars was used. Refer to Sect. 3.5 and Table 5 for a description of RBF interpolation and the definitions of these kernels. That combination of linear and thin-plate kernels yields very competitive error statistics on both turbulent and non-turbulent sets: Tables 14 and 15 as well as plots Fig. 7 show RBF is the most accurate on turbulent sets whereas its results on non-turbulent sets are the second best behind ordinary Kriging. The possibility of selecting the most suitable kernel for a given PSF patterns is a very attractive feature of RBF interpolation.
- *Inverse distance weighted interpolation (IDW)*: the IDW methods (see Sect. 3.4) obtains the second best average $E(e)$ behind RBF over all sets as seen in Table 13. It does so thanks to very competitive $E(e)$ results on turbulent sets, just behind RBF (Table 15). But IDW's estimates of the FWHM

on non-turbulent sets are by far the worst of all five interpolation algorithms, both on ellipticity and size (Table 14). As found in Sect. 6.2, IDW looks quite sensitive to variations in star density. In fact, we observe that IDW underperforms on star fields with low-density and smaller FWHM (sets 5, 6, 13, 14, 25, 26). We were unable to find a setup that significantly improves that level of accuracy, which suggests the method has difficulty coping with such constraints in density and size. All in all, IDW performs quite well overall, knowing it is based on a very simple interpolation algorithm, with fewer adjustable parameters than RBF or ordinary Kriging (see Sect. 3.4).

- *Interpolation with ordinary Kriging (OK)*: despite its reputation of best interpolator on spatially-scattered data, ordinary Kriging, introduced in Sect. 3.6, arrives only third behind RBF and IDW when considering error statistics in Table 13. As see in shown in Table 14 and plots Fig. 8, Kriging's estimates on non-perturbed sets are the best of all five methods. But this cannot compensate for its relatively poor performance on estimating the FWHM on turbulent sets, as shown in the value of $E(R^2)$ in Table 15. The reason for this is probably related to the significant spatial drift of the FWHM values across the image. The condition of intrinsic stationarity required by ordinary Kriging is no longer fulfilled in some areas, especially near the edges of the image. As a result, we were forced to reduce the size of the search neighborhood over which the Kriging weights are calculated, which leads to a loss in accuracy in the corresponding regions. Kriging variants with ability to correct such a drift, like Universal Kriging, would probably achieve better results. Also, our implementation of Kriging for the Star Challenge assumes spatial isotropy, even though experimental variograms for ellipticity on non-turbulent sets also show evidence of geometric anisotropy, A more sophisticated implementation could have corrected these effects by rescaling and rotating coordinate axes along the direction of maximum spatial continuity.
- *Polynomial fitting (Polyfit)*: the results of Polyfit are of particular interest since polynomial fitting is currently the method of choice for modeling spatial variations of a PSF in lensing studies (see Sects. 2 and 3.1). polynomial fitting performs relatively well on non-turbulent sets with $E(e)$ and $E(R^2)$ statistics fairly close to those of RBF (Table 14). However, the corresponding statistics on turbulent sets are significantly worse that those achieved by local methods, as seen in Table 15. This confirms the conclusion of Sect. 3.1 whereby polynomials have difficulty coping with small or rapid variations found in a PSF pattern. Low-degree polynomials generally produce satisfactory result but tend to underfit the data, which leads to suboptimal accuracy. The resulting interpolation surfaces are characteristically smooth, as clearly observed in the Polyfit plot of Fig. 9. The Star Challenge images without KM power spectrum are smooth enough for Polyfit to approach the accuracy of RBF and ordinary Kriging. These results were obtained with a fifth-degree polynomial, higher degrees degrading the fit.
- *Interpolation with basis splines (B-splines)*: polynomial splines are generally considered superior for interpolation than simple polynomials as explained in Sect. 3.3, and we would have expected B-splines to achieve better results than Polyfit on the Star Challenge data. But this is not reflected in the averaged results from Tables 13. The level of accuracy reached by both interpolators is nevertheless of the same order.

  As seen in Table 14 and plots Fig. 8, the ellipticity estimates from B-splines are superior to those of Polyfit on non-turbulent sets and of similar accuracy on turbulent ones. This tends to confirm the better ability of splines to capture small-scale and rapid variations in the data than polynomials. The results show, however, errors $E(R^2)$ on the FWHM much larger for B-splines than for Polyfit, which explains the relative lower performance compared to Polyfit. The FWHM spatial distribution being overall quite smooth in the Star Challenge images, this result suggests polynomials may be better suited than splines for modeling smoothly-varying patterns of variation. Combining both schemes may also be worth investigating.

7. Conclusions

The GREAT10 Star Challenge gave us the opportunity to evaluate several interpolation methods on spatially-varying PSF fields:

- Two global, approximate and deterministic spatial interpolation schemes: polynomial fitting (Polyfit) and basis splines (B-splines).
- Two local, exact and deterministic techniques relying on inverse distance weighting (IDW) and radial basis functions (RBF).
- An implementation of ordinary Kriging, a local, exact and stochastic spatial prediction method, frequently used in Geostatistics and environmental sciences.

We used a three-stage PSF estimation pipeline, which we described in Sect. 4.2 and Appendix A. Elliptical Moffat profiles were fitted to the stars contained in each Star Challenge image and then estimated and reconstructed at new positions in the same image using one of the five interpolation schemes listed above.

That approach proved quite successful since it allowed us to win the GREAT10 Star Challenge. We were, however, disappointed by the relatively high $\sigma^2_{\rm sys}$ values reached, on the order of 10^{-4}, i.e., still far from the $\sigma^2_{\rm sys} \lesssim 10^{-7}$ target demanded by future large weak lensing surveys. The lack of accuracy could be traced to the suboptimal fitting of Airy PSF profiles by our pipeline and not to a deficiency in the PSF interpolation methods. However, this issue made it difficult to unambiguously conclude on the level of accuracy of individual interpolation algorithms, which is the main objective of this article.

In order to measure errors purely due to interpolation and only these, we used the true input ellipticity and FWHM catalog for the input Star Challenge images instead of our fitted estimates for these quantities. We also chose new metrics, better suited than the P-factor for assessing estimates on ellipticity and size. The results are summarized in Tables 13–15 along with the corresponding plots in Figs. 7 and 8. We highlight our main conclusions below.

- Table 13 shows the overall $E(e)$ and $E(R^2)$ errors to be on the order of 10^{-2} and 10^{-3} respectively. Figure 14, however indicates that $E(e) \sim 10^{-4}$ and $E(R^2) \sim 10^{-3}$ on images devoid of Kolmogorov turbulence, to be compared with the $E(e) \lesssim 10^{-3}$ and $E(R^2) \lesssim 10^{-3}$ estimated requirements of future next-generation surveys. Although the Star Challenge PSF fields lack realism in certain aspects, this suggests that the best methods, RBF, IDW and OK, may already be suitable for space-based surveys where turbulence is absent.

- All interpolation methods see their accuracy drastically degraded in images where atmospheric turbulence effects have been simulated, with $E(e)$ and $E(R^2)$ errors increased by a factor of ~100. The better performance on turbulent images of RBF, IDW and OK compared to Polyfit and B-splines in the GREAT10 Star Challenge, suggests local methods may be able to better cope with turbulence than global ones. We note, however, that these results are only valid for the specific turbulence model used in the simulations and would have to be confirmed on real data.
- After turbulence, the factors influencing results the most are the density of stars and their size. As far as density is concerned, local methods are more impacted than global ones and generally improve their estimates on denser sets much more than global methods. A similar conclusion is reached concerning local methods as far as PSF size is concerned. However, the results suggest both global and local methods have difficulty coping with objects smaller than the fiducial FWHM of 3 pixels. Among all methods, IDW suffered the most from sparse star fields with small FWHM.
- The RBF interpolator proved the most accurate, reaching the best results on both turbulent and non-turbulent sets. The use of kernel functions brings additional versatility compared to a simpler interpolator like IDW, while avoiding the complexity of Kriging. The selection of the most suitable kernel function and associated parameters can be greatly simplified by the use of cross-validation or Jackknifing. These techniques, as shown in Sect. 4.3, can prove very helpful to tune the runtime parameters of an interpolation schemes and evaluate the accuracy of its results.
- Despite its simplicity, the IDW interpolation method obtained better than expected results, outperforming polynomials and splines in the simulations. Fast and easy to tune, it could potentially constitute a simple alternative/complement to polynomials before trying more elaborate interpolation schemes such as Kriging or RBF.
- Ordinary Kriging is, in our opinion, potentially the most accurate method as shown especially by its results on non-turbulent images. However, the FWHM spatial distributions in the Star Challenge have a significant spatial drift that the standard ordinary Kriging algorithm is unable to correct. Another Kriging variant such as Universal Kriging would possibly have proved more accurate. It remains that Kriging, because of its sophistication, is more difficult and time consuming to operate than the other interpolators we evaluated.
- Overall, our analysis of the Star Challenge results suggests local interpolators should be preferred over global ones based on splines and polynomials. However, one should bear in mind that (1) these results are based on simulated data where star images are isolated, bright enough and well sampled; (2) the spatial variation of the PSF as simulated in GREAT10 may tend to favor local interpolators over global ones. We strongly believe, nevertheless, that local interpolation schemes for PSF interpolation have the potential to improve the accuracy of existing and future ground-based lensing surveys and deserve to be investigated further.

Acknowledgements. We thank Tom Kitching for his help and especially for providing us with the true input ellipticity and FWHM catalog of the Star Challenge PSF images. We also acknowledge support from the International Space Science Institute (ISSI) in Bern, where some of this research has been discussed. This work is supported by the Swiss National Science Foundation (SNSF).

Appendix A: Our PSF prediction pipeline (PSFPP)

A.1. Overview

The PSF prediction pipeline used in the Star Challenge is outlined in Fig. 5. A PSF field is fed into the pipeline and goes through three processing stages:

1. *Fitting* stage: the Moffat PSF model described in Sect. 4.1 is fitted to each star at known position (x_c, y_c) in the PSF field image. A catalog is produced, containing a set of fitted parameters $\{(x_c, y_c); (e_1, e_2); \phi, (\alpha, \beta)\}$ for each star. Instead of an out-the-box minimizer, we employ a custom minimizer we developed at the EPFL Laboratory of astrophysics and well suited to fitting faint and noisy images like those frequently found in weak lensing. The minimizer uses an "adaptive cyclic coordinate descent algorithm" that find a local minimum with the lowest χ^2 of the residuals. That same minimizer has also been used in the version of the *gfit* shear measurement method that competed in the GREAT10 Galaxy Challenge (Kitching et al. 2012). The star images processed by the minimizer are 16 × 16-pixel cutouts instead of the original 48 × 48-pixel postage stamps.
2. *Prediction* stage:
   - First, an analysis of the spatial distribution of each parameter across the image is performed. In particular, all separation distances between stars are recorded in the form of *KD-trees* (Bentley 1975) for efficiently finding the nearest neighboring stars located within a given separation distance $\|\boldsymbol{h}\|$.
   - Second, a spatial prediction scheme is applied to estimate the values $Z'_p(x'_i, y'_i)$ of the parameter p at asked locations (x'_i, y'_i), given the fitted parameter values $Z_p(x_i, y_i)$ obtained in the previous stage. One of the four methods described in Sect. 3 is applied here.
3. *Reconstruction* stage: All stars in a PSF field are reconstructed based on the elliptical Moffat model described in Sect. 4.1, but using the parameters predicted for that star during the *Prediction* stage.

A.2. Pipeline implementation and configuration

The pipeline code is written in Python, a programming language known for its power, flexibility and short development cycle. The usual standard Python libraries are used, notably: NumPy, SciPy, PyFITS and matplotlib. SciPy is the standard scientific library for Python. Most of its functions are thin Python wrappers on top of fortran, C and C++ functions. SciPy takes advantage of installed optimized libraries such as LAPACK (Linear Algebra PACKage) library (Anderson et al. 1990). We employ the cross-validation and Jackknifing resampling techniques (see Sect. 4.3) to tune the run-time parameters for the interpolation schemes and evaluate the accuracy of the results. We highlight below a few aspects related to the implementation of the methods.

- IDW: the code for Inverse Distance Weighted interpolation is written in Python, based on Eq. (2) with weighting factors specified by (8). The free parameters are the power factor β and the neighborhood size N (see Sect. 3.4). A configuration with $\beta = 2$, with $5 \leq N \leq 15$ depending on the density of stars in images gives the best results according to our tests.
- RBF: we use the `rbf()` interpolation function available in the *SciPy interpolate* module. The number of parameters to tune is greater compared to IDW: a kernel function chosen among those listed in Table 5; the neighborhood search size

N; a shape parameter ϵ for the *multiquadric*, *inverse multiquadric* and *Gaussian* kernels; and a last parameter for controlling the smoothness of the interpolation (see Sect. 3.5). Only the *linear*, *thin-plate* and *multiquadric* kernels gave stable enough predictions. Choosing $25 \leq N \leq 30$ and disabling smoothing (i.e. use exact interpolation) yielded the best cross-validation and Jackknifing results for the chosen kernels.

- splines: we have selected the `bisplrep()` and `bisplev()` bivariate B-spline interpolation functions provided by the SciPy *interpolate* module. These functions are Python wrappers on top of the fortran FITPACK package (Dierckx 1995). The underlying algorithms follow the *constructive* approach for spline interpolation described in Sect. 3.3 and are specified in Dierckx (1980). The main parameters affecting the interpolation are the degree p of the spline, the number of knots N and a smoothing factor s. We have fixed p to 3 but let the algorithm automatically set N and s.
- Kriging: we have used our own custom-developed Python code of ordinary Kriging (see Sect. 3.6). The Kriging used in the Star Challenge and in this article is isotropic and does not implement any spatial anisotropy or drift correction scheme. The accuracy of the ordinary Kriging interpolation scheme was influenced by the following set of parameters:
  - The interpolation range, i.e. the range in pixels used for interpolation. Depending on the images, we chose a circular area with a radius between 700 and 1000 pixels from the center of the 4800×4800 PSF field.
  - Lag distance h in pixels. We used values in the range $100 \leq h \leq 300$ depending on the image and the PSF model parameter to estimate.
  - The number of observations N in Eq. (19) to include in the neighborhood: we used $5 \leq N \leq 20$ depending on the image star density.
  - Tolerance distance Δh (pixels) and angle $\Delta\theta$ considered when locating neighboring observations. As a rule of thumb, we selected $\Delta h \approx h/2$ and $\Delta\theta = 22.5°$.
  - A theoretical variogram model such as those listed in Table 7. The experimental variograms were fitted using the *Levenberg-Marquardt* least-squares `leastsq` routine from the SciPy *optimize* module. The program dynamically selected the theoretical variogram models and parameters that produced the best fit.
- The Polyfit code is based on the `leastsq()` function from the *SciPy optimize* Python module. A least-squares fit to a bivariate polynomial of degree 5 gave the best estimates.

References

Abramowitz, M., & Stegun, I. A. 1965, Handbook of Mathematical Functions, 1st edn., Dover books on mathematics (Dover Publications)

Amara, A., & Réfrégier, A. 2008, MNRAS, 391, 228

Anderson, E., Bai, Z., Dongarra, J., et al. 1990, in Proc. 1990 ACM/IEEE conference on Supercomputing, Supercomputing '90 (Los Alamitos, CA, USA: IEEE Computer Society Press), 2

Bentley, J. L. 1975, Commun. ACM, 18, 509

Bergé, J., Pacaud, F., Réfrégier, A., et al. 2008, MNRAS, 385, 695

Bergé, J., Price, S., Amara, A., & Rhodes, J. 2012, MNRAS, 419, 2356

Bernstein, G. M., & Jarvis, M. 2002, AJ, 123, 583

Born, M., & Wolf, E. 1999, Principles of Optics: Electromagnetic Theory of Propagation, Interference and Diffraction of Light, ed. C. U. Press, Vol. 7th Edition (Cambridge University Press), 986

Bridle, S., Balan, S. T., Bethge, M., et al. 2010, MNRAS, 405, 2044

Browne, M. 2000, J. Math. Psychol., 44, 108

Brus, D. J., Gruijter, J. J. D., Marsman, B. A., et al. 1996, Environmetrics, 7, 1

Buhmann, M. D. 2003, Radial basis functions: theory and implementations (Cambridge University Press), 12, 274

Burrough, P. 1988, Principles of geographical information systems for land resources assessment, Monographs on soil and resources survey (Oxford University Press)

Burrough, P., & McDonnell, R. 1998, Principles of geographical information systems, Spatial information systems and geostatistics (Oxford University Press)

Chilès, J., & Delfiner, P. 1999, Geostatistics: modeling spatial uncertainty, Wiley series in probability and statistics (Wiley)

Clowe, D., & Schneider, P. 2002, A&A, 395, 385

Collins, F. C., & Bolstad, P. V. 1996, A Comparison of Spatial Interpolation Techniques in Temperature Estimation (USA: NCGIA – National Center for Geographic Information and Analysis)

Cox, M. G. 1972, IMA J. Appl. Math., 10, 134

Craven, P., & Wahba, G. 1978, Numerische Mathematik, 31, 377

Cressie, N. 1991, Statistics for spatial data, Wiley series in probability and mathematical statistics: Applied probability and statistics (J. Wiley)

Davis, B. M. 1987, Mathematical Geology, 19, 241

de Boor, C. 1972, J. Approximation Theory, 6, 50, collection of articles dedicated to J. L. Walsh on his 75th birthday, V (Proc. Internat. Conf. Approximation Theory, Related Topics and their Applications, Univ. Maryland, College Park, Md., 1970)

de Boor, C. 1978, A Practical Guide to Splines (Springer-Verlag Berlin and Heidelberg GmbH & Co. K)

Deutsch, C. V., & Journel, A. 1997, GSLIB: geostatistical software library and user's guide, Applied Geostatistics (Oxford University Press)

Dierckx, P. 1980, An algorithm for surface fitting with spline functions (Katholieke Univ. Leuven)

Dierckx, P. 1995, Curve and surface fitting with splines, Monographs on numerical analysis (Clarendon Press)

Duchon, J. 1976, RAIRO Analyse numérique, 10, 1

Efron, B. 1982, The Jackknife, the Bootstrap and Other Resampling Plans, ed. S. I. A. MEditors (Society for Industrial and Applied Mathematics), 38, 92

Efron, B., & Gong, G. 1983, Am. Stat., 37, 36

Foley, T. A. 1994, J. Appl. Sci. Comput., 1, 51

Franke, R. 1982, Mathematics of Computation, 38, 181

Fu, L., Semboloni, E., Hoekstra, H., et al. 2008, A&A, 479, 9

Geisser, S. 1975, J. Am. Stat. Assoc., 70, 320

Goovaerts, P. 1997, Geostatistics for natural resources evaluation, Applied geostatistics series (Oxford University Press)

Green, P., & Silverman, B. 1994, Nonparametric regression and generalized linear models: a roughness penalty approach, Monographs on statistics and applied probability (Chapman & Hall)

Hardy, R. L. 1971, J. Geophys. Res., 76, 1905

Hardy, R. L. 1990, Computers Mathematics with Applications, 19, 163

Hayes, J. G., & Halliday, J. 1994, Teaching Mathematics and its Applications, 14, 89

Hetterscheidt, M., Simon, P., Schirmer, M., et al. 2007, A&A, 468, 859

Heymans, C., Brown, M. L., Barden, M., et al. 2005, MNRAS, 361, 160

Heymans, C., Van Waerbeke, L., Bacon, D., et al. 2006, MNRAS, 368, 1323

Heymans, C., Rowe, B., Hoekstra, H., et al. 2012, MNRAS, 421, 381

Hirata, C., & Seljak, U. 2003, MNRAS, 343, 459

Hoekstra, H. 2004, MNRAS, 347, 1337

Hoekstra, H., Franx, M., Kuijken, K., & Squires, G. 1998, ApJ, 504, 636

Hutchinson, M. F. 1995, International journal of geographical information systems, 9, 385

Hutchinson, M., & Gessler, P. 1994, Geoderma, 62, 45

Isaaks, E. H., & Srivastava, R. 1989, Applied geostatistics (Oxford University Press)

Jain, B., Jarvis, M., & Bernstein, G. 2006, J. Cosmology Astropart. Phys., 2, 1

Jarvis, M., & Jain, B. 2004, unpublished [`arXiv:0412234`]

Jarvis, M., Schechter, P., & Jain, B. 2008, PASP submitted [`arXiv:0810.0027`]

Jee, M. J., & Tyson, J. A. 2011, PASP, 123, 596

Jee, M. J., White, R. L., Benítez, N., et al. 2005a, ApJ, 618, 46

Jee, M. J., White, R. L., Ford, H. C., et al. 2005b [`arXiv:0508044`]

Jee, M. J., White, R. L., Ford, H. C., et al. 2006, ApJ, 642, 720

Jee, M. J., Blakeslee, J. P., Sirianni, M., et al. 2007a, PASP, 119, 1403

Jee, M. J., Ford, H. C., Illingworth, G. D., et al. 2007b, ApJ, 661, 728

Journel, A., & Huijbregts, C. 1978, Mining geostatistics (Academic Press)

Kaiser, N. 2000, ApJ, 537, 555

Kaiser, N., Squires, G., & Broadhurst, T. 1995, ApJ, 449, 460

Kitching, T. D., Amara, A., Abdalla, F. B., Joachimi, B., & Refregier, A. 2009, MNRAS, 399, 2107

Kitching, T., Amara, A., Gill, M., et al. 2011, Ann. Appl. Stat., 5, 2231

Kitching, T. D., Balan, S. T., Bridle, S., et al. 2012, MNRAS, 423, 3163

Krige, D. G. 1951, J. Chem. Metall. Mining Soc. of South Africa, 52, 119

Kuijken, K. 2008, A&A, 482, 1053

Laslett, G. M., McBratney, A. B., Pahl, P. J., & Hutchinson, M. F. 1987, J. Soil Sci., 38, 325
Lauer, T. 2002, in SPIE Conf. Ser. 4847, eds. J.-L. Starck, & F. D. Murtagh, 167
Lukaszyk, S. 2004, Computational Mechanics, 33, 299
Luppino, G. A., & Kaiser, N. 1997, ApJ, 475, 20
Lupton, R., Gunn, J. E., Ivezić, Z., et al. 2001, in Astronomical Data Analysis Software and Systems X, eds. F. R. Harnden Jr., F. A. Primini, & H. E. Payne, ASP Conf. Ser., 238, 269
Massey, R., & Refregier, A. 2005, MNRAS, 363, 197
Massey, R., Heymans, C., Bergé, J., et al. 2007, MNRAS, 376, 13
Matheron, G. 1962, Traité de Géostatistiques Appliquées, Tome I., Vol. 14 (Editions Technip)
Matheron, G. 1963, Economic Geology, 58, 1246
Matheron, G. 1965, Les variables régionalisées et leur estimation, Paris
Meinguet, J. 1979, ZAMP, 30, 292
Melchior, P., Böhnert, A., Lombardi, M., & Bartelmann, M. 2010, A&A, 510, A75
Micchelli, C. A. 1986, Constructive Approximation, 2, 11
Miller, R. G. 1974, Biometrika, 61, 1
Moffat, A. F. J. 1969, A&A, 3, 455
Nakajima, R., Bernstein, G. M., Fadely, R., Keeton, C. R., & Schrabback, T. 2009, ApJ, 697, 1793
Paulin-Henriksson, S., Antonuccio-Delogu, V., Haines, C. P., et al. 2007, A&A, 467, 427
Paulin-Henriksson, S., Amara, A., Voigt, L., Refregier, A., & Bridle, S. L. 2008, A&A, 484, 67
Paulin-Henriksson, S., Refregier, A., & Amara, A. 2009, A&A, 500, 647
Prenter, P. 2008, Splines and Variational Methods (Dover Publications)
Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. 2007, Numerical Recipes, The Art of Scientific Computing, 3rd edn. (Cambridge University Press)
Quenouille, M. H. 1956, Biometrika, 43, 353
Refregier, A. 2003, MNRAS, 338, 35
Refregier, A., & Bacon, D. 2003, MNRAS, 338, 48
Renka, R. J. 1988, ACM Trans. Math. Softw., 14, 139
Rippa, S. 1999, Adv. Comp. Math., 11, 193
Romano, A., Fu, L., Giordano, F., et al. 2010, A&A, 514, A88
Schrabback, T., Hartlap, J., Joachimi, B., et al. 2010, A&A, 516, A63
Schumaker, L. 2007, Spline functions: basic theory, Cambridge mathematical library (Cambridge University Press)
Shepard, D. 1968, in ACM '68: Proc. 1968 23rd ACM national conference (New York, NY, USA: ACM), 517
Stabenau, H. F., Jain, B., Bernstein, G., & Lampton, M. 2007, ArXiv e-prints
Stone, M. 1974, J. Roy. Stat. Soc. Ser. B Methodol., 36, 111
Tomczak, M. 1998, J. Geog. Inf. Dec. Anal., 2, 18
Tukey, J. 1977, Exploratory data analysis, Addison-Wesley series in behavioral sciences (Addison-Wesley Pub. Co.)
Tukey, J. W. 1958, Bias and confidence in not-quite large samples
Umetsu, K., Medezinski, E., Broadhurst, T., et al. 2010, ApJ, 714, 1470
Van Waerbeke, L., Mellier, Y., Radovich, M., et al. 2001, A&A, 374, 757
Van Waerbeke, L., Mellier, Y., Pelló, R., et al. 2002, A&A, 393, 369
Van Waerbeke, L., Mellier, Y., & Hoekstra, H. 2005, A&A, 429, 75
Wackernagel, H. 2003, Multivariate geostatistics: an introduction with applications (Springer)
Wahba, G. 1990, Spline models for observational data, CBMS-NSF regional conference series in applied mathematics (Society for Industrial and Applied Mathematics)
Wahba, G., & Wendelberger, J. 1980, Monthly Weather Rev., 108, 1122
Waller, L., & Gotway, C. 2004, Applied spatial statistics for public health data, Wiley series in probability and statistics (John Wiley & Sons)
Webster, R., & Oliver, M. 2007, Geostatistics for environmental scientists, Statistics in practice (Wiley)