An introduction to *shapelets* based weak lensing image processing.

Volume I : Shapelets
Shapelets package version 2.2
Manual for users

Version 2.0

Joel Bergé, Richard Massey & Alexandre Réfrégier

An introduction to shapelets based weak lensing image processing.
Volume I : *Shapelets*

◊◊◊

Joel Bergé

CEA-Saclay
Service d'Astrophysique
Orme des Merisiers, Bat 709
91191 Gif-sur-Yvette
FRANCE
e-mail : joel.berge@cea.fr

◊◊◊

Richard Massey

California Institute of Technology
MC 105-24
1200 East California Boulevard
Pasadena, CA 91125
USA
e-mail : rjm@astro.caltech.edu

◊◊◊

Alexandre Refregier

CEA-Saclay
Service d'Astrophysique
Orme des Merisiers, Bat 709
91191 Gif-sur-Yvette
FRANCE
e-mail : refregier@cea.fr

Foreword

This manuscript is an updated version of the first "Introduction to shapelets based weak lensing image processing". For convenience, the original manual has been split into two volumes. The current Volume I deals with, and only with, shapelets formalism and the publicly available shapelets software ; the upcoming Volume II deals with the whole method, based upon shapelets, that we use in weak lensing data analysis, particularly PSF modelling.

This volume can still be considered as twofold : a brief overview of the shapelets formalism gives their most useful features ; then most of the shapelets software's methods and IDL routines are described. Of course, explaining each routine of the package would have been not only painless, but also pointless, and only the most important ones have been dealt with. In that sense, shapelets algebra codes are not dealt with, since this manuscript, aiming at emphasizing cosmic shear analysis, focuses on the shape measurement with shapelets. An exhaustive list of the shapelets package's routines is given in an appendix, though, and some help can be found on the Internet, at the below address.

Thank you Alex and Richard for your support when I was learning how to use shapelets. I want to thank Mandeep Gill for comments on the first version of this manuscript, and Richard Ellis for providing me computer space at Caltech, where I can find a home for this manual.

All routines of the shapelets package can be downloaded at the URL :
http://www.astro.caltech.edu/∼rjm/shapelets/code/

Updated versions of this manuscript can be found at the URL :
http://www.astro.caltech.edu/∼jberge/shapelets/manual/

For any comment or suggestion, feel free to contact me directly.

J.B., May 13, 2006

Contents

Introduction

Because it relies only on General Relativity, and not on the physics of halos, weak lensing represents an observable directly related to the mass (or to the potential) of halos, and therefore, it allows to have recourse to scaling laws (that we do not necessarily perfectly know), from which degeneracies in the detection of clusters of galaxies can originate (for instance, using X-ray wavelengths, or the Sunyaev Zel'dovich effect. That is why one can define weak lensing as a unique probe to detect dark matter halos.

This dark matter has been expected to exist since the mid 20th century, when astronomers noticed a discrepancy between virial and luminosity masses of clusters of galaxies. Later, they observed that the rotation curves of galaxies were flat, and could not be explained through the luminous mass only : dark matter halo was "added" around all galaxies.

Nowadays, some new theories are said to be able to explain these discrepancies without invoking dark matter, but just a modified potential. The MOND paradigm [3] deals with a modified newtonian dynamics, which may act almost the same way as the relativistic potential of General Relativity, but without dark matter. Despite some troubles in this new theory, it is said to predict, between others, the rotation curves of galaxies, the Tully-Fisher law for galaxies, and weak lensing. Intuitively, thinking that General Relativity may be wrong for great masses/potential and/or great spatial scales is more elegant than introducing a strange dark matter never seen in more than twenty years (even though supersymmetry theories could account for it...). Although this question is still wide open, it is not our intend here to take part in this debate.

Whatever dark matter exists or not, weak lensing is expected to occur, because it depends on the gravitational potential ; consequently, given a mass greater than the one astronomers see (dark matter paradigm), or given a different gravitational potential than the one General Relativity predicts for the mass we see, the effects on local space-time will be the same. Hence, weak lensing must be detectable in both theories. It could even, maybe, allow astronomers to discriminate against concurrent theories : if someday one detects a dark clump, *i.e.* a halo only composed of dark matter, with no luminous counterpart, then dark matter would become more strongly accepted. Or, if weak lensing properties (its power spectrum for example) could be different on both theories, one could expect to use it in order to confirm a theory.

In the framework of a theory of gravitation, weak lensing is due to the effect of a gravitational potential on its neighborhood. Indeed, according to the duality between matter (energy) and space pointed out by Einstein, a gravitational potential curves space-time. Consequently, geodesics are no longer straight lines, but they are curved by gravitational wells, "glued" to space-time ; hence, as all kind of matter travelling through space follow geodesics, this stands for photons : their trip is not straight, but perturbed by all the gravitational wells they encounter from their emission to their detection by an observer.

Each geodesic is unique, and so two light rays emitted by two different parts of a same galaxy will be differently curved, *i.e.* deviated, by a same gravitational well, just because they travel on two different geodesics : that is what is called differential deviation. As a result, the image one sees of a distant galaxy will be distorted : one does not see the galaxy as it really is, but amplified and sheared. There is a direct analogy with optical lenses : physicists talk about lensing. Some cases appear in which the gravitational potential is so large that the image of a distant galaxy is extremely amplified and sheared, and appears as an arc : this is strong lensing. In other cases, the distortion is pretty negligible and is rather called weak lensing or cosmic shear, and that is what weak lensing observers intend to detect and measure. For details about lensing, the reader is referred to [2, 23, 25].

Since General Relativity predicts it, the idea of (both strong and weak) gravitational lensing grew very early, in the first half of the 20th century, initiated by Zwicky. In the 1980s, a team first observed strong lensing [30, 31], with gravitational arcs in some clusters. Then, in the early 1990s, weak lensing was detected in clusters of galaxies [8]. Since the early 2000s, astronomers have had more and more accurate telescopes and methods dedicated to weak lensing, and cosmic shear due to large scale structures is currently being detected and measured [1, 16, 33].

If it took so long to detect and to measure cosmic shear, that is because its effect is rather small, because large surveys are needed, and because astronomical images are used to be smeared. Indeed, cosmic shear is particularly a statistical effect : it is so small that it is absolutely impossible to detect it on a single galaxy. One has to measure the shape of lots of galaxies so as to detect a shear. The larger and the deeper the image, the better the results. Not so long ago, no large surveys existed, and therefore, no cosmic shear could be detected. Furthermore, the effect of weak lensing being so small, astronomers need excellent telescopes. They have had some for just a few years (Subaru Telescope [32], CHFT and its Megacam camera, Keck Telescopes, Hubble Space Telescope...). But the main difficulty is due to the response of the telescope and the camera, the Point Spread Function (PSF). On the one hand, a CCD camera smears images (the smearing depends on the camera, and can be known very accurately and easily corrected for), on the other hand, the atmosphere troubles images. These two effects add to each other, creating this so called PSF, which is convolved to the "real" image (the one one could see in space, with perfect camera and telescope). It smears images and therefore, contaminates weak lensing measurements. Consequently, if one wants to measure the shear on an image, one must correct it for the PSF. Knowing furthermore that the PSF has a 8-10 % effect, when cosmic shear is expected to be a 1-3 % effect, it is compulsory to correct for the PSF very well. For the moment, this remains the main systematic in weak lensing analyses, even if new methods allow to correct for it in satisfactory ways. Other systematics exist, such as intrinsic correlations between close pairs of galaxies whose (physical) shapes depend on the density field they inhabit, or intrinsic ellipticity - shear correlation between distant galaxies, but in the same line of sight, one being physically sheared, and the other one lensed, by the same potential. Those used to be assumed to be negligeable, but this assumption seems to be wrong, and one should be very cautious when measuring a cosmic shear effect (see [9, 11, 12, 19] for recent references).

Nowadays, several methods are implemented to measure well cosmic shear [4, 7, 10, 14, 15, 20, 26, 27, 28, 29]. Weak lensing being a shearing of the image of galaxies, it is natural to work on shapes of galaxies. Hence, a method completely based upon shapes should be ideal to deal with weak lensing. This is the original idea of the shapelets formalism, which relies on the decomposition of the image of galaxies into elementary shapes.

Our intend here is to introduce the shapelets formalism and method. The first chapter gives some useful features of the shapelets formalism. Chapter 2 presents the method and the IDL pipeline for decomposing a galaxy into shapelets (version 2.2) [21]; chapter 3 explains how to correct for a PSF. Finally, chapter 4 describes how we do visualization when dealing with shapelets.

We present a whole weak lensing measurement method, based upon the shapelets formalism, from detection of galaxies in an image, to the creation of a mass map, in Volume II.

Chapter 1

Shapelets formalism

For all this chapter, we refer the reader to Refregier 2003 [26], Refregier & Bacon 2003 [27], Massey & Refregier 2006 [20], Massey, Refregier & Bacon 2004 [22]

Shapelets are new mathematical entities [26, 27, 22] aimed at image analysis. They represent a complete, orthogonal, set of basis functions onto one can efficiently decompose every object, such as galaxies for example.

Although the shapelets formalism, because of its richness, goes beyond the the aim of this manual, it is necessary to get an overview of its basics. This is the intend of the present section.

1.1 Generalities

People who work on image processing may be familiar with wavelets, appeared in the mid-1980s. Since they allows one to decompose an object or an image on several scales, they are a performant tool to prospect all scales on an image. But they fail in describing the real shape of an object.

Instead, shapelets do not decompose objects on different scales, but on different elementary shapes. More precisely, the shapelets decomposition is a linear decomposition into a series of localised basis functions with different shapes, the shapelets. Two different types of shapelets exist, which are very similar. The *cartesian shapelets* are weighted Hermite polynomials, which correspond to perturbations about a circular gaussian and, in their asymptotic form, to the Edgeworth expansion in several dimensions. They are also the eigenstates of the 2-dimensional Quantum Harmonic Oscillator (QHO), and they allow to use the powerful formalism developed in quantum mechanics. The *polar shapelets* are weighted Laguerre polynomials, and are closely linked to the cartesian ones. Both kinds of shapelets have several interesting mathematical properties, some of which we will deal with hereafter.

1.2 Cartesian shapelets

1.2.1 One-dimensional cartesian shapelets

To describe a 1-dimensional object, we define the dimensionless basis functions

$$\phi_n(x) \equiv \left[2^n \sqrt{\pi} n!\right]^{-1/2} H_n(x) e^{-x^2/2} \tag{1.1}$$

in which $n \in \mathbb{N}^*$ and H_n is an nth order Hermite polynomial.

They are orthogonal, *i.e.* $\int_{-\infty}^{\infty} \mathrm{d}x \phi_n(x)\phi_m(x) = \delta_{mn}$, where δ_{mn} is the Kronecker symbol. They can be seen as perturbations around the gaussian ϕ_0. The first few basis functions ϕ_n are shown on figure 1.1 (a).

In practice, we rather use the dimensional basis functions

$$B_n(x;\beta) \equiv \beta^{-1/2} \phi_n\left(\frac{x}{\beta}\right) \tag{1.2}$$

where β is a characteristic size, close to that of the object to decompose. The basis functions B_n also are orthogonal. They create a complete set of basis functions for C^∞ and integrable functions, so as the profile $f(x)$ of an object can be decomposed on these :

$$f(x) = \sum_{n=0}^{\infty} f_n B_n(x;\beta) \tag{1.3}$$

where the shapelet coefficient is given by :

$$f_n = \int_{-\infty}^{\infty} \mathrm{d}x f(x) B_n(x;\beta) \tag{1.4}$$

The series converges quickly if the object is sufficiently localised, and if β and the origin $x = 0$ are close enough of the size and of the center of the object. In such a case, a decomposition onto the first few basis functions suffices to catch the shape information of the object of interest, and its decomposition into shapelets can be truncated to some maximum order of decomposition, denoted $n_{\max}$ hereafter. That is,

$$f(x) = \sum_{n=0}^{n_{\max}} f_n B_n(x;\beta) \tag{1.5}$$

is a good description of the object.

The functions ϕ_n and B_n have many interesting and useful properties : we refer the reader to [26] for details. Let us just note their invariance up to a rescaling under Fourier transform, $\tilde{\phi_n}(k) = i^n \phi_n(k)$, $\tilde{\phi_n}$ being the Fourier transform of ϕ_n.

A very useful property about the integration of B_n must be pointed out ; the shapelets basis functions obey the integral property :

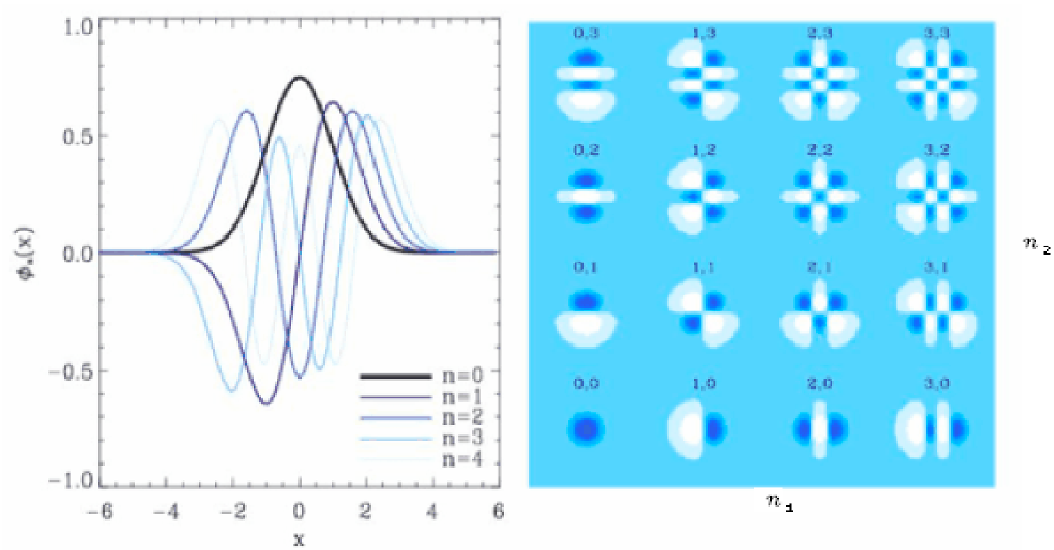

Figure 1.1: (a) First few one-dimensional basis functions $\phi_n(x)$. (b) First few two dimensional basis functions ϕ_{n_1,n_2}. The figure is from [26].

$$\int_{-\infty}^{\infty} \mathrm{d}x B_n(x;\beta) = \left[2^{1-n}\sqrt{\pi}\beta\right]^{1/2}\left(\mathrm{C}_n^{\frac{n}{2}}\right)^{1/2} \tag{1.6}$$

where $\mathrm{C}_\mathrm{n}^{\frac{\mathrm{n}}{2}}$ is the binomial coefficient ; note that this is true only for even n.

One can notice an analogy between shapelets and the quantum harmonic oscillator : shapelets basis functions are QHO's eigenstates. Hence, we can transpose quantum mechanics formalism to deal with shapelets. In particular, we introduce the creation and annihilation operators, defined respectively as :

$$\hat{a}^\dagger \equiv \frac{1}{\sqrt{2}}(\hat{x} - i\hat{p}), \quad \hat{a} \equiv \frac{1}{\sqrt{2}}(\hat{x} + i\hat{p}) \tag{1.7}$$

where $\hat{x} = x$ is the position operator and $\hat{p} = \frac{1}{i}\frac{\partial}{\partial x}$ is the momentum operator, and † is the hermitic conjugate.

The $\hat{a}$ and $\hat{a}^\dagger$ operators act on the basis functions as :

$$\hat{a}\phi_n = \sqrt{n}\phi_{n-1}, \quad \hat{a}^\dagger\phi_n = \sqrt{n+1}\phi_{n+1} \tag{1.8}$$

With these operators, we define the numeration operator $\hat{N} \equiv \hat{a}^\dagger\hat{a}$, which has the useful property

$$\hat{N}\phi_n = n\phi_n \tag{1.9}$$

and which we will use hereafter to compute the moments of an object.

1.2.2 Two-dimensional cartesian shapelets

The dimensionless two-dimensional shapelets are the product of the one-dimensional ones.

$$\phi_{\mathbf{n}}(\mathbf{x}) \equiv \phi_{n_1}(x_1)\phi_{n_2}(x_2) \tag{1.10}$$

where $\mathbf{x} = (x_1, x_2)$ and $\mathbf{n} = (n_1, n_2)$. The first few basis functions ϕ_{n_1,n_2} are shown on figure 1.1 (b).

The dimensioned basis functions are defined in an equivalent way :

$$B_{\mathbf{n}}(\mathbf{x}; \beta) \equiv \beta^{-1}\phi_{\mathbf{n}}\left(\frac{\mathbf{x}}{\beta}\right) \tag{1.11}$$

They are still orthonormal.

Decomposing an object into 2-dimensional shapelets is done in the same way than into 1-dimensional shapelets :

$$\begin{cases} f(\mathbf{x}) = \sum_{n_1,n_2=0}^{\infty} f_{\mathbf{n}} B_{\mathbf{n}}(\mathbf{x}; \beta) \\ f_{\mathbf{n}} = \int \mathrm{d}^2x f(\mathbf{x}) B_{\mathbf{n}}(\mathbf{x}; \beta) \end{cases} \tag{1.12}$$

where $f_{\mathbf{n}} = f_{n_1} f_{n_2}$.

Similarly to a one-dimensional shapelet decomposition, most of the shape information of a two-dimensional object is contained in the first few 2-dimensional shapelet basis functions. Hence, one can restrict the decomposition to orders less than a certain maximum order of decomposition, so as equation (1.12) becomes :

$$f(\mathbf{x}) = \sum_{n_1,n_2=0}^{n_1+n_2\leq n_{\max}} f_{\mathbf{n}} B_{\mathbf{n}}(\mathbf{x}; \beta) \tag{1.13}$$

Figure 1.2 (a) shows how a galaxy can be decomposed and reconstructed to different orders of decomposition ; note that as $\mathbf{n}$ increases, the reconstruction goes better. For big enough $\mathbf{n}$, the reconstruction is almost indistinguishabe from the original image. Figure 1.2 (b) shows the shapelets coefficients associated with the same galaxy ; in other words, it shows the previous galaxy in shapelets space.

1.2.3 Shape parameters associated with shapelets

Shapelets allow an accurate computation of the astrometry, the photometry and the shape parameters of the object to decompose, for example its moments.

Using equation (1.6), the total flux F is given by :

$$F = \int \mathrm{d}^2x f(x) = \sqrt{\pi}\beta \sum_{n_1,n_2}^{\text{even}} 2^{\frac{1}{2}(2-n_1-n_2)} \left(\mathrm{C}_{n_1}^{\frac{n_1}{2}}\right)^{1/2} \left(\mathrm{C}_{n_2}^{\frac{n_2}{2}}\right)^{1/2} f_{n_1,n_2} \tag{1.14}$$

where $f_{n_1,n_2} = f_{\mathbf{n}}$.

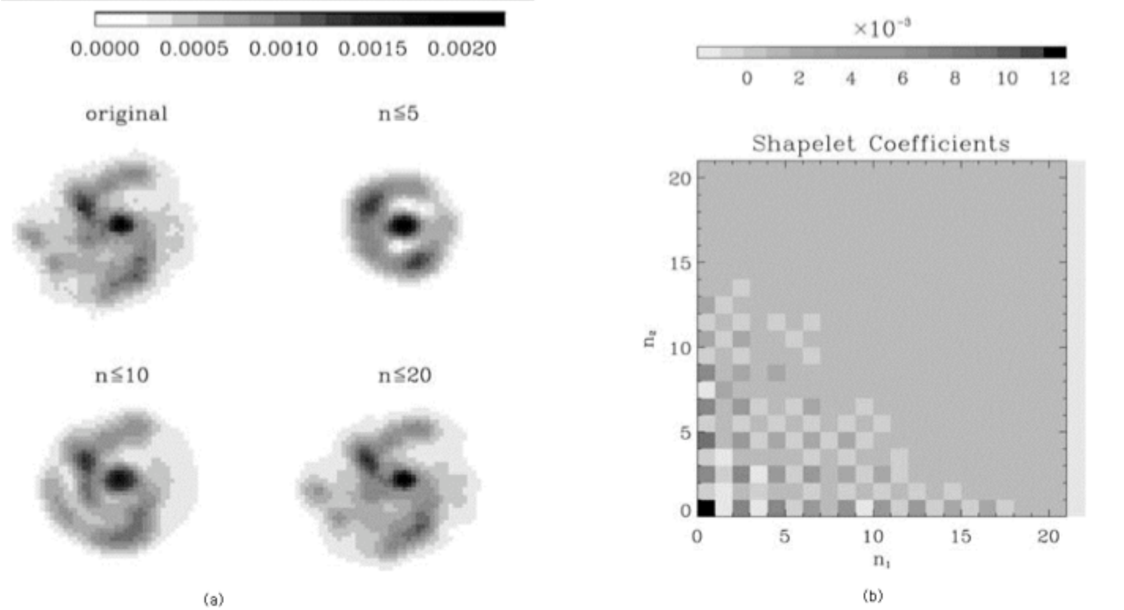

Figure 1.2: (a) A galaxy and its reconstruction after a shapelets decomposition at different **n** ; (b) the shapelets coefficients of the same galaxy. The figure is from [26].

Identically, and using creation and annihilation operators, it can be shown that the centroid of the object $x_1^f = \int \mathrm{d}^2 x_1 f(x)/F$ is given by :

$$x_1^f = F^{-1}\sqrt{\pi}\beta^2 \sum_{n_1}^{\text{odd}} \sum_{n_2}^{\text{even}} (n_1+1)^{1/2} 2^{(2-n_1-n_2)/2} \left(\mathrm{C}_{n_1+1}^{\frac{n_1+1}{2}}\right)^{1/2} \left(\mathrm{C}_{n_2}^{\frac{n_2}{2}}\right)^{1/2} f_{n_1,n_2} \tag{1.15}$$

and so on for x_2^f.

The quadrupoles $J_{\mathrm{ij}} = \int \mathrm{d}^2 x x_{\mathrm{i}} x_{\mathrm{j}} f(\mathrm{x})/F$ are obtained the same way. x_1 and x_2 being symmetric, the centroid expression makes it obvious that :

$$J_{12} = J_{21} = F^{-1}\sqrt{\pi}\beta^3 \sum_{n_1,n_2}^{\text{even}} (n_1+1)^{1/2}(n_2+1)^{1/2} 2^{(2-n_1-n_2)/2} \left(\mathrm{C}_{n_1+1}^{\frac{n_1+1}{2}}\right)^{1/2} \left(\mathrm{C}_{n_2+1}^{\frac{n_2+1}{2}}\right)^{1/2} f_{n_1,n_2} \tag{1.16}$$

Computing the other two quadrupoles demands to use the numeration operator. It gives :

$$J_{11} = F^{-1}\sqrt{\pi}\beta^3 \sum_{n_1,n_2}^{\text{even}} 2^{(2-n_1-n_2)/2}(1+2n_1) \left(\mathrm{C}_{n_1}^{\frac{n_1}{2}}\right)^{1/2} \left(\mathrm{C}_{n_2}^{\frac{n_2}{2}}\right)^{1/2} f_{n_1,n_2} \tag{1.17}$$

and so on for J_{22}.

Finally, one can calculate the size of the object :

$$R^2 = F^{-1}\sqrt{\pi}\beta^3 \sum_{n_1,n_2}^{\text{even}} 2^{(4-n_1-n_2)/2}(1+n_1+n_2)\left(\mathrm{C}_{n_1}^{\frac{n_1}{2}}\right)^{1/2}\left(\mathrm{C}_{n_2}^{\frac{n_2}{2}}\right)^{1/2} f_{n_1,n_2} \tag{1.18}$$

1.3 Polar shapelets

Polar shapelets have been introduced in [26], and fully described in [20].

1.3.1 Formalism

They are directly linked to cartesian shapelets, and share all their useful properties, and a similar gaussian weighting function of scale size β. But as they are separable in r and θ, polar shapelets coefficients are easier to understand in terms of rotational symmetries, and many operations are made simpler and more intuitive.

A function $f(r,\theta)$ in polar coordinates can be decomposed as a weighted sum of the basis functions $\chi_{\mathrm{n,m}}(r,\theta;\beta)$:

$$\begin{cases} f(r,\theta) = \sum_{\mathrm{n}=0}^{\infty}\sum_{\mathrm{m}=-\mathrm{n}}^{\mathrm{n}} f_{\mathrm{n,m}}\chi_{\mathrm{n,m}}(r,\theta;\beta) \\ \\ f_{\mathrm{n,m}} = \iint_{\mathbb{R}} f(r,\theta)\chi_{\mathrm{n,m}}(r,\theta;\beta) r\mathrm{d}r\mathrm{d}\theta \end{cases} \tag{1.19}$$

$f_{\mathrm{n,m}}$ $(n \in \mathbb{N},\ -n \leq m \leq n)$ are the polar shapelets coefficients of order (n,m).

The basis functions $\chi_{\mathrm{n,m}}$ are related to Laguerre polynomials $L_{(n-|m|)/2}^{|m|}$ through (note that only the states with n and m both even or odd are allowed) :

$$\chi_{n,m}(r,\theta;\beta) = \frac{(-1)^{(n-|m|)/2}}{\beta^{|m|+1}}\left\{\frac{[(n-|m|)/2]!}{\pi[(n+|m|)/2]!}\right\}^{1/2} r^{|m|} L_{(n-|m|)/2}^{|m|}\left(\frac{r^2}{\beta^2}\right) e^{-r^2/2\beta^2} e^{-\mathrm{i}m\theta} \tag{1.20}$$

The real and imaginary parts of the first few polar shapelets basis functions are shown on Figure (1.3). The basis functions with $m=0$ are wholly real.

As polar shapelets basis functions are complex number, the shapelets coefficients $f_{n,m}$ are complex number. Their moduli (shown on Figure (1.4)) determine the strength, and their phase the orientation, of a component.

Like in the case of cartesian shapelets, most of the shape information is caught by the first few shapelets basis functions. This allows us to truncate the decomposition (1.19) to some maximum order of decomposition $n_{\max}$, such as :

$$f(r,\theta) = \sum_{\mathrm{n}=0}^{n_{\max}}\sum_{\mathrm{m}=-\mathrm{n}}^{\mathrm{n}} f_{\mathrm{n,m}}\chi_{\mathrm{n,m}}(r,\theta;\beta) \tag{1.21}$$

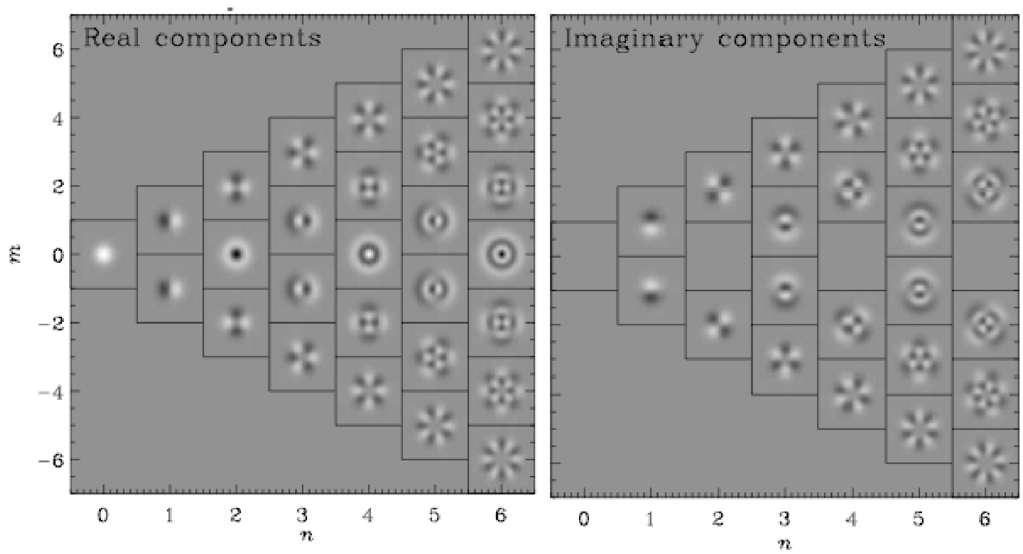

Figure 1.3: First few polar shapelets basis functions $\chi_{n,m}$.

Moreover, as clearly visible on Figure (1.3), basis functions of which (n, m) verify the condition $n + |m| = \text{cste}$ all have the same scale of oscillation, *i.e.* they describe same sized details of the object. Instead, for some fixed n, the scale of oscillation of basis functions decreases with increasing $|m|$. It then seems natural not to use a simple "$n_{\max}$" truncation as in Equation (1.21), but to use a "diamond" criterium $n+|m| \leq n_{\max}$ for truncating the shapelets decomposition. The shapelets coefficients considered in such a decomposition,

$$f(r, \theta) = \sum_{\mathrm{n}=0}^{n_{\max}} \sum_{|m| \leq n_{\max}-n} f_{\mathrm{n,m}} \chi_{\mathrm{n,m}}(r, \theta; \beta) \tag{1.22}$$

are enclosed within the diamond symbol depicted by the solid lines on Figure (1.4). Besides its natural definition, this truncation schemes allows to further compress the decomposition, by getting rid of inappropriate, with too much oscillations, basis functions.

1.3.2 Shape parameters

One can use polar shapelets to compute the photometry and astrometry of objects. Here we give some formulae. Their demonstration can be found in [20].

The flux is given by

$$F = \iint_{\mathbb{R}} f(x) \mathrm{d}^2 x = \beta \sqrt{4\pi} \sum_{n}^{\text{even}} f_{n,0} \tag{1.23}$$

A centroid (x_c, y_c) is given by

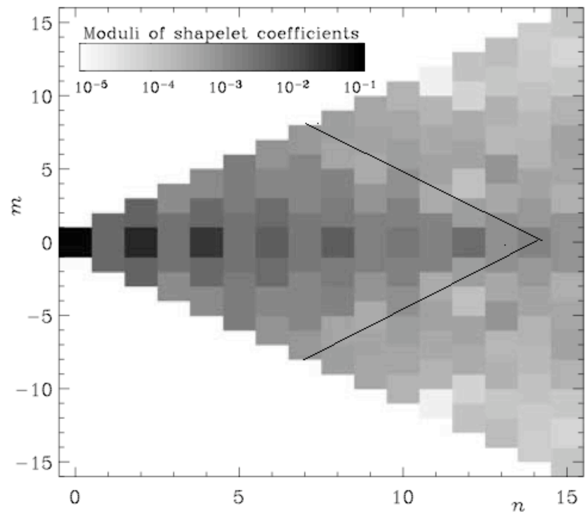

Figure 1.4: Shapelets coefficients of the decomposition of a galaxy.

$$x_c + \mathrm{i}y_c = \frac{\sqrt{8\pi}\beta^2}{F}\sum_n^{\text{odd}}\sqrt{n+1}f_{n,1} \tag{1.24}$$

The size and the ellipticity are derived from the unweighted quadrupoles moments :

$$R^2 = \frac{\sqrt{16\pi}\beta^3}{F}\sum_n^{\text{even}}(n+1)f_{n,0} \tag{1.25}$$

and

$$\varepsilon = \frac{\sqrt{16\pi}\beta^3}{FR^2}\sum_n^{\text{even}}\sqrt{n(n+2)}f_{n,2} \tag{1.26}$$

with $\varepsilon = |e|\cos 2\theta + \mathrm{i}\sin 2\theta$.

1.3.3 Conversion polar-cartesian shapelets

Although polar shapelets are complex functions and cartesian shapelets are real functions, it is possible to convert a set of cartesian shapelets coefficients f_{n_1,n_2} with $n_1 + n_2 \leq n_{\max}$ into polar shapelets coefficients with $n \leq n_{\max}$, with a direct one-to-one mapping, through the relation [20] :

$$\begin{aligned} f_{n,m} = 2^{-n/2}\mathrm{i}^m \left\{\tfrac{n_1!n_2!}{[(n+m)/2]![(n-m)/2]!}\right\}^{1/2} \delta_{n_1+n_2,n} \\ \times \textstyle\sum_{n_r'=0}^{n_r}\sum_{n_l'}^{n_l} \mathrm{i}^{m'}\mathrm{C}_{(n+m)/2}^{n_r'}\mathrm{C}_{(n-m)/2}^{n_l'}\delta_{n_r'+n_l',n_1}f_{n_1,n_2} \end{aligned} \tag{1.27}$$

1.4 Scales of decomposition

It must be noted from figure 1.1 (b) that as $\mathbf{n} = n_1 + n_2$ increases (with fixed β), the cartesian $B_{\mathbf{n}}$ basis functions acquire both a larger extend and smaller scale oscillations. The same trend is visible for polar $\chi_{n,m}$ basis functions, for increasing $n + |m|$, as shown by figure (1.3). That implies that a decomposition into shapelets done with certain β and $n_{\max}$ will describe only objects of size within a certain range. One can show that the features described by such a decomposition are of size ranging between [26]

$$\theta_{\min} \approx \beta(n_{\max}+1)^{-\frac{1}{2}} \quad \text{and} \quad \theta_{\max} \approx \beta(n_{\max}+1)^{\frac{1}{2}}. \tag{1.28}$$

This can be applied to choose the optimum β and $n_{\max}$ for a shapelets decomposition, as will be described in Volume II.

1.5 Convolution and deconvolution

Let f and g be two functions, and h their convolution product ; α, β and γ are their respective characteristic scales. The convolution product reads :

$$h(\mathbf{x}) = (f * g)(\mathbf{x}) = \int \mathrm{d}^2 x f(\mathbf{x}-\mathbf{x}')g(\mathbf{x}') \tag{1.29}$$

One can link the coefficients of the shapelets decomposition of the three functions, thanks to a convolution tensor $C_{\mathbf{nml}}$ function of the characteristic scales α, β and γ:

$$h_{\mathbf{n}} = \sum_{\mathbf{m},\mathbf{l}} C_{\mathbf{nml}} f_{\mathbf{m}} g_{\mathbf{l}} \tag{1.30}$$

It can be shown that the tensor $C_{\mathbf{nml}}$ depends on the shapelets basis functions B_n, B_m and B_l. So the shapelets formalism allows one to analytically evaluate the convolution product of two functions, by decomposing them into shapelets.

As a consequence, it is possible to implement a PSF deconvolution method entirely based upon shapelets formalism. Assume that one knows the PSF $g(\mathbf{x})$, convolved with f. Writing $P_{\mathbf{nm}} \equiv \sum_{\mathbf{l}} C_{\mathbf{nml}} g_{\mathbf{l}}$ the PSF matrix , equation (1.30) becomes:

$$h_{\mathbf{n}} = \sum_{\mathbf{m}} P_{\mathbf{nm}} f_{\mathbf{m}} \tag{1.31}$$

It can be shown that, to low enough order, the PSF matrix is invertible, which allows us to deconvolve the PSF and to obtain the wanted coefficients $f_{\mathbf{m}}$:

$$f_{\mathbf{m}} = P^{-1}_{\mathbf{nm}} h_{\mathrm{n}} \tag{1.32}$$

and so, to obtain the deconvolved profile f of the PSF.

In practice, we do not use the scheme presented above, in order to avoid numerical instabilities while inverting the matrix $P_{\mathbf{nm}}$. Instead, we create an *a priori* unconvolved model of galaxies, that we convolve (Eq. 1.30) with a previously measured model of the PSF. This deconvolution scheme will be further developped in chapter 3.

1.6 Pixelization

Real data are usually stored into pixels. Instead, shapelets provide an analytical decomposition of objects. Therefore, to eventually match the continuous analytical decomposition and the discrete data object, we pixelize our analytical shapelets model. Although some caution must be taken in pixelizing an analytical model [20], we make use of the fact that cartesian shapelets basis function are separable in x and y, and can be analytically integrated within rectangular pixels. This is similar to what really happens to photons hitting a CCD, which are integrated in pixels.

Let the pixel in which we want to integrate our model be enclosed in the region $(x \in [a_1, b_1], y \in [a_2, b_2])$; one can show that if there is no missing border, due for instance to electronics, the integration is given by [20] :

$$I_{n_1,n_2} = \int_{a_1}^{b_1} \int_{a_2}^{b_2} \phi_{n_1}(x)\phi_{n_2}(y)\mathrm{d}x\mathrm{d}y = I_{n_1} I_{n_2} \tag{1.33}$$

where the integrals $I_n = \int_a^b \phi_n(x)\mathrm{d}x$ are given by the recurrence relation :

$$\begin{cases} I_0 = \sqrt{\frac{\beta\pi^{1/2}}{2}}[\mathrm{erf}(x)]_a^b \\ I_1 = -\sqrt{2\beta}[\phi_0(x)]_a^b \\ I_n = -\beta\sqrt{\frac{2}{n}}[\phi_{n-1}(x)]_a^b + \sqrt{\frac{n-1}{n}} I_{n-2}, \;\; n \geq 2 \end{cases} \tag{1.34}$$

Note that missing pixel borders can be dealt with by altering the limits of the integration.

Finally, note that when using polar shapelets, which cannot be integrated in rectangular pixels, we first convert the model into cartesian shapelets, then convert the pixelised cartesian model in polar shapelets.

As a result, the unconvolved analytical shapelets model, after being convolved with the PSF, has been pixelized. It therefore undergoes the same operations as real photons through an observation. Thus, it can be directly matched to data, so as to perform a least-square fitting on our model to make it reliable. It is then possible to measure the shape of the model, and to infer a shear estimator.

1.7 A shear estimator (for cartesian shapelets)

Shapelets were primarily defined for weak lensing dedicated image analysis. As a result, they naturally permit to estimate a shear estimator based upon the introduction of a new quantic operator $\hat{S}_i$, the shear operator, which acts on the profile of the object as shown :

$$f' \simeq (1 + \gamma_i \hat{S}_i) f \tag{1.35}$$

at the first order in γ_i, γ_i being the shear applicated to the object ; f is the 'unsheared' profile of the object, f' is its sheared profile.

The operator $\hat{S}_i$ depends on the creation and annihilation operators, and it permits to define the shear matrix (see [27] for details) :

$$S_{i\mathbf{mn}} \equiv \int \mathrm{d}^2 x B_{\mathbf{n}}(\mathbf{x}) \hat{S}_i B_{\mathbf{m}}(\mathbf{x}) \tag{1.36}$$

It is then possible, thanks to some properties of the shear in the shapelets space, to obtain an unbiased estimator for each component of the shear (see [27] for details):

$$\begin{cases} \hat{\gamma}_{1\mathbf{n}} &= \frac{f'_{\mathbf{n}} - <f_{\mathbf{n}}>}{S_{1\mathbf{nm}<f_{\mathbf{m}}>}}, \quad n_1, n_2 \,\text{pairs} \\ \hat{\gamma}_{2\mathbf{n}} &= \frac{f'_{\mathbf{n}} - <f_{\mathbf{n}}>}{S_{2\mathbf{nm}<f_{\mathbf{m}}>}}, \quad n_1, n_2 \,\text{odd} \end{cases} \tag{1.37}$$

The following chapter describes the publicly available IDL package we implemented to use shapelets for image processing.

Chapter 2

Shapelets decomposition

The "shapelets package" is the publicly available IDL software which contains the shapelets formalism, *i.e.* which deals with decomposition into shapelets, and all the shapelets algebra (convolution, deconvolution, rotation, dilatation, etc.). This section is aimed at introducing the most useful routines for the shear analysis of astronomical images. Due to the package's richness, it cannot be an exhaustive explanation of all routines gathered therein. Nevertheless, each one is sufficiently explained in its header so as the user can easily understand the way it works. The purpose of each routine can also be found in appendix A.

2.1 Detect objects to decompose

A prior to a shapelets-based weak lensing measurement is the detection of all galaxies to be decomposed. This is done with SExtractor [5, 6, 13], and is described in Volume II.

Once all objects have been detected, we perform a shapelets decomposition on each object idividually, one after the other.

2.2 Isolate the object to decompose

The first step in a shapelets decomposition is to isolate the object from the SExtractor catalogue containing the all detected objects. The goal is to cut the wanted object and its near neighborhood out of the image ; in other words, one wants to extract a postage stamp of the image, illustrated by figure 2.1. This is done thanks to a function called shapelets_sexcat2pstamp.pro, which we present in the next subsection.

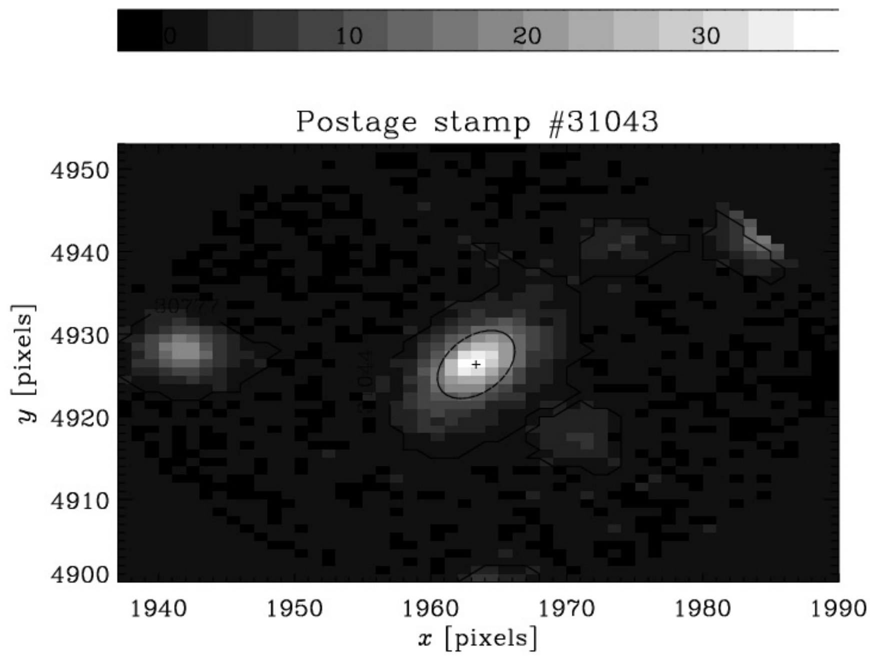

Figure 2.1: A postage stamp, isolating a galaxy.

2.2.1 The shapelets_sexcat2pstamp routine

As its name suggests, this routine extracts postage stamps of the SExtractor catalogue.

In order to create a postage stamp named pstamp, call it as :

pstamp=shapelets_sexcat2pstamp(*file_name,image,sexcat,id,options*

using the arguments :

- *image* is an image structure, given by shapelets_read_image [1].
- *sexcat* is the SExtractor catalogue out of which one wants to isolate pstamp.
- *id* is the identity number of the object to place in the postage stamp.
- /NOISE_MAP : if set, a noise map from *image* is used ; otherwise, the background is locally estimated (a noise map is created).
- /SEG_MAP : if set, a locally determined segmentation map is used; otherwise, the one from SExtractor (stored in *image*) is used.
- SATURATION_LEVEL=saturation_level : not used.

[1] see section 4.1.1

- /SQUARE : if set, the postage stamp will be squared ; otherwise, it will be circular. If /SQUARE is not set, the routine starts with creating a square postage stamp, then masks the corner off so as it becomes a circle.
- /BORDER : if set, the border around the edges of postage stamp is increased, to make pretty plots in papers but it is useless elsewhere.
- BACK_SIZE=back_size : is the size of postage stamp used for noise estimation ; default value is 120 pixels.
- N_GROW=n_grow : is the number of times SExtractor's segmentation map is grown to mask objects during noise estimation.
- /VARY_LOCAL_NOISE_CALC : allows to estimate the noise by a simple iteration on the statistics of the background.
- /SHOT_NOISE : allows to add the photon noise in the noise map.
- /LAZY : if it is set, then if a deffect is found in the surrounding of the object (in such a case, the object will eventually be eliminated), the noise map is not computed, but a fake is created instead.
- NFWHM=nfwhm is the size of the postage stamp, in units of the a parameter of SExtractor (default = 3).
- NEIGHBOUR=neighbour : this keyword is needed for the treatment of neighbours (which pollute the postage stamp) ; if neighbour=0 (default), the errors will be infinite *i.e.* the fit is unconstrained ; if neighbour=1, neighbour pixels are set to the background level.
- N_PIXELS=n_pixels : (output) is the size of the postage stamp.
- /PLOTIT : if set, the postage stamp is ploted to the screen.
- /SILENT : if set, the routine acts silently.

In practice, shapelets_sexcat2pstamp guesses the size and the coordinates (edges) of the postage stamp, from the position and the FWHM of the object given by SExtractor. It then checks if there are no problems (*e.g.* the object can be near an edge of the image, or can be masked out...), before extracting the desired region (pstamp), and creating segmentation and noise maps. Finally, neighbors are removed from the postage stamp, so as the desired object can be alone, and all useful quantities are stored into the pstamp structure.

2.2.2 Creation of a segmentation map

The segmentation map shows the position and shape of each detected object, represented by its ID. SExtractor can create one, that can be read later. Thus, if the /SEG_MAP keyword is not set, shapelets_sexcat2pstamp creates the segmentation map of the postage stamp only by extracting the region of the postage stamp off the SExtractor segmentation map (of the whole image).

Otherwise, if the /SEG_MAP is set, shapelets_sexcat2pstamp creates a brand new segmentation map within the postage stamp. To achieve this, the object's neighbors within the postage stamp are detected (besides the object itself), their spatial extension determined, and their potential overlap with the object to decompose checked. The creation of the segmentation map is finalised by giving to each object's pixels the value of (its ID)+1 (to follow SExtractor convention) ; background pixels' (which do not belong to any object) value is set to 0.

2.2.3 Noise estimation and creation of a noise map

One crucial task is to estimate the noise, which will be needed later for the computation and minimization of the least-square χ^2 during the focus process (see section 2.3). Doing so, we create a noise map of the postage stamp, which is an inverse variance map of the image. Two methods can be used, either the /NOISE_MAP keyword is set or not. The first step in both methods is to discriminate background pixels against object pixels, using the segmentation map (as background pixels have zero segmentation value).

If /NOISE_MAP is set, we do an external estimation of the noise : we use the noise already estimated when reading the image with shapelets_read_image [2]. In the mean time, the rms of the noise is computed,

$$\text{back\_rms\_ext} = \frac{1}{\sqrt{\text{noise\_ext}}} \tag{2.1}$$

where we use the notations used in the routine ; the "ext" stands for "external" estimation of the noise. Note that this estimation method can be used only if the rms is strictly positive.

If /NOISE_MAP is not set, we do a local estimation of the noise : we compute the noise without taking the SExtractor noise estimation into account. Here, one can choose between two different methods to compute the noise.

If /VARY_LOCAL_NOISE_CALC is set, we iterate on the estimation of the mean, the mode [3] and the rms of the background until the variation of the rms between two iterations is less than 1 %. When this is the case, we create the noise map by giving each of its pixels the value :

$$\text{noise\_local} = \frac{1}{\text{back\_rms\_local}^2} \tag{2.2}$$

Note that this noise map is an inverse variance of the background map.

[2]see section 4.1

[3]the mode is computed as being : mode = 2.5 median - 1.5 mean

If /VARY_LOCAL_NOISE_CALC is not set, we first extract an area larger than the postage stamp in order to have more background pixels (which improves the statistics and, thus, the estimation). We then grow the region of interest to mask nearby pixels. The mean and rms of the background are finally computed (if there are enough background pixels, no fatal default, and not too much zero pixels in the postage stamp) using the sky.pro routine (which computes the sky level in an image, and is available in the astrolib package [17]). Here again, we create the noise map by giving each of its pixels the value :

$$\text{noise\_local} = \frac{1}{\text{back\_rms\_local}^2} \tag{2.3}$$

In equations (2.2) and (2.3), the notations are the same as in the routine ; the "local" stands for "local" estimation of the noise.

Note that the noise we compute is gaussian and uncorrelated. To be fair, this might decrease the efficiency of a shapelets decomposition, especially if the intrinsic data noise is correlated, and/or non-gaussian. This is an aspect we still have to work on...

2.2.4 The pstamp structure

The pstamp structure is the output of shapelets_sexcat2pstamp, it contains all useful quantities for the postage stamp and the extracted object. Among them :

1. the image in the postage stamp
2. the object's position, in the image coordinates, as given by SExtractor
3. the coordinates of the corners and edges of the postage stamp (the preliminary square one if it is circular)
4. the object's centroid in pstamp coordinates
5. the noise and segmentation maps
6. the estimated noises's mean and rms
7. some SExtractor quantities
8. a flag giving the possible errors

The flag in pstamp

This flag is aimed at further detecting possible bad objects. It consists in a number between 0 and 10. For example, an object can lie near the object of interest (flag = 1), or the object can be masked out (flag = 6), or the background could not be estimated (flag = 8).

Possible flags and their meaning are given in the left column of table (2.1).

Eventually, in all shapelets decomposition, all objects with postage stamp flag greater than 3 will be rejected.

The pstamp structure is then the seed of all the decomposition process, which must begin by finding optimal parameters for the decomposition into shapelets.

2.3 Find the optimal parameters for the decomposition

When an object is isolated in a postage stamp, and even before decomposing [4] it into shapelets, one must determine the parameters that will define it : its scale parameter β, its centroid x_c (*i.e.* the center of the basis functions), its maximum order of decomposition $n_{\max}$. Indeed, while β and x_0 are well fixed, the object shape information is contained within only the first few shapelet coefficients, and that allows to truncate the expansion at $n_{\max}$. A crucial but very delicate part of the shapelets decomposition is to focus these three parameters.

To achieve this, we search the best β, x_0 and $n_{\max}$ which minimize the difference between the observed and reconstructed images, renormalized with respect to the local noise level,

$$\chi^2 = \frac{1}{\text{number of pixels}} \sum_{\text{pixels}} \frac{(I^{\text{obs}}(\vec{x_p}) - I^{\text{rec}}(\vec{x_p}))^2}{\sigma_p^2} \tag{2.4}$$

Doing so, one must note that a decomposition into shapelets done with certain β and $n_{\max}$ will describe objects of size within a certain range (eq. 1.28) [26]. Thus, we focus β, x_0 and $n_{\max}$ by changing them until χ^2 is minimized, and putting geometrical constraints. For instance, the size of the eventual reconstructed image must be larger than the seeing of the image, and less than the size of the postage stamp, and lie between $\theta_{\min}$ and $\theta_{\max}$ (equation (1.28)). If this is not the case, we search other (better) β, x_0 and $n_{\max}$.

Figure 2.2 shows this process : β and $n_{\max}$ are varied along the solid black line, labeled "Algorithm", until the reduced χ^2 (shown in color) is $1 \pm \epsilon$. During the path of the algorithm, β and $n_{\max}$ must be such as they lie between the dashed lines labeled $\theta_{\min}$ and $\theta_{\max}$.

[4] in practice, focus and decomposition are simultaneous

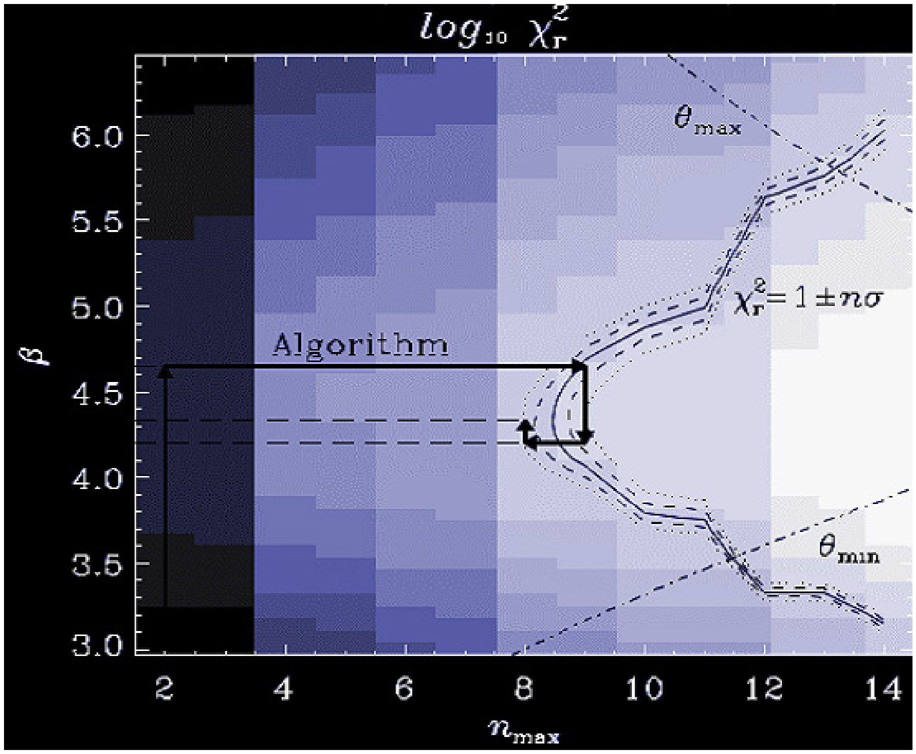

Figure 2.2: Focus of β and $n_{\max}$. They are varied along the solid black line, labeled "Algorithm", until the reduced χ^2 (shown in color) is $1 \pm n\sigma$. The "Algorithm" line must lie between the dashed lines labeled $\theta_{\min}$ and $\theta_{\max}$.

The minimisation of χ^2 is done using the Numerical Recipes routine Amoeba [24], re-written in idl langage.

Practically, this focus work is achieved in two times. Firstly, we focus x_0 and β, and secondly, $n_{\max}$ is fixed. The former task is done through the shapelets_focus_beta.pro routine, the latter by shapelets_focus_nmax.pro. But first of all, we must guess with which β, center, and $n_{\max}$ we shall begin the focus process.

2.3.1 Guess starting β, x_0 and $n_{\max}$

As we saw in chapter 1, section 1.2.2, shapelets scale β and maximum order of decomposition $n_{\max}$ can represent objects with sizes between certain $\theta_{\min}$ and $\theta_{\max}$ given by equation (1.28). Consequently, we choose starting β and $n_{\max}$[5] using these rules.

A routine called shapelets_geometric_constraints computes $\theta_{\min}$ (given by the user, or set to 0.2 pixels by default) and $\theta_{\max}$ from the size of the postage stamp. We then define a minimum and a maximum β as being :

[5]in practice, we always use $n_{\max} = 2$ as a starting point for focus

$$\beta_{\max} = \frac{\theta_{\max}}{\sqrt{n_{\max}+1}} \quad , \quad \beta_{\min} = \min(\theta_{\min}\sqrt{n_{\max}+1}, \beta_{\max}) \tag{2.5}$$

and set β to be $\min(\max(\beta_{\min}, \mathrm{FWHM}/2.86), \beta_{\max})$, where FWHM is the FWHM of the object, given by SExtractor. The guessed centre of the basis functions x_0 is set to be the center of the postage stamp. Those two operations are made by the routine shapelets_guess_nmaxbeta, the arguments of which we do not describe here.

2.3.2 Focus β and x_0

The shapelets_focus_beta.pro routine allows one to find the best scale parameter and centroid for the shapelets decomposition, assuming that the maximum order of decomposition $n_{\max}$ is known. It uses β and x_0 as guessed by shapelets_guess_-nmaxbeta (unless starting β and $n_{\max}$ are given by the user) as starting points, then it iterates on them with Amoeba till χ^2 is minimised, keeping all steps in a history structure, so as one can check later how the focus has gone. It decomposes the object with these best parameters, and outputs a focus structure containing notably β, x_0, $n\_\max$ (not focussed yet), χ^2 and the history structure.

As in practice shapelets_focus_beta.pro is never directly called by the user, it is not worth detailing its arguments here (just let us say that they are sensibly identical to those of shapelets_focus - see below). Its code is sufficiently commented so as one can easily use it if needed.

2.3.3 Focus $n_{\max}$

The shapelets_focus_nmax.pro allows one to find the best maximum order of decomposition $n_{\max}$ for the shapelets decomposition, assuming that the scale parameter β is known. It looks for the best $n_{\max}$ by exploring a range of possible $n_{\max}$ until the criterion $\chi^2 = 1 + \epsilon$ is reached (ϵ is a tolerance value, it must negligible compared to 1). The range of checked $n_{\max}$ depending on the size of the postage stamp, we can compute it and restrict the search for $n_{\max}$ at this range. Then, we try several allowed $n_{\max}$, starting with the lowest, and incrementing it of 2 for each step. The χ^2 parameter is computed by the shapelets_decomp.pro routine [6] and compared to 1. Comparing the two neighbors $n_{\max}$, we can decide which one gives the best result. If they are the same, we define the good $n_{\max}$ as the mean of the two possible $n_{\max}$. Iterating within the range of allowed $n_{\max}$, the routine finally outputs the focussed $n_{\max}$, stored in a focus structure. The history of the search is also kept in memory. Note that this focus structure is the same as the one output by shapelets_focus_beta.pro, but updated with the focussed $n_{\max}$.

As in practice shapelets_focus_nmax.pro is never directly called by the user, it is not worth detailing its arguments here (just let us say that they are sensibly

[6] see section 2.4

identical to those of shapelets_focus - see below). Its code is sufficiently commented so as one can easily use it if needed.

2.3.4 The shapelets_focus routine

In practice, the scale parameter, the centroid and the maximum order of decomposition are focussed with the shapelets_focus.pro routine. Of course, it relies on shapelets_focus_beta and shapelets_focus_nmax. After guessing the starting β and $n_{\max}$, it passes them to shapelets_focus_beta, which searches for the best β and x_0 for the guessed $n_{\max}$, then computes the best $n_{\max}$ with these β and x_0. Some iterations are made in which those two routines run one after each other, so as the three searched parameters are focussed the best way possible.

While performing the focus of β, x_0 and $n_{\max}$, the routine decomposes the object into shapelets, and outputs a decomp structure, calling shapelets_decomp.pro [7]. A focus structure is also output (see below).

Call shapelets_focus.pro as :

```
decomp=shapelets_focus(pstamp, options)
```

using the arguments :

- *pstamp* : the (input) postage stamp containing the object for which β, x_0 and $n_{\max}$ must be focussed.
- FOCUS=focus : the (output) focus structure (see below for details).
- RECOMP=recomp : the (optional output) recomposed image (see section 2.4 for details).
- BETA_GUESS=beta_guess : the starting point for β ; if not set, the routine guesses itself.
- BETA_TOLERANCE=beta_tolerance : the fractional tolerance for finding β. The default value is 10^{-3}.
- FIXED_BETA=fixed_beta : fixed value of beta that should be used for decomposition.
- N_MAX_GUESS=n_max_guess : the starting point for $n_{\max}$; if not set, the routine guesses itself.
- N_MAX_RANGE=n_max_range : range of allowed and tried values for n_max ; default is [2,20].
- CENTRE_GUESS=centre_guess : the starting point for the centre of shapelets basis functions.

[7] see section 2.4

- CHISQ_TARGET=chisq_target : the target value of χ^2 to achieve; default value is 1.
- CHISQ_TOLERANCE=chisq_tolerance : the fractional tolerance for χ^2 minimisation (called ϵ in the main text).
- CHISQ_FLATNESS=chisq_flatness : is the minimum difference in χ^2 between two decompositions with $n_{\max}$ differing by two to trigger the flatness constraint in shapelets_focus_nmax.
- THETA_MIN_GEOM=theta_min_geom : minimum geometrical scale $\theta_{\min}$; default value is 0.5 pixels.
- /GAUSSIAN_RECENTRING : if it is set, the basis functions are recentered on the previous decomposition's centre of light.
- MAX_LOOPS=max_loops : maximum number of alternative runs of shapelets_focus_beta and shapelets_focus_nmax ; default value is 20.
- /FULL_FOCUS : if it is set, all attempts made at decomposition during iteration are recorded.
- PSF=psf : to be set when the focus is done while deconvolving the PSF [8].
- SKY=sky : to be set when subtracting the background ; to be put to :
  - 1 to fit it with a constant value around the object
  - 2 to fit it with a plane around the object
- /NON1 : if set, shapelets coefficients with $n = 1$ are forced to be zero.
- /POLAR : to be set to use polar shapelets ; by default, cartesian shapelets are used.
- /DIAMOND : to be set to use the diamond truncation scheme for decomposition [9]
- /VERBOSE : if set, the routine operates noisily.
- /SILENT : if set, the routine operates silently.

Here, "decomp" is the output decomp structure given by shapelets_decomp.pro (see section 2.4 for details).

[8] see chapter 3 section 3.2
[9] see chapter 1, section 1.3

2.3.5 Possible problems and the focus flag

Problems can occur both during the focus of β and $n_{\max}$. They are identified with a flag, which consists in a number between 0 and 10. Note that the bigger the flag, the worse the error. If several errors have occurred, just the worst is written in the flag. Here, we list the possible problems, together with their representative flag.

- β can bounce off geometrical constraint (*i.e.* become less than the minimum allowed scale or greater than the maximum one) ; in that case, flag = 1.
- While deconvolving a PSF, the least-square fitting matrix[10] can become singular, if the convolved $n_{\max}$ is lower than the $n_{\max}$ used for focus : in that case, flag = 2.
- Two successive iterations can have with very similar χ^2 ; in this case, χ^2 is "flat", and it represents a problem in the focus process : flag = 3.
- In general, χ^2 should be monotonic in $n_{\max}$, that is, when increasing $n_{\max}$, the model should become better and χ^2 lower ; for some objects, it is not the case, and χ^2 becomes worse : then, flag = 5.
- amoeba can be unable to converge (flag = 6), or converge towards a bad χ^2 (flag = 7).
- $n_{\max}$ can reach the maximum $n_{\max}$ allowed for focus ; in that case, flag = 8.
- The focus on the center of the basis function can make it wander towards the edge of the postage stamp ; hence, basis functions are pushed off the postage stamp : flag = 9.
- The centre of basis functions can be pushed off the postage stamp, or the focus process can simply crash : flag = 10.

Flags and their meaning are listed in the right column of table (2.1).

Eventually, in all shapelets decomposition, all objects with focus flag greater than 8 will be rejected.

2.3.6 The focus structure

The focus structure is output by shapelets_focus, it contains all useful characteristics of the focus. Among them :

1. the focussed β, x_0 and $n_{\max}$.
2. the firstly guessed β, x_0 and $n_{\max}$.

[10] see chapter 3 section 3.2

	postage stamp	**focus**
0	OK	OK
1	Nearby object	Bounced off geometrical constraints
2	Severe overlapping with nearby object	Entered the region where the least-squares fitting matrix may be singular
3	Object is near a saturated pixel	Converged by flatness limit *
4	Object is near a masked region	Not used
5	Object is near the edge of the image	χ^2 is not monotonic in $n_{\max}$ *
6	Object is itself masked out	Amoeba dithered about and not converged to target χ^2 *
7	Object has 0 FWHM	Focus iteration did not converge to target χ^2 *
8	Too few background pixels around object	Maximum $n_{\max}$ reached *
9	Object entirely overlapped by neighbors	Centroid wandered, pushing basis functions off the edge of the pstamp
10	Routine sexcat2pstamp crashed	Fatal crash during focus routine

Table 2.1: Meaning of postage stamp and focus flags ; focus flags with a * imply that shapelets may incompletely represent the object.

3. the minimised χ^2.
4. the history of focus.
5. the flag giving which errors might have occurred during the focus process.

We have just seen that the focus procedure is intimately linked to the decomposition one. The next section describes how to decompose an object into shapelets.

2.4 Decompose an object into shapelets

Once an object has been isolated in a postage stamp, and its β, x_0 and $n_{\max}$ parameters focussed, one can decompose it into shapelets. This section is aimed at introducing the method of decomposition into shapelets, and presenting the routine dedicated to this task.

2.4.1 How to decompose into shapelets ?

The decomposition into shapelets is nothing more than fitting a model on data. That is, we create a shapelets model of a certain galaxy, that we fit (after possible convolution with a PSF and pixelization) to the galaxy we want to model. Technically, two methods can be used : the first one uses a least-squares fitting method, the other one is a normal linear overlap integral method. As the default's one is the least-squares fitting, hereafter we only describe this method.

The needed least-squares matrix (which gathers the shapelets basis functions needed for the fit) is used as prescribed by [18] to compute the shapelets coefficients of the fitting model of the galaxy. That is, the galaxy has been decomposed into shapelets.

2.4.2 In practice : shapelets_decomp.pro

The shapelets_decomp.pro routine is aimed at decomposing all kind of function into shapelets. In practice, it decomposes the intensity of galaxies or stars. It should always use the scale parameter and the maximum order of decomposition as found by the focus process. That is why we call this routine only within the shapelets_focus.pro one. Thus, we are certain that we do not incorporate errors by hand.

The practical decomposition into shapelets is done as follows. The function to be decomposed is the intensity of the object, *i.e.* the value of pixels in the postage stamp which contains the object (so, the postage stamp is an input of the routine). The routine computes the size of the postage stamp and the centroid of the object (if necessary) which will be the center of the basis functions. Then, it effectively decomposes the object into shapelets (see section 2.4.1), recomposes it (still convolved with the PSF, even though PSF deconvolution is performed when decomposing), computes the covariance matrix of the shapelets coefficients and χ^2 (the difference between the original and reconstructed image)[11], and writes the shapelets coefficients in a decomp structure, together with other useful quantities.

The user does not have to call the routine himself, because it is directly used by the shapelets_focus.pro function (which outputs the decomp structure - see below) :

decomp=shapelets_decomp(*f, beta, n_max, options*)

- *f* is the 2D function to decompose into shapelets. Practically, this is the intensity of the postage stamp containing the object to decompose, *pstamp.image* ; in the command line, *pstamp* (a postage stamp given by shapelets_pstamp) is passed as argument. The routine then extracts and decomposes only pstamp.image (*i.e.* the value of the pixels in the postage stamp).
- *beta* is the focussed scale parameter.

[11] To compute χ^2, we use the noise calculated when creating the postage stamp, by shapelets_sexcat2pstamp ; this noise is entered as an optional input in shapelets_decomp

- *n_max* is the focussed maximal order for decomposition.
- CENTER=center is the center of the basis functions. It should be the focus one, but if it is not set, the center of the postage stamp is taken to be the center of the basis functions.
- NAME=name is the name of the object. By default, the object has no name.
- PSF=psf is a decomp structure representing the PSF on the object; if set, the PSF is deconvolved from the object.
- NOISE=noise is a noise map (*i.e.* an inverse variance map of pixels). If it is not set, the noise is assumed to be constant. In practice, we use the noise estimated by shapelets_sexcat2pstamp and stored in the pstamp structure.
- RECOMP=recomp is the output recomp image (still convolved with the PSF ; to remove this, one must use the shapelets_recomp routine [12]).
- OVERSAMPLE=oversample forces an oversampling factor to evaluate the basis functions. There is no oversampling if over=1.
- /OVERLAP is to be set to fit the coefficients with Fourier-like overlap integrals instead of the least-square method.
- INTEGRATE=integrate serves, for pixelizing the shapelets model [13], to use basis functions integrated within pixels instead of pixels central value. It is set by default (to take center, set integrate=2).
- SKY=sky : to be set when subtracting the background ; to be set to :
  - 1 to fit it with a constant value around the object
  - 2 to fit it with a plane around the object
- /NON1 : if set, a_{01} and a_{10} are required to be 0.
- /POLAR : to be set to use polar shapelets ; by default, cartesian shapelets are used.
- /DIAMOND : to be set to use the diamond truncation scheme for decomposition
- /FULL_ERROR is to be set to return the full covariance matrix of the coefficients in “error” tag name (deprecated in current version).
- /SILENT is to be set so as the routine operates silently.

[12] see chapter 3, section 3.2

[13] see chapter 1, section 1.6

- /LS is obsolete ; it used to be set to fit the coefficients with the least squares method (now by default) instead of Fourier-like overlap integrals (now used when setting /OVERLAP).
- X0=x0 is obsolete ; it used to be CENTER

We describe the decomp structure hereafter.

2.4.3 The decomp structure

It gathers all useful parameters of the shapelets decomposition of an object. It notably contains :

1. beta : the scale parameter β used for decomposition.
2. n_max : the maximum order for decomposition $n_{\max}$ used for decomposition.
3. n_coeffs : the number of coefficients used in decomposition.
4. coeffs and coeffs_error : the coefficients of the decomposition and their errors.
5. n_pixels : the size of the postage stamp, expressed in pixels.
6. n1 and n2 : the one-dimensional orders of coefficients.
7. chisq : χ^2, the difference between the original and recomposed image.

So far, we have shown how to properly decompose an object into shapelets, either a star or a galaxy. But the goal of a weak lensing (or simply an astronomical) analysis of an image requires dealing with thousands of objects (*e.g.* Subaru images contain approximately 70000 utilizable objects). It would be inhuman to decompose them one by one, creating thousands of postage stamps and then calling each time shapelets_focus (let us still emphasize on the fact that the user does not have to call shapelets_decomp ; the routine is included in and automatically called by shapelets_focus.pro). One program, built upon those two former routines, deals with all objects. We present it in the next section.

2.5 Analysing all objects in an image

So as to perform a shapelets analysis of an image, the user ideally has to run only one program, which is in charge of decomposing each object detected by SExtractor into shapelets. As it is not limited in terms of number of possible decompositions (just the memory let available by idl and the upper limit of long type variable can limit it), one can easily treat a physical, current image. For example, a Subaru image covers 0.25 deg^2 and contains approximately 70000 galaxies : this is easy to treat them all.

2.5.1 The shex.pro routine

A routine called shex.pro is the orchestra leader of all the shapelets analysis (we exclude from this term the processes aimed at modeling the PSF [14], and at estimating the shear, in the case of cosmic shear measurements). It reads in the SExtractor catalogue, and extracts and decomposes each object into shapelets, one after each other. Therefore, the routine is "just" a loop on objects to decompose. Each loop is aimed at decomposing one object, by creating a postage stamp (shapelets_sexcat2pstamp), focussing the needed parameters and decomposing the object into shapelets (shapelets_focus). The routine can decompose objects either with or without PSF correction (using deconvolution [15]).

Each object is analysed, and all those that have a focus flag greater than 8 and a postage stamp flag greater than 3 are skipped, but still written in the output catalogue (except if explicitly requested that they are not recorded). Before describing the routine's options, two features must be emphasized. First, weak lensing measurements usually deal with images with thousands of galaxies, and shex can run for hours, even for days, until each galaxy is decomposed ; since it would be frustrating that an object makes shex crash just before the end (say, the 65000th object over 67000 in an image...), and a great waste of time, shex is enabled to periodically record temporary output catalogues, and to recover them in case it crashes. Second, defining the size of postage stamps using SExtractor's parameter can prove wrong, and give too small a postage stamp ; as a consequence, shex can be enabled to automatically increase the size of postage stamps.

shex is simply called as :

```
shex,file_name, options
```

using the arguments and options :

- FILENAME_OUTPUT=filename_output : is the name of the output file containing the shape catalogue ; by default, it is *file_name*.
- FULL_PATH=full_path : (optional) gives the full path of data files. If it is not set, the path is assumed to be given by shapelets_paths.
- INDEX=index : allows to decompose only the objects given in index (must be a scalar or an array).
- /RESTART : shape catalogues are periodically written to disk, so as to avoid re-decomposing thousands of objects in case shex crashed late in its run ; in that case, setting /RESTART allows shex to start from the last written to disk decomposition.

[14] see Volume II
[15] see chapter 3

- SAVE_EVERY=save_every defines the frequency at which decomposition are periodically saved to disk ; default is every 2000 objects.
- IMAGE=image : is an image structure of *file_name*, given by shapelets_read_image ; if it is not set, it is automatically loaded by shex, and it can be optionally output if none is entered as input.
- SEXCAT=sexcat : is a SExtractor catalogue of *file_name*, given by shapelets_read_sexcat ; if it is not set, it is automatically loaded by shex, and it can be optionally output if none is entered as input.
- SHAPECAT=shapecat : if it is set, the shape catalogue is output under this name.
- BETA_TOLERANCE=beta_tolerance : is the accuracy with which β is obtained. The default value is 10^{-3}.
- FIXED_BETA=fixed_beta : if it is set, all objects will be decomposed using this scale ; in that case, there is no focus on β.
- CHISQ_TARGET=chisq_target : is the ideal value of χ^2 for the residual image. The default value is 1.0
- CHISQ_TOLERANCE=chisq_tolerance : is the acceptable accuracy for χ^2 ; default value is 1.0
- CHISQ_FLATNESS=chisq_flatness : is the minimum difference in χ^2 between two decompositions with $n_{\max}$ differing by two to trigger the flatness constraint in shapelets_focus_nmax.
- THETA_MIN_GEOM=theta_min_geom : is the minimum scale on which it is possible for the image to contain data ; default value is 0.2 pixels.
- /FULL_FOCUS : if it set, all attempts made at decomposition during iteration are recorded.
- /GAUSSIAN_RECENTRING makes the basis functions recentered on the previous decomposition's centre of light.
- N_MIN=n_min : is the minimum value allowed for $n_{\max}$ for decomposition ; default value is 2.
- N_MAX=n_max : is the maximum value allowed for $n_{\max}$ for decomposition ; default value is 20.
- /NOISE_MAP : if it is set, the noise map supplied in the postage stamp is used ; otherwise, the noise is determined locally.
- /SEG_MAP : if it is set, the segmentation map is determined locally; otherwise, the segmentation map supplied in the image structure is used.

- TOO_BIG=too_big : is the maximum radius of postage stamp ; default value is 100 pixels. Objects that are bigger than this radius are skipped.
- SG_CUT=sg_cut : is the threshold for SExtractor star/galaxy classification. If this is positive, then only galaxies with lower class are decomposed ; if this is negative, only stars with class greater than -sg_cut are decomposed ; if it is zero or not set, everything is decomposed.
- /LAZY_PSTAMP allows not to decompose objects for which defects have been detected in their postage stamp.
- /SQUARE : if it set, squared postage stamps are used ; otherwise, circular postage stamps are used.
- /NEIGHBOUR : this keyword codes the treatment of neighbors. The default behaviour is to do an unconstraint fit ; if the keyword is set, neighbors pixels are set to the background level and associated errors.
- NFWHM=nfwhm : is the size of the postage stamp in units of SExtractor semi-major axis a.
- /TRIM_FAILURES : if it is set, objects with bad postage stamp or bad focus iterations are discarded and not kept in memory.
- N_OBJECTS=n_objects is the (output) number of decomposed objects during the run of shex, and written in the shape catalogue.
- PSF=psf : is the PSF structure to deconvolve from the object ; if it is not set, the object is decomposed without doing any deconvolution.
- SKY=sky : allows one to subtract sky level. It can be set to :
  - 0 (default) not to subtract the sky.
  - 1 to fit it with a constant value around the object
  - 2 to fit it with a plane around the object
- /NON1 : if set, shapelets coefficients with $n = 1$ are forced to be zero.
- TRUE_IMAGE=true_image : is an additional input image, for plotting purpose only (*e.g.* an image in other color).
- SCAT_IN=scat_in : is a SExtractor catalogue, for plotting purpose only.
- PIXEL_SCALE=pixel_scale is the scale of pixel, in arcsec ; the default one is given in shapelets_read_image.pro

- UNITS=units is the units of the image ; the default one is given in shapelets_read_image.pro
- PHOTO_ZP=photo_zp is the photometric zero-point for the image; the default one is given in shapelets_read_image.pro
- EXPOSURE_TIME=exposure_time is the exposure time ; the default one is given in shapelets_read_image.pro
- /REDRAW_ADAPTIVE allows shex to increase the size of the postage stamp if the decomposition crashes because the original postage stamp is too small ; N_REDRAWS and DELTA_REDRAW parametrize this option.
- N_REDRAWS=n_redraws is the number of times shex is authorized to increase the postage stamp, each time by a fractional factor DELTA_REDRAW ; the default value is 5.
- DELTA_REDRAW=delta_redraw is the fractional increase of the postage stamp if an increase is necessary ; the default value is 0.05.
- /PLOTIT : allows to draw some plots on the screen.
- /SILENT : is to be set to operate silently.
- /VERBOSE : is to be set to operate noisily.

shex outputs a catalogue containing all useful quantities about each decomposed object : shapelets parameters, astrometry, photometry... The name of this structure is of the form *file_name.shapecat*, .shapecat being a reference to "shapelets". The next section presents it.

2.5.2 The shapecat catalogue

It gathers lots of parameters, all of them can be useful later. Among them, one can notably find :

1. the number of objects shex tried to decompose n
2. β and $n_{\max}$ for each object.
3. the maximum $n_{\max}$, called maxn_max.
4. the position of each object, x
5. the number of coefficients, n_coeffs
6. the shapelets coefficients and their errors for each decomposed object
7. the postage stamp and focus flags for each object

8. the reduced χ^2 for each object
9. the estimated background level for each object
10. the used CHISQ_TARGET
11. the rms of χ^2
12. the estimated seeing
13. SExtractor parameters, the name of which begins by sex, for each object
14. the edges of postage stamps
15. if requested, the moments of each objects : flux, centroid, size, quadrupole, ellipticity, and their errors

This catalogue, stored in *file_name.shapecat*, can be later read using the shapelets_-read_shapecat.pro routine, and will be used for all shape and cosmic shear analysis.

2.5.3 Keep only good objects

If shex.pro has been used without the /trim_failures options, then the shapecat catalogue contains all detected objects on the image. That is, it contains objects for which decomposition into shapelets has not been achieved for some reason. These objects are thus not reliable for futher analysis, and must be rejected. The easiest way to do that is to use the /trim_failures option in shex ; otherwise, one can still eliminate them after the shapecat catalogue has been created, using the shapelets_split.pro routine.

This routine splits any kind of catalogue so as to keep only a selection of objects, this selection being entered as an argument of the routine.

Chapter 3

PSF correction

One of the predominant systematics in weak lensing measurements is brought by the Point Spread Function (PSF), due typically to the atmosphere and to the response of the telescope. Keeping in mind that cosmic shear effects are of order 1-2 %, and that the PSF can be of order 6-7 %, it becomes obvious that correcting for it is crucial. A prior to the correction is to model the variations of the PSF across the image (*i.e.* on each galaxy), as shown in Volume II. It then becomes possible to correct for the PSF.

3.1 How to correct for the PSF ?

Since the PSF is just a convolved response to the image, a simple deconvolution suffices to get rid of its effects. Theoritically, the problem is very easy, but it is less evident from a numerical point of view. Indeed, as seen earlier (chapter 1 section 1.5), one has to invert a matrix to compute a deconvolution (this is real not only for the shapelets case, but also for every method implying a deconvolution), and the problem becomes numerically ill as soon as the matrix is degenerate or not inversible.

So as to avoid this problem, some methods have been developped to correct the PSF without a direct deconvolution. See for example the RRG (Rhodes, Refregier & Groth, [28]) method. Nevertheless, it must confessed that a method using a direct deconvolution is more elegant and accurate (provided that it can deal with numerical problems).

Two kinds of method can be thought of to deconvolve the PSF. Both rely on reproducing data (*i.e.* observed objects), by modeling either (1) the smeared object (*i.e.* the object one sees through the telescope) or (2) the unsmeared object (*i.e.* the object as it would be seen without a PSF). In the case (1), the PSF is deconvolved afterwards ; in case (2), the PSF is convolved to the model, and the model tuned so as the convolved model reproduces data. Case (2) has the advantage that it does not compute a deconvolution, but a convolution, and hence, does not need to invert a matrix : as already mentionned, shapelets use this method to correct for the PSF.

3.2 Correcting for the PSF with shapelets

3.2.1 Convolution formalism

The convolution formalism for shapelets has already been briefly introduced in chapter 1, section 1.5. We develop it here, and give some useful formulae needed to understand the shapelets deconvolution scheme (section 3.2.3). They can all been found in [27], together with their justification.

Let f and g be two 2-dimensional functions, and h their convolution product. Their shapelets coefficients (respectively $f_{\mathbf{n}}$,$g_{\mathbf{n}}$ and $h_{\mathbf{n}}$) are linked by the equation :

$$h_{\mathbf{n}} = \sum_{\mathrm{m,l}} C_{\mathbf{nml}} f_{\mathbf{m}} g_{\mathbf{l}} \tag{3.1}$$

where the 2-dimensional convolution tensor $C_{\mathbf{nml}}(\gamma,\alpha,\beta)$ can be written in terms of the one-dimensional one $C_{\mathrm{nml}}(\gamma,\alpha,\beta)$:

$$C_{\mathbf{nml}}(\gamma,\alpha,\beta) = C_{\mathrm{n_1m_1l_1}}(\gamma,\alpha,\beta) C_{\mathrm{n_2m_2l_2}}(\gamma,\alpha,\beta) \tag{3.2}$$

The convolution tensor C_{nml} depends on the scales of f, g and h, respectively named α, β and γ, through the relation :

$$C_{\mathrm{nml}}(\gamma,\alpha,\beta) = \sqrt{(2\pi)}(-1)^n i^{n+m+l} B^{(3)}_{\mathrm{nml}}(\gamma^{-1},\alpha^{-1},\beta^{-1}) \tag{3.3}$$

where $B^{(3)}_{\mathrm{nml}}$ is defined as :

$$B^{(3)}_{\mathrm{lmn}}(a_1,a_2,a_3) \equiv \int_{-\infty}^{\infty} \mathrm{d}x B_{\mathrm{l}}(x,a_1) B_{\mathrm{m}}(x,a_2) B_{\mathrm{n}}(x,a_3) \tag{3.4}$$

and $B_{\mathrm{n}}(x,a)$ is the nth order one-dimensional shapelets basis function and is given by equation (1.2).

One can evaluate the integral expression (3.4) giving $B^{(3)}_{\mathrm{lmn}}$:

$$B^{(3)}_{\mathrm{lmn}}(a_1,a_2,a_3) = \nu \left[2^{l+m+n-1}\sqrt{\pi} m! n! l! a_1 a_2 a_3\right]^{-1/2} \times L\left(\sqrt{2}\frac{\nu}{a_1}, \sqrt{2}\frac{\nu}{a_2}, \sqrt{2}\frac{\nu}{a_3}\right) \tag{3.5}$$

where we define $\nu^{-2} \equiv a_1^{-2} + a_2^{-2} + a_3^{-2}$ and :

$$L_{\mathrm{lmn}} \equiv \frac{1}{\sqrt{\pi}} \int_{-\infty}^{\infty} \mathrm{d}x \ \mathrm{e}^{-x^2} H_{\mathrm{l}}(ax) H_{\mathrm{m}}(bx) H_{\mathrm{n}}(cx) \tag{3.6}$$

H_{n} being an nth order Hermite polynomial.

By parity, $L_{\mathrm{lmn}} = 0$ if $\mathrm{m+n+l}$ is odd. Table (3.1) gives the first few non-zero components of this function.

$L_{000} = 1$
$L_{002} = -2 + 2c^2$
$L_{011} = 2cb$
$L_{022} = 4 - 4b^2 - 4c^2 + 12b^2c^2$
$L_{112} = -4ab + 12abc^2$
$L_{013} = -12bc + 12bc^3$
$L_{004} = 12 - 24c^2 + 12c^4$
$L_{006} = -120 + 360c^2 - 360c^4 + 120c^6$

Table 3.1: First few components of the normalized three-product integral $L_{\rm lmn}(a, b, c)$; other components can be obtained by symmetry (*e.g.* $L_{020} = -2 + 2b^2$); courtesy from [27].

Shapelets thus allow to analytically compute convolution products. To calculate the convolution product of two functions f and g, one just has to decompose them into shapelets, compute their convolution tensor $C_{\mathbf{lmn}}$ and apply equation (3.1).

The main difficulty in this calcul is to evaluate shapelets scales α, β and γ and maximum order of decomposition $n_{\rm max}$, $m_{\rm max}$ and $l_{\rm max}$ of the two convolved functions f, g, and of their product of convolution h. Contrary to $(\alpha, n_{\rm max})$ and $(\beta, m_{\rm max})$ which are directly known when decomposing f and g into shapelets, γ and $l_{\rm max}$ are unknown. But as they are needed to compute h, one has to be able to evaluate them.

Convolution algebra allows one to justify that

$$\left\{ \begin{array}{rcl} \theta_{\gamma,\rm min} & = & \theta_{\alpha,\rm min} + \theta_{\beta,\rm min} \\ \theta_{\gamma,\rm max} & = & \theta_{\alpha,\rm max} + \theta_{\beta,\rm max} \end{array} \right. \tag{3.7}$$

where $\theta_{\alpha,\rm min}$ (resp. $\theta_{\beta,\rm max}$) is the minimum (resp. maximum) size described by shapelets decomposition of function f (resp. g), as defined in chapter 1, section (1.2.2).

Thus, using equations (1.28), we can conclude that γ and $l_{\rm max}$ should be given by :

$$\gamma = \left(\frac{[\alpha^2(n_{\rm max}+1) + \beta^2(m_{\rm max}+1)]\ [\alpha^2(m_{\rm max}+1) + \beta^2(n_{\rm max}+1)]}{(n_{\rm max}+1)(m_{\rm max}+1)} \right)^{\frac{1}{4}} \tag{3.8}$$

and

$$l_{\rm max} = \left(\frac{\alpha^2(n_{\rm max}+1) + \beta^2(m_{\rm max}+1)}{\alpha^2(m_{\rm max}+1) + \beta^2(n_{\rm max}+1)} (n_{\rm max}+1)(m_{\rm max}+1) \right)^{\frac{1}{2}} - 1 \tag{3.9}$$

In practice, instead of those formulae, we use empirical recipes which give better results. We evaluate γ through the equation :

$$\gamma^2 = \alpha^2 + \beta^2 \tag{3.10}$$

and we set $l_{\max}$ as being the maximum of $(n_{\max}, m_{\max})$:

$$l_{\max} = \max(n_{\max}, m_{\max}) \tag{3.11}$$

Using these rules, we can either convolve or deconvolve every function in an analytic way. In astronomy, we are interested in deconvolving the PSF from the image ; thus the convolution formalism will be mostly used for deconvolution.

3.2.2 Deconvolution formalism and method

As evoked above, deconvolution is needed to clear smeared galaxies (described by the function h in the above notations) from the PSF (described by the function g). In other words, it is aimed at recovering the coefficients $f_{\mathbf{m}}$ from the measured coefficients $h_{\mathbf{n}}$ of galaxies and the interpolated coefficients $g_{\mathbf{l}}$ of the PSF at the position of each galaxy.

For this purpose, it is convenient to rewrite equation (3.1) as :

$$h_{\mathbf{n}} = \sum_{\mathbf{m}} P_{\mathbf{nm}} f_{\mathbf{m}} \tag{3.12}$$

where $P_{\mathbf{nm}} = \sum_{\mathbf{l}} C_{\mathbf{nml}} g_{\mathbf{l}}$ is the 'PSF matrix'. As briefly noted in chapter 1, section 1.5 and well explained in [27], this matrix is invertible under some hypothesis. Therefore, equation (3.12) can be directly inverted to obtain the coefficients of unsmeared galaxies :

$$f_{\mathbf{m}} = P^{-1}_{\mathbf{nm}} h_{\mathbf{n}} \tag{3.13}$$

This method is intuitive and gives an estimate of the low-order coefficients of unconvolved galaxy coefficients. But it is not satisfactory, because it restricts the deconvolution to low-order coefficients, whereas we need to know the biggest amount information available to eventually analyze unsmeared coefficients, and because the PSF matrix may be not inversible.

Consequently, as already said in section 3.1, we prefer to convolve the PSF to an "unconvolved" model, than deconvolving it to a "convolved model".

That is, while decomposing a smeared galaxy into shapelets, assuming that the PSF coefficients on this galaxy are known, we directly search a model of the unsmeared galaxy. We convolve this model to the PSF, and tune it (by focusing its scale and its maximum order of decomposition) until the convolution product (model $*$ PSF) reproduces the observed smeared galaxy.

Thus, we do not proceed to a direct deconvolution, and never invert a matrix: all we do is a convolution. The next section describes how the shapelets package deals with this process.

3.2.3 Deconvolution in practice with shapelets

The PSF deconvolution occurs in the whole shapelets decomposition process. Thus, it is not a well defined routine which achieves it, but it is a sparse process, in which each routine implicated in the decomposition takes part. We do not describe this process again, and refer the reader to chapter 2 for a precise view of it. Here, we just focus on how the PSF deconvolution works through it.

It begins at the very begining of the shapelets decomposition, when we try to guess starting scale β and maximum order $n_{\max}$. We search them using the usual geometric constraints, and transform β (which, as this stage, is the scale of the smeared galaxy - convolved with the PSF) inverting the above empirical rule for convolution (equation (3.10)) ; thus, the β we will use in further steps is the scale of the unsmeared (deconvolved from the PSF) galaxy.

We then pass this scale β to further routines : shapelets_focus.pro, shapelets_-focus_beta.pro, shapelets_focus_nmax.pro, all relying on shapelets_decomp.pro. The goal of these routines is to search for an accurate model of the unsmeared galaxy, that is, of the unconvolved galaxy. The deconvolution heart lies in shapelets_decomp.pro, where the model of the unsmeared galaxy is convolved to the PSF and compared to the observed (smeared) galaxy (while focus is done so as we have a model as accurate as possible).

To achieve that, one routine is essential, which we describe here : shapelets_-convolution_matrix.pro As its name suggests, it is aimed at computing the convolution matrix $P_{\mathbf{nm}}$ which links the smeared galaxy, the unsmeared one, and the PSF.

It is called as :

$P_{\mathbf{nm}}$ = shapelets_convolution_matrix(*psf,alpha,n_max_alpha,gamma,n_max_gamma*)

where :

- *psf* is the PSF structure to deconvolve, *e.g.* given by psf_interpolation.
- *alpha* is the scale of the unconvolved object - that is, of the model we want to create ; basically, it is the β given by shapelets_guess_-nmaxbeta
- *n_max_alpha* is the maximum order of the unconvolved object
- *gamma* is the scale of the convolved (observed) object - if it is not set, it is computed using the above rules for deconvolution (equation (3.10)).
- *n_max_gamma* is the maximum order of convolved object ; if it is not set, it is computed using the above empirical rule for deconvolution (equation (3.11)).

In practice, we could measure the scale and maximum order of the observed (convolved) galaxy, and set them as arguments of shapelets_convolution_matrix, together with the guessed one for the model of the unsmeared galaxy. This would probably be the most efficient way to obtain an accurate model, but it would need a first iteration of shex on the image, without deconvolution, and then, a second one with deconvolution. Since computer time would become prohibitive, we prefer to use the default *gamma* and *n_max_gamma* given by the above rules for deconvolution. As long as simulations give good results with this method, we trust and use it.

NB : *Note that doing so, we never measure the shapelets scale and maximum order of decomposition, and we never compute shapelets coefficients of the smeared galaxy.*

Once we are in possession of the convolution matrix, we can compute the convolved basis functions, using a least squares fitting, then the unconvolved ones, and finally, we can compute the unconvolved shapelets coefficients.

To conclude, at the end of these operations, we have computed a model of the unsmeared galaxy. In other words, we have deconvolved the PSF from the observed image. We will eventually use this shape information to measure cosmic shear. Nevertheless, we must be sure that the deconvolution process went good : we use some diagnosis that we present in the next section.

3.2.4 Recompose an object deconvolved from the PSF

We just saw how to deconvolve the PSF from a given object, but we did not evoke how to look at this corrected object. This can be done using the shapelets_recomp.pro routine, which recomposes an object after it has been decomposed into shapelets. Particularly, if one introduces the PSF in this routine, it recomposes the object (re)convolved from the PSF ; otherwise, the unconvolved object is recomposed.

3.3 Is the PSF correction reliable ?

3.3.1 The two-point correlation function of the PSF

The two-point correlation function of the PSF ellipticity is defined as the average over all couples of stars, of the product of the ellipticity of both members of each couple. It depends on the separation r of the stars.

$$C_0 = < \varepsilon_1(\theta)\varepsilon_1(\theta + r) > + < \varepsilon_2(\theta)\varepsilon_2(\theta + r) > \quad (3.14)$$

If the ellipticity of two stars is positively (negatively) correlated, then their correlation function will be positive (negative). If no correlation exists, then the correlation function should be zero. A two-point correlation function can also be measured for the size of the PSF.

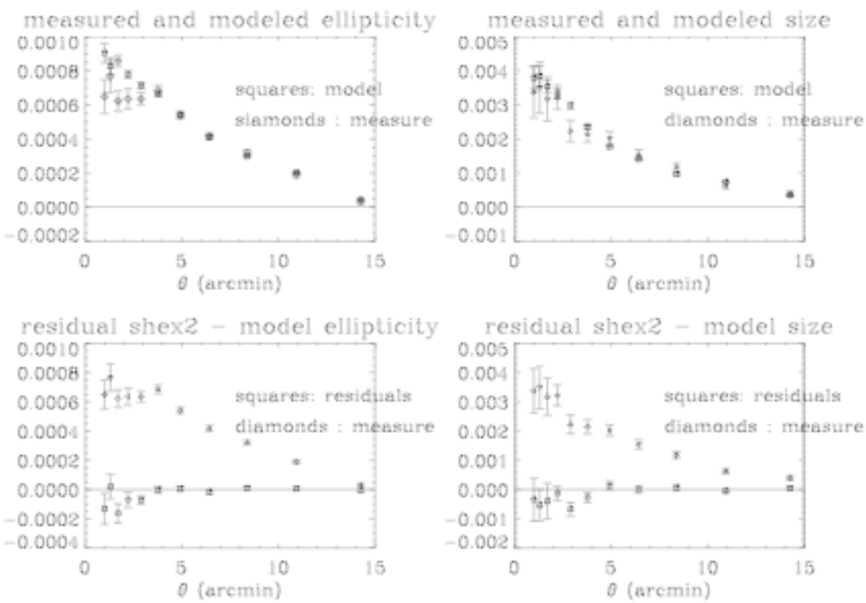

Figure 3.1: Two-point correlation functions for the PSF ellipticity (left panels) and PSF size (right panels). In both cases, top panels compare the raw correlation function (diamonds) and the correlation function of the modeled PSF (squares) ; bottom panels compare the raw correlation function (diamonds) and the correlation function of the residuals between the raw and the modeled PSF (squares).

As the PSF consists in a pattern, stars within a same neighbourhood should be correlated, whereas distant stars should be uncorrelated. As a result, the two-point correlation of the PSF (*i.e.* of the stars before correction for the PSF) is expected to be significant for small scales, and negligible for large ones.

On the contrary, after correction for the PSF, the pattern of the PSF should have been smoothed, almost erased. Thus, the two-point correlation function should be negligible at all scales.

Figure 3.1 shows the behavior of the two-point correlation function of a PSF ellipticity (left panels) and of the same PSF size (right panels). Left-top panel shows the correlation of the raw (measured) PSF ellipticity (diamond symbols), together with the correlation of the modeled PSF ellipticity (square symbols). Left bottom panel shows the correlation of the raw PSF ellipticity (diamond symbols), together with the correlation of the residuals (model - raw) ellipticity PSF (square symbols). Right panels show the same quantities, but for the PSF size. The model of the PSF here was done using a polynomial interpolation of PSF shapelets coefficients. Note the slight inaccuracy at scales less than 3 arcmin, which is the smoothing scale used for the interpolation.

If the two-point correlation of the stars after correction for the PSF is negligible, then one can be really confident on one's model of the PSF and its correction.

Nevertheless, something wrong could have occurred in the deconvolution process. The next section describes how comparing some objects at different steps of their treatment can help decide if the deconvolution is reliable.

3.3.2 Looking at objects

After deconvolution, stars should be Dirac function. Plotting their unconvolved model can allow one to decide instantly whether the deconvolution seems OK or not: an unconvolved stars well peaked on a pixel is the sign that the deconvolution has worked ; on the opposite, an unconvolved star which spreads out on lots of pixels (thus, not a Dirac function), reveals the fact that the deconvolution is likely to have gone wrong. One can also reconvolve unconvolved stars with the PSF, and compare the results to the original star, as seen before deconvolution : they should be identical. If this is not the case, that means that a problem occured during deconvolution. The same work can be done on galaxies, but seems to be less efficient to check problems.

3.3.3 Limits of deconvolution diagnosis

Using the two diagnosis described above, one can check whether the correction of the PSF has gone well or wrong. Nevertheless, it must be emphasized that computing the two point correlation function of the PSF and of its model is not a real test on deconvolution. It just allows to diagnose the starting point of the deconvolution, that is, the goodness of the model of the PSF.

Furthermore, plotting objects after deconvolution is not a really accurate test, as we do not know their shape without PSF (that is precisely what we aim at obtaining *in fine*) ; here, plotting unconvolved stars seems the best way to proceed: they must be Dirac functions. Reconvolving objects already deconvolved from the PSF, and comparing them to observed objects, may be a very good way to look at the precision of the deconvolution ; unfortunately, a same problem could occur both in deconvolution and in reconvolution, and vanish when reconvolving a previously deconvolved object. As a result, even this way to proceed does not seem perfect.

Knowing the difficulty of these steps, deconvolving and diagnosing deconvolution, we use some tests which are not as accurate as we would wish, but which allow to discard very bad deconvolution. However, so far, they have proved to be precise enough for our purposes.

Chapter 4

Interfacing shapelets

4.1 The input/output programs : how to read and write data ?

Most shapelets routines need to read information stored in some catalogues or structures, and to write results to the disk. We have already met such input/output programs (*e.g.* shapelets_read_image) without explaining what they actually do, and how they do it. This is the goal of this section.

4.1.1 Read in a fits image

The first step for a shapelets analysis is to read the fits file containing the image data, and create an image structure with which shapelets can deal. This work is done by shapelets_read_image, which also reads in the mask image (composed of 0 and 1 whether or not a mask exists or not). It can also read in the segmentation map created by SExtractor (if requested), and read in the SExtractor's noise map or estimate the noise by itself.

shapelets_read_image,*image, file_name, options*

It uses the arguments and options :

- *image* is the output image structure.
- FULL_PATH=full_path is the full path of *file_name*.
- /FITS_READ is to be set to use the fits_read.pro procedure instead of the readfits.pro one for reading the fits file.
- PIXEL_SCALE=pixel_scale is the scale of pixel, in arcsec ; the default value is the one stored in the header of the fits file, if available.
- UNITS=units is the units of the image ; the default value is the one stored in the header of the fits file, if available.

- PHOTO_ZP=photo_zp is the photometric zero-point for the image; the default value is the one stored in the header of the fits file, if available.
- EXPOSURE_TIME=exposure_time is the exposure time ; the default value is the one stored in the header of the fits file, if available.
- /NO_MASK allows not to read in the mask image (to save memory and/or CPU).
- /NO_SEGMENTATION allows not to read in the SExtractor's segmentation map.
- /NO_NOISE allows not to read in the SExtractot's noise map.
- /ESTIMATE_NOISE allows to estimate the noise in the image.
- NOISE_LEVEL=noise_level is an (optional) output containing an estimate of the rms of the background noise.
- N_GROW=n_grow is the number of times SExtractor image is grown to mask objects during noise estimation.
- /SKY_SUBTRACT allows to subtract the sky level.
- /SILENT makes the routine operate silently.

4.1.2 Read in a SExtractor catalogue

SExtractor outputs a catalogue containing astrometric and photometric parameters of detected stars and galaxies. It is described in Volume II. The eventual shapelets analysis relies on this catalogue. The interface with it is done by shapelets_read_sexcat.

shapelets_read_sexcat,*sexcat*,*file_name*, *options*

The routine outputs *sexcat*, a SExtractor catalogue structure. The options are :

- /FULLPATH : if set, the routine accepts the *file_name* as is (can be absolute or relative). The default behaviour is to prepend a path name from shapelets_paths.pro.
- /ASCII : if set, the routine assumes the catalogue is in ASCII format.
- COMMENT=comment is an optionally output comment on each field ; it gives the significance of each returned SExtractor's parameter
- UNIT=unit is the unit of each returned SExtractor's parameter.

- /XBUGFIXED : seemingly, a bug often occurs in SExtractor, the x-coordinate are increased by 1. The default behaviour of the routine is to take this bug into account, y subtracting 1 to SExtractor's x-coordinates. XBUGFIXED must be set if the bug does not occur.
- /SILENT : if set, the routine operates silently.

4.1.3 Write a decomp or a shapecat catalogue

We saw that shapelets_decomp and shex output shapelets structures. To achieve this, they use the shapelets_write routine, aimed at writing shapelet coefficients catalogues to disk. As it is just a blind routine, it is pointless to describe it in details here.

4.1.4 Read in a shapecat catalogue

Once a shapelets analysis has been made on an image, and a shapecat catalogue written to disk, containing shapelets parameters of each object in the image, one must be able to read it so as to use it. The shapelets_read_shapecat is aimed at doing so. Futhermore, it computes the moments of each object if requested.

The user can call it as :

shapelets_read_shapecat,*shapecat, file_name, options*

with the following arguments and options :

- *shapecat* is the (output) shapecat catalogue.
- /CARTESIAN converts all objects to cartesian coefficients.
- /POLAR converts all objects to polar coefficients.
- N_MAX=n_max : if it set, all objects are truncated at this $n_{\max}$ (*i.e.* only coefficients corresponding to lower $n_{\max}$ are considered).
- /MOMENTS allows to compute objects moments : unweighted quadrupoles, size and flux.
- FULL_PATH=full_path is the full path of *file_name*.
- /PARITY is not available yet.
- N_MOMENTS=n_moments is the $n_{\max}$ till which moments are computed.
- /SILENT is to be set to operate silently.
- /VERBOSE is to be set to operate noisily.
- DESCRIPTION=description is an optional output which contains information about the catalogue.

4.1.5 Obtain a decomp structure from a shapecat catalogue

After all objects have been decomposed into shapelets by shex, one could want to look at a given object. As the shapecat catalogue contains information for all objects, and a decomp catalogue contains information for one given object, one must be able to extract a decomp structure from a shapecat catalogue. This is done using the shapelets_shapecat2decomp routine.

Being able to deal with input/output procedures, the user must now be able to draw some plots, for example to look for possible problems during the decomposition process. Plotting procedures are the topic of the next section.

4.2 Plotting routines

When processing an image, one needs one to be able to visualize it, and to draw several plots linked to the image analysis. That is why some important plotting routines are included in the shapelets package, which are often called automatically by numerous analysis routines if requested (by the /PLOTIT keyword when it exists).

4.2.1 Draw an image : shapelets_plot_image

The first compulsory task an image processer must be able to achieve is to plot the image he/she wants to analyze. This can be done through the shapelets_plot_image routine : it is aimed at drawing whatever 2-dimensional pixelated array, including astronomical images.

It is simply called as :

shapelets_plot_image,*image, options*

using the arguments and options :

- *image* is an image structure (as given by shapelets_read_image) or a 2D array.
- FRAME=frame is the frame to give to the image ; note that it can be independent of the image ; if it is set as /FRAME, it is the real frame of the image.
- /COLBAR or /CBAR allow to draw a color bar above the image.
- CRANGE=crange is the range for color ; default value is [min(*image*),max(*image*)].
- /CLOG allows to draw a logarithmic color scale.
- CSIZE=csize is the size of the color bar.
- TITLE=title is the title of the plot.

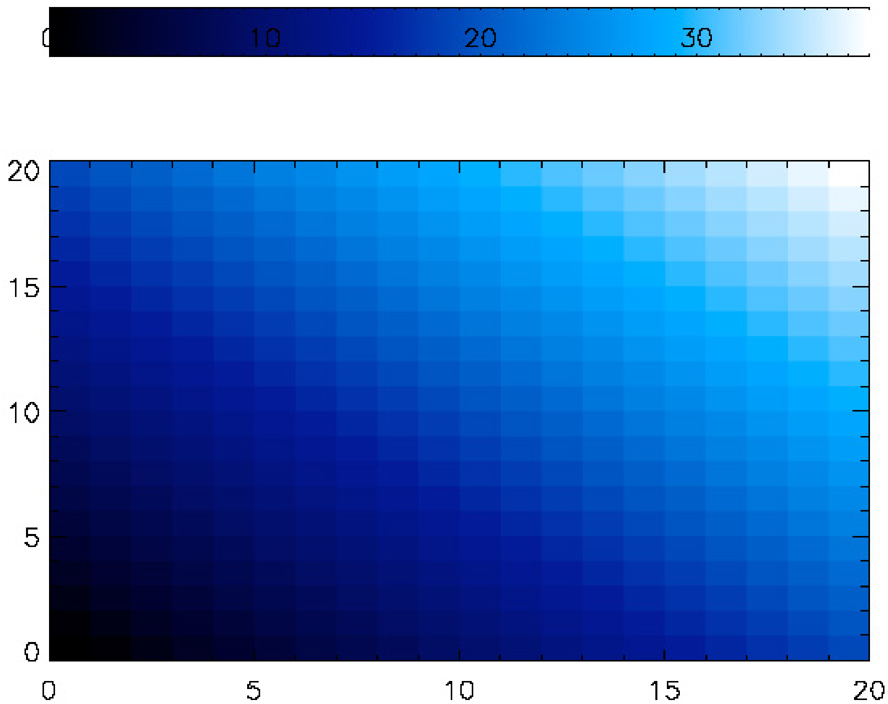

Figure 4.1: A very simple array, as plotted by shapelets_plot_image.

- XTITLE=xtitle (YTITLE=ytitle) is the title of the x (y) coordinates of the image.
- CTITLE=ctitle is the title of the color bar.
- /INVERSE allows to inverse the color coding.
- /SCALABLE allows to use scalable pixels.
- /NOERASE allows to keep everything was previously drawn in the plotting window ; otherwise, everything is erased before plotting the image.
- POS=pos is identical to an IDL !p.region variable, and allows to specify the position of the plot on a multiplot image.
- /ISOTROPIC allows to force scales on the x and y axis to be identical (note that it is not stable yet).

Figure 4.1 shows a simple array as plotted by shapelets_plot_image. The same routine is involved when plotting postage stamps like figure 2.1.

4.2.2 Statistics of an image : shapelets_plot_image_statitics

Once one possesses an image, after drawing and visualizing it, one can be interested in its statistics : distribution of the intensity in pixels, mean and rms of the image, statistics of the background...

This can be done using :

```
shapelets_plot_image_statitics,image
```

where *image* is an image structure.

The two routines presented above are very general and could be adapted to all kind of astronomical image processing, even not using the shapelets formalism. But some other routines are typical of the shapelets processing ; they are presented below.

4.2.3 Draw a postage stamp : shapelets_plot_pstamp

When isolating an object in a postage stamp (see section 2.2), one must be able to plot it so as to see what it looks like, and to check whether or not its decomposition must be problematic. In particular, some bad objects can make the whole process crash ; in such a case, being able to draw the incriminated object is of primordial importance. This task is done by plotting the object's postage stamp, using :

```
shapelets_plot_pstamp,pstamp, options
```

with the arguments :

- *pstamp* is the postage stamp to be plotted.
- /OFRAME allows to plot *pstamp* labeled with the object frame coordinates ; otherwise, it is labeled with the *pstamp* frame coordinates.
- /MASK allows to plot contours of masked out regions.
- /NOISE allows to plot the noise.
- TITLE=title is the title of the plot.
- CRAN=cran is the range for color ; default value is [min(*image*),max(*image*)].
- CSIZE=csize is the size of the color bar.
- POS=pos is identical to an IDL !p.region variable, and allows to specify the position of the plot on a multiplot image.
- /INVERSE allows one to invert the color scale.

This plotting routine is automatically used when using the /PLOTIT keyword in the shapelets_sexcat2pstamp routine. Figure 2.1 shows the output of the routine.

4.2.4 Plot the focus route : shapelets_plot_focus

Looking at the route taken through possible β and $n_{\max}$ during the focus process to obtain shapelets decomposition is important to check how well the focus worked and to which extend it is reliable. The shapelets_plot_focus routine achieves this task, called as :

shapelets_plot_focus,*focus*,*pstamp*, *options*

with the following arguments :

- *focus* is the focus structure to be examined.
- *pstamp* is its corresponding postage stamp.
- DECOMP=decomp is the corresponding final decomp structure.
- RECOMP=recomp is the object recomposed after shapelets decomposition.
- PSF=psf is the PSF to be convolved to the image, if it exists.
- IMAGE_IN=image_in is a second input image to plot (*e.g.* noise free simulated image).
- SCAT_IN=scat_in is the SExtractor catalogue of the image ; it allows to plot the ellipse whoch encloses the object.
- NOISE_MAP=noise_map : if it is set, background statistics are computed from the estimation made during the postage stamp extraction ; otherwise, they are locally computed.

The focus route as plotted by shapelets_plot_focus is shown in figure 4.2. This figure is to be looked at in a similar way to figure 2.2.

The profile of the object is also plotted, as shown by figure 4.3.

4.2.5 Draw the model of the object : shapelets_plot_decomp

Last step of a shapelets decomposition, the decomposed object can be recomposed, for example to be compared to the observed object. Plotting it can be done using :

shapelets_plot_decomp,*decomp*,*recomp*, *options*

with the following arguments :

- *decomp* is the final decomp structure.
- *recomp* is the (output) object recomposed after shapelets decomposition.

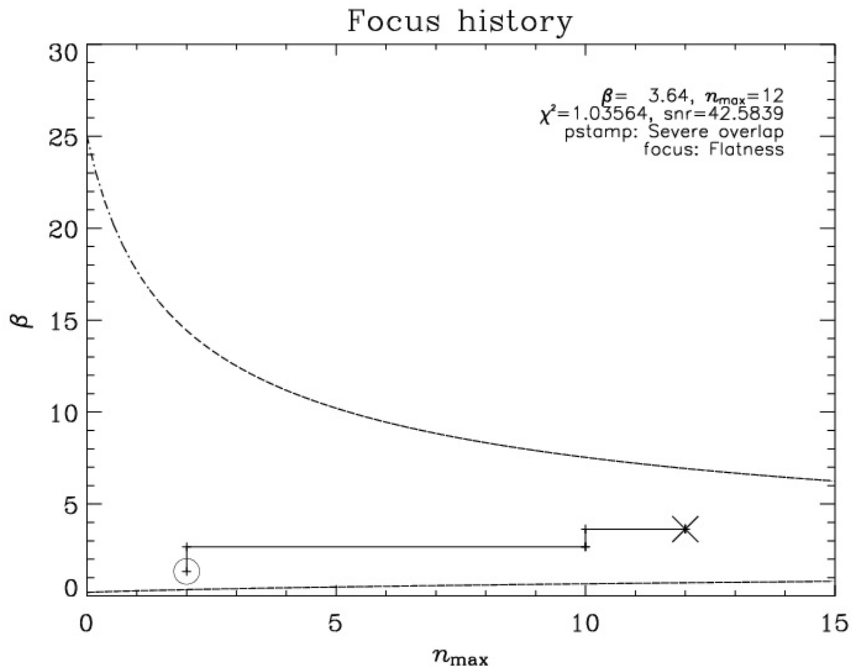

Figure 4.2: A focus route.

- /CARTESIAN or /COEFFICIENTS : if it is set, the Cartesian shapelet coefficients are plotted.
- POLAR=polar : if it set, one aspect of the polar shapelet coefficients is plotted, depending on the set options :
  1. Magnitude (default or /MODULUS option)
  2. Relative phase (/ARGUMENT or /PHASE option)
  3. Real part (/REAL option)
  4. Imaginary part (/IMAGINARY option)
  5. Composite, half and half
- ERRORS=errors : plots on noise, depending on the value this keyword is given :
  1. Plot S/N of object or coefficients
  2. Plots absolute errors
- NRAN=nran sets the n to be considered.
- TOP=top : if set, keep the top largest coefficients.
- /NOOVER allows not to oversample the basis functions.
- CRAN=cran is the range for color ; default value is [min(*image*),max(*image*)].
- /CLOG allows to draw a logarithmic color scale.

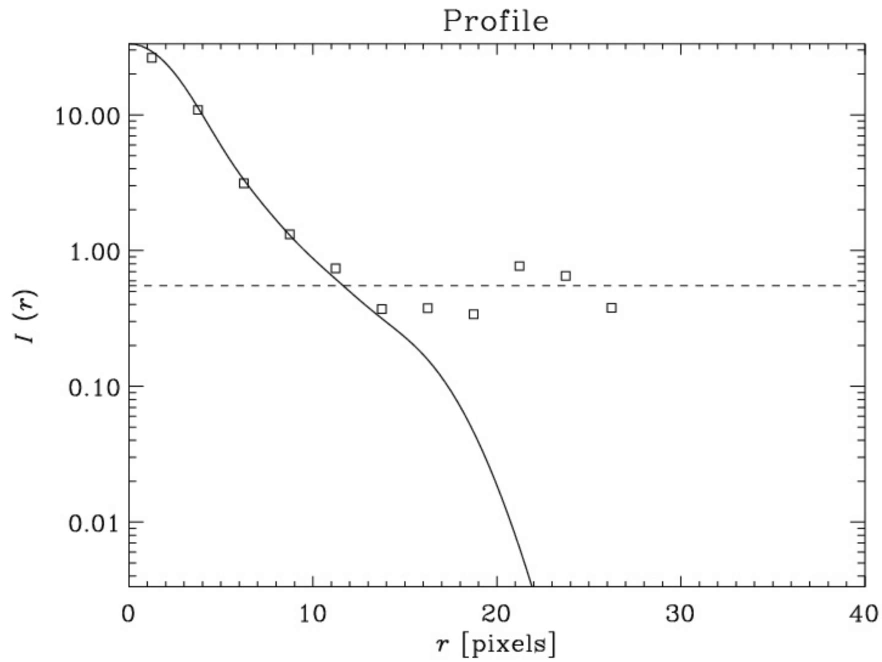

Figure 4.3: The profile of a galaxy. The dashed horizontal line represents the noise rms. Features of the profile which are in the noise (under this line) were ignored when fitting the galaxy.

- /CBAR allows to draw a color bar.
- /ISOTROPIC allows to draw a frame for which x and y scale are identical.
- /REAL : for polar shapelets, the real part of the coefficients is plotted.
- /IMAGINARY : for polar shapelets, the imaginary part of the coefficients is plotted.
- COMPOSITE_RI : not used.
- /MODULUS allows to plot the modulus of polar coefficients (this is the default option if /POLAR is set).
- /ARGUMENT or /PHASE allow to plot the phases of polar coefficients.
- /COMPOSITE_MA : not used.
- TITLE=title is the title of the plot.
- XTITLE=xtitle (YTITLE=ytitle) is the title of the x (y) coordinates of the image.
- FRAME=frame is the frame to give to the image.

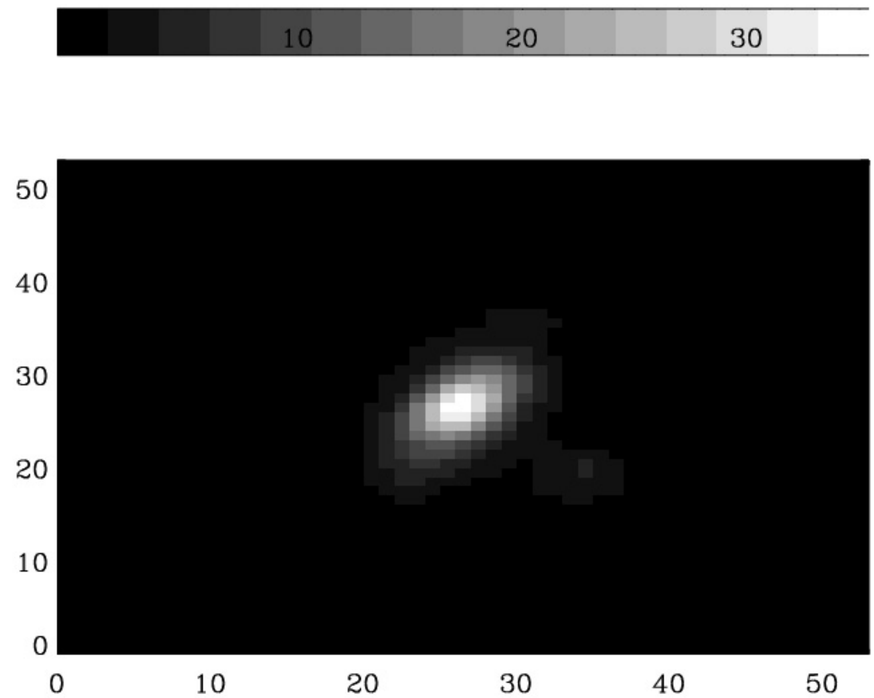

Figure 4.4: A typical output of shapelets_plot_decomp : the recomposition of the galaxy shown in Fig. 2.1, after being decomposed into shapelets.

- /CROSSHAIRS allows to overlay crosshairs on the centre of the basis functions.
- /INVERSE allows one to invert the color scale.

The default plot is the recomposed object ; options allow to plot the shapelets coefficients of its decomposition.

Note that most of the idl plotting keywords are allowed by shapelets_plot_decomp.

Figure 4.4 shows a typical output of shapelets_plot_decomp : the galaxy of Fig. 2.1 has been decomposed into shapelets, then recomposed, and its recomposition plotted by the shapelets_plot_decomp routine.

4.2.6 Gather all plots together : shapelets_plot

For convenience, a routine is able to use each one of the previous one, to plot whatever the user wants. It is called by shex when the /PLOTIT keyword is set, and allows to check all useful information on an object (profile, noise...), the focus of its shapelets parameters, and its decomposition. This routine is :

shapelets_plot,*structure, structure2, options*

and its arguments are :

- *structure* is one shapelet structure to deal with (pstamp or focus or decomp)
- *structure2* is another shapelet structure.
- /CARTESIAN_COEFFICIENTS : if it is set, the Cartesian shapelet coefficients are plotted.
- POLAR_COEFFICIENTS=polar_coefficients : if it set, one aspect of the polar shapelet coefficients is plotted :
  1. Magnitude (default)
  2. Relative phase
  3. Composite, half and half
- /STATISTICS allows to plot the image statistics.
- All optional inputs and keywords of other plotting routines are allowed.

4.3 Other routines of the shapelets package

So far, we have presented and described the more useful and visible routines of the shapelets package. But other ones exist, which are aimed at dealing with shapelets algebra, or which are just useful (and sometimes, used everywhere) programs used for some repetitive operations. It would be irrelevant and pointless to describe them all here. Appendix A lists all the routines of the shapelets package, together with their purpose.

Conclusion

Shapelets are a new generation shear measurement technique, that is now reliable for precision cosmology. We gave a short overview of their formalism, and described the pipeline we developed for dealing with shapelets algebra and decomposition. The analytical deconvolution of the PSF allows to get rid of the PSF in an efficient way, and has been well described. Finally, the plotting and interface machinery of the shapelets package have been presented.

Having introduced shapelets, we are now in a position where we can use them for weak lensing data analyses. In Volume II, we explain how we proceed, from the detection of galaxies on an image, to the measure of the shear and the reconstruction of the dark matter distribution. In particular, we present our shapelets-based PSF modelling scheme. We also describe the practical use that we make of the shapelets package. That is, we give a full overview of our whole shapelets based weak lensing image processing.

The shapelets package can be downloaded on the Internet at :
http://www.astro.caltech.edu/∼rjm/shapelets/code/

Appendix A

Routines of the shapelets package

In this appendix, we list all the routines belonging to the shapelets package (version 2.1β), together with their purpose. Each (sub)section corresponds to a (sub)directory of the package, the name of which is given between brackets.

A.1 General routines

shapelets_paths : This stores the locations of data on a locally accesible hard disc. Please update this file so that all the strings point to the correct locations !

shapelets_demo : Runs through the main shapelet routines to demonstrate their use and to check that they are installed correctly

A.2 Alias routines (alias)

plt_chi2_grid : alias for shapelets_plot_chisq_grid

plt_colbar : alias for shapelets_plot_colourbar

plt_decomp : alias for shapelets_plot_decomp

plt_decomp_polar : alias for shapelets_plot_decomp_polar

plt_focus : alias for shapelets_plot_focus

plt_image : alias for shapelets_plot_image

plt_imstat : alias for shapelets_plot_image_statitics

plt_obj : alias for shapelets_plot

plt_polar : alias for shapelets_plot_image_polar

plt_pstamp : alias for shapelets_plot_pstamp

plt_scat : alias for shapelets_plot_sexcat

plt_sexcat : alias for shapelets_plot_sexcat

plt_shcat : alias for shapelets_plot_shapecat

shapelets_c2p : alias for shapelets_polar_convert, cartesian to polar

shapelets_concatenate_shapecats : alias for shapelets_add (two shapecats)

shapelets_p2c : alias for shapelets_polar_convert, polar to cartesian

shapelets_shapecat_c2p : alias for shapelets_polar_convert

shapelets_shapecat_nmax : Alter $n_{\max}$ in a shapelet catalogue. (obsolete)

shapelets_split_sexcat : alias for shapelets_split

shapelets_split_shapecat : alias for shapelets_split

shapelets_write_shapecat : alias for shapelets_write

A.3 Decomposition and focus (decomp)

shapelets_chi : Compute (dimensionless) 2D polar shapelet basis functions $\chi_{nm}(x, y)$, based on Laguerre polynomials. Dimensional basis functions can be calculated with shapelets_chi(nm,x1/beta[,x2/beta])/beta.

shapelets_decomp : Decompose a function f into ϕ_n basis functions and output the coefficients in an IDL structure.

shapelets_hermite : Compute the polynomial coefficients for $H_n(x)$, the 1D Hermite polynomial of order n

shapelets_make_ls_matrix : Create an array with which to compute the least-squares linear algebra fit of shapelets basis functions to data. Can also use this array to perform the overlap integrals calculation.

shapelets_make_nvec : Set up look-up tables to specify which number in a vector of shapelet coefficients corresponds to which n_1 and n_2, or which n_r and n_l.

shapelets_make_xarr : Make arrays containing the values of the x ($x1$) and y ($x2$) coordinates of a grid. This is convenient for evaluating 2D functions (e.g. shapelet basis functions) on a grid.

shapelets_phi : Compute (dimensionless) Cartesian shapelet basis functions $\phi(x)$ in 1D or 2D, based on Hermite polynomials. Dimensionful basis functions can be calculated with shapelets_phi(n,x1/beta[,x2/beta])/beta.

shapelets_recomp : Compute the recomposed image corresponding to a set of shapelet coefficients calculated using shapelets_decomp.pro. The input decomp structure also contains a few meta-parameters, including the shapelet scale size beta, and whether or not the basis functions should be integrated within pixels, or merely evaluated at the centre of each pixel.

A.3.1 Focus routines (focus)

shapelets_focus : Find the optimal β, $n_{\max}$ and centroid for a shapelet decomposition via an iterative search throughout the space spanned by these (meta-) variables.

shapelets_focus_beta : Find the optimal beta and centroid centre_guess for the decomposition of an image by minimising χ^2. This is done using a 1-dimensional Amoeba search. The maximum shapelet order $n_{\max}$ is assumed to be known.

shapelets_focus_nmax : Find the optimal $n_{\max}$ for the decomposition of an image by exploring different values until χ^2 is equal to a set value χ_0^2. The shapelet scale β is assumed to be known.

shapelets_geometric_constraints : Determine $\theta_{\min}$ and $\theta_{\max}$ for an object.

shapelets_guess_nmaxbeta : Guess the optimal β and $n_{\max}$ for an object.

shapelets_make_chi2_grid : Compute the χ^2 difference between the input and the reconstructed image on a grid of values of β and $n_{\max}$.

shapelets_make_chisq_grid : Compute the χ^2 difference between the input and the reconstructed image on a grid of values of β and $n_{\max}$.

A.3.2 Polar shapelets decomposition (polar)

shapelets_polar_convert : Converts Cartesian shapelet coefficients to their equivalent polar shapelet representation, or vice versa.

shapelets_polar_expand : Manufactures and reinserts the degenerate complex parts of polar shapelet coefficients so they can be transformed back into cartesian shapelets.

shapelets_polar_matrix : Calculates a matrix to convert Cartesian shapelet coefficients to their equivalent polar shapelet representation.

shapelets_polar_reduce : Removes the duplicated/degenerate coefficients when cartesian shapelets have been converted into polar form and leaves only the minimum number of independent parameters.

A.4 Inputs/Outputs (io)

shapelets_create_decomp : Initiliase a brand new decomp structure. Works like an IDL_DEFINE procedure, but for an anonymous structure (which we need because they will shrink or expand to accomodate varying amounts of data).

shapelets_create_focus : Initiliase a brand new decomp structure. Works like an IDL_DEFINE procedure, but for an anonymous structure (which we need because they will shrink or expand to accomodate varying amounts of data).

shapelets_read_image : Read in fits image, plus segmentation and inverse variance pixel map if available (it looks for the same filename, but with _objects and _weight added before the extension).

shapelets_read_psf : Creates a shapelet decomp structure representing a PSF (this PSF will eventually be the same across one image).

shapelets_read_sexcat : Reads in a SExtractor catalogue.

shapelets_read_shapecat : Reads in a shapelet coefficient catalogue of objects, created by shex.pro, then stores them in a shapecat IDL structure. This may then be converted into decomp structures using shapelets_shapecat2decomp.pro.

shapelets_sexcat2pstamp : Extract a postage stamp of image data around an object detected by SExtractor data. Uses the fact that the inverse variance is zero on saturated stars.

shapelets_shapecat2decomp : Extracts one (decomp structure) object from a (shapecat structure) catalogue.

shapelets_structure_type : Determine whether a variable is a known shapelets structure, in a very robust manner.

shapelets_update_history : Append a string to an object's history record

shapelets_write : Writes a shapelet coefficient catalogue of objects to disc.

NB : *all these routines work with binary catalogue files*

A.4.1 Deal with ascii catalogues (asciicatalogues)

shapelets_ascii2idl_shapecat : Converts a shapecat stored on disc from a shapelets .shape file (in ASCII format) to a shapelets .shapelet file (in IDL SAVE format).

shapelets_concatenate_ascii_shapecats : Concatenates two (shapecat structure) catalogues.

shapelets_idl2ascii_shapecats : Converts a shapecat stored on disc from a shapelets .shapelet file (in IDL SAVE format) to a shapelets .shape file (in ASCII format).

shapelets_read_ascii_sexcat : Reads in a SExtractor catalogue.

shapelets_read_ascii_shapecat : Reads just the header information in shapelet catalogues.

shapelets_read_ascii_shapecat_hdf : Reads in and concatenates shapelet catalogues containing all of the objects from both Hubble Deep Fields.

A.5 Library of useful routines (library)

check_fits : Check that keywords in a FITS header array match the associated data

comb : Calculates binomial coefficients C_n^r

cond : This function computes the condition number of an N by N array.

congrid : Shrink or expand the size of an array by an arbitrary amount.

convolve : Convolution of an image with a Point Spread Function (PSF)

daycnv : Converts Julian dates to Gregorian calendar dates

delvarx : Delete variables for memory management (can call from routines)

determ : This function computes the determinant of an N by N array.

factorial : This function computes the factorial N! as the double-precision product, (N) * (N-1) * (N-2) * ...... * 3 * 2 * 1.

fdecomp : Routine to decompose a file name for any operating system

fits_close : Close a FITS data file

fits_help : To print a summary of the primary data units and extensions in a FITS file.

fits_info : Provide information about the contents of a FITS file

fits_open : Opens a FITS (Flexible Image Transport System) data file.

fits_read : To read a FITS file.

fits_write : To write a FITS primary data unit or extension.

fxaddpar : Add or modify a parameter in a FITS header array.

fxmove : Skip a specified number of extensions in a FITS file

fxpar : Obtain the value of a parameter in a FITS header.

fxposit : Return the unit number of a FITS file positioned at specified extension

gamma : Computes the Gamma function.

gauss2dfit : Fit a 2 dimensional elliptical gaussian equation to rectilinearly gridded data.

get_date : Return the (current) UTC date in CCYY-MM-DD format for FITS headers

gettok : Retrieve the first part of the string up to a specified character

idl_validname : Modify a string if necessary, so that it can used as a IDL variable name.

ieee_to_host : Translate an IDL variable from IEEE-754 to host representation

inside : see if point is inside polygon

is_ieee_big : Determine if the current machine is use IEEE, big-endian numbers.

loadct : Load predefined color tables.

match : Routine to match values in two vectors.

mean : This function computes the mean of an N-element vector.

mkhdr : Make a minimal primary (or IMAGE extension) FITS header

mmm : Estimate the sky background in a stellar contaminated field.

moment : This function computes the mean, variance, skewness and kurtosis of an N-element vector.

mrd_hread : Reads a FITS header from an opened disk file or Unix pipe

mrd_skip : Skip a number of bytes from the current location in a file or a pipe

mrd_struct : Return a structure as defined in the names and values data.

mrdfits : Read all standard FITS data types into arrays or structures.

numlines : Return the number of lines in a file

path_sep : Return the proper file path segment separator character for the current operating system. This is the character used by the host operating system for delimiting subdirectory names in a path specification. Use of this function instead of hardwiring separators makes code more portable.

pca : Carry out a Principal Components Analysis (Karhunen-Loeve Transform)

plothist : Plot the histogram of an array with the corresponding abcissa.

plt_ellipse : plot one or several ellipse centered at (x,y) and with major and minor axes a and b, respectively, and with position angle pa

plt_rectangle : Draws a rectangle on the current output device.

randome : Generates a random number (or nxmxl array of) from a rescaled Epanechnikov kernel $K(x) = 0.75(1 - x^2)$ for $|x| < 1$; $K(x) = 0$ elsewhere.

readcol : Read a free-format ASCII file with columns of data into IDL vectors

remchar : Remove all appearances of character (char) from string (st)

repchr : Replace all occurrences of one character with another in a text string.

reverse : Reverse the order of rows or columns in an array or vector.

sign : Compute the sign(x) of a variable x, with sign(x)=0,+1,-1, for x=0,¿0,¡0 respectively. The variable x can have any dimension.

sky : Determine the sky level in an image using the the procedure MMM

spec_dir : Complete a file specification by appending the default disk or directory

stddev : This function computes the stddev of an N-element vector.

strnumber : Function to determine if a string is a valid numeric value.

sxaddpar : Add or modify a parameter in a FITS header array.

sxdelpar : Procedure to delete a keyword parameter(s) from a FITS header

sxpar : Obtain the value of a parameter in a FITS header

tag_exist : To test whether a tag name exists in a structure.

valid_num : Check if a string is a valid number representation.

writefits : Write IDL array and header variables to a disk FITS file.

xgen : Generate a vector containing equally spaced numbers between x1 and x2. This is typically used to generate an x-axis vector in a plot.

zparcheck : Routine to check user parameters to a procedure

A.6 Shapelets algebra (operations)

shapelets_add : Superimposes two images (stored as shapelet decompositions), append a shapelet decomposition to a shapelet catalogue, or concatenates two catalogues.

shapelets_circularise : Circularises objects by setting to zero all of their polar shapelet coefficients where m is nonzero. When averaged around all angles, the negative parts of these basis states cancel out the positive parts.

shapelets_convolution_matrix : Create an array of which unconvolved basis functions (with scale alpha) combine to create a basis function convolved with a (decomp structure) PSF (and scale gamma).

shapelets_convolve : Convolves one image with another (*e.g.* a PSF).

shapelets_detect_diamond : Detect if a shapelet structure has been created using the diamond shape or not

shapelets_dilate : Dilates (changes the size of) an object. There are four methods available to perform this operation:

1. Simply rescale beta, the shapelet size parameter.
2. To first order, in Cartesian shapelets, using $a\&a^{\mathrm{ladde}}r$ operators.
3. Convert to polar shapelets and use ladder operators, but again only to first order. Then convert back to Cartesian coefficients.
4. An alternative method uses the C_{nml} formalism for convolution. Convolution with a delta function (*i.e.* a PSF with $\beta \to 0$) leaves the object unchanged, but we can now choose a new β (in shapelets_convolution_matrix.pro, the notation is to change alpha to gamma). This is a general case, no longer limited to first order.

Option 4 is the default; the others are selected using switches.

shapelets_exponentiate_operations : Applies a (pure) shear to an object, using ladder operators in (Cartesian) shapelet space.

shapelets_extend_nmax : Increases the value of $n_{\max}$ in a structure by n_{increase}, by padding the coefficient arrays with zeros. Can also decrease $n_{\max}$ by truncating the coefficient arrays.

shapelets_flex : Applies a flexion (the slight bending due to gradients in a shear field) to one object in a decomp structure, or to all objects in a shapecat catalogue.

shapelets_keep_it_real : Makes sure a polar shapelet model is wholly real by discarding any imaginary part. Has no effect on a Cartesian shapelet model.

shapelets_recentre : Applies a translation to an object to ensure that its centre of light lies exactly on the origin (the centre of the basis functions).

shapelets_reflect : Makes a mirror image of (an) object(s) by swapping parity.

shapelets_rotate : Rotates an object by converting it's Cartesian shapelet coefficients to polar shapelet coefficients, then applying the rotation matrix - which, in this basis, is a simple multiplication.

shapelets_shear : Applies a (pure) shear to one object in a decomp structure, or to all objects in a shapecat catalogue.

shapelets_split : Splits a (IDL shapecat or sexcat structure) catalogue.

shapelets_subtract : Subtracts one image from another (when both are stored as a shapelet decomposition).

shapelets_translate : Translates an object (to first order, using ladder operators) some number of pixels.

A.7 Pipeline (pipeline)

config : subdirectory containing SExtractor configuration files.

sex : Runs SExtractor (by spawning it as a child process under the shell used to run IDL) to locate objects within an image.

shapelets_interpolate_psf : Fourier interpolation of the PSF (*not dealt with neither in Volume I nor in Volume II*).

shapelets_interpolate_psf_lsmatrix : Fourier interpolation of the PSF (*not dealt with neither in Volume I nor in Volume II*).

shapelets_make_image_mask : Create a binary mask of an image that is zero everywhere, except for ones around saturated or potentially corrupted regions. (*not used in this manual, neither in Volume I nor in Volume II*).

shapelets_select_stars : Interactively select various different object types. (*not used in this manual, neither in Volume I nor in Volume II*).

shapelets_wl_pipeline : As automatically as possible, do everything needed to create a PSF-corrected shapelet catalogue from an image, to measure weak lensing. (*this is is not exactly the pipeline described in Volume II !!!*)

shex : Having located objects within an image using SExtractor, this routine now decomposes all of the objects into shapelets. It uses the shapelets focus suite of routines to optimise the nmax, beta and centroid parameters. All of the decompositions are written to disc in a shapelets catalgoue. This can be later read to memory using shapelets_read_sexcat.pro

A.8 Plotting routines (plotting)

extract_pixsc : extract the value of PIXEL_SCALE from SExtractor configuration files

plt_gals : plot various figures relative to the galaxies in a catalog.

plt_gals_shcat : plot various figures relative to the galaxies in a shapelet catalog (shcat).

plt_objs : plot the results of the focussing of shapelets on an object. The history of the search of optimal shapelet parameters is also shown.

plt_scat_mom : plot statistics for the scat_mom catalog. In particular, compare derived moments from the image to the input catalog

plt_sgsep : plot the various figures relative to the star-galaxy separation for a catalog of objetcs

plt_shear : plot shear estimator statistics

plt_shear_scat : plot shear statistics derived from sextractor catalog. This is particularly useful if the sextractor catalog is is scat_in input catalog of a simulation

plt_stars : plot various figures relative to the stars in a catalog.

shapelets_plot : Generic plotting routine. Acts as a wrapper for many more specific plotting utilities. First decides what the structure type is, rather as you would expect from object-oriented code, then calls the relevant plotting subroutine.

shapelets_plot_basis : Makes a plot of all the polar shapelet basis functions out to $n_{\max}$

shapelets_plot_chisq_grid : plot χ^2 on the $\beta - n_{\max}$ grid

shapelets_plot_colourbar : draw an annotated color bar at the top of the plotting window. The window parameters are then set to the remainder of the window.

shapelets_plot_decomp : Repixellate a shapelet model (decomp structure) and display it.

shapelets_plot_focus : Plot the route taken through possible $n_{\max}$ and β values to obtain a final decomposition.

shapelets_plot_image : Draws a 2D pixellated image.

shapelets_plot_image_statitics : Plot statistics about an image.

shapelets_plot_pstamp : plot the image and sextractor parameters for an object

shapelets_plot_sexcat : Displays an image (if input) and overlays the positions of all objects found on it by SExtractor. If no image structure is given, it just plots the positions of objects.

shapelets_plot_shapecat : plot statistics relative to shapelet catalog

A.9 Objects properties (properties)

shapelets_asymmetry : Returns asymmetry morphology index of a Cartesian decomp structure.

shapelets_centroid : Computes the centroid from a linear summation of shapelet coefficients, read in from a Cartesian decomp structure.

shapelets_concentration : Returns concentration morphology index of a Cartesian decomp structure.

shapelets_ellipticity : Calculates the ellipticity of a shapelet model.

shapelets_flux : Returns total flux of a Cartesian decomp or shapecat structure.This does the same as shapelets_moments.pro but in function form.

shapelets_image_moments : Compute the zeroth (flux) and first (centroid) moment of an image.

shapelets_moments : Compute the zeroth (flux) and first (centroid) moment for a basis decomposition. Also computes the characteristic order n_f and scale parameter β_f.

shapelets_profile : Compute the azymuthally averaged profile f(r) of an object from its shapelet coefficients.

shapelets_quadrupole : Calculates the unweighted quadrupole moments of a shapelet model.

shapelets_rsquared : Computes the R^2 size measure from a linear summation of shapelet coefficients, read in from a Cartesian decomp structure.

Bibliography

[1] Bacon D., Refregier A. & Ellis R., 2000,MNRAS,318,625

[2] Bartelmann M. & Schneider P. Physics Report 340(2001) 291-472

[3] Bekenstein J., 2004, Physics ReviewD,70h3509B

[4] Bernstein G. & Jarvis M., 2002, AJ 123, 583

[5] Bertin E., SExtractor User's Manual

[6] Bertin E. & Arnouts,S.,1996,A&AS,117,393

[7] Bridle S., Gull S., Bardeau S. & Kneib J.P., 2001, in Scientific NW., ed, Proceedings of the Yale Cosmology Workshop The Shapes of Galaxies and their Dark Halos

[8] Fort B. & Mellier Y.,1994, *Astron. Astrophys. Rev.*,5:239

[9] Heavens A., Refregier A. & Heymans C., 2000, MNRAS, 319, 649

[10] Heymans C. *et al* , 2006, MNRAS, in press, astro-ph/0506112

[11] Heymans C. *et al* , 2006, astro-ph/0604001

[12] Hirata C. & Seljak U., 2004, Physical ReviewD, 70, 063526

[13] Holwerda B., SExtractor for Dummies, available at http://www-int.stsci.edu/∼holwerda/se.html

[14] Kaiser N., 2000, ApJ 537, 555

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

[16] Kaiser N., Wilson G. & Luppino GA. 2000

[17] Landsman W., http://idlastro.gsfc.nasa.gov/

[18] Lupton R., 1993, Statistics in theory and practice, Princeton University Press

[19] Mandelbaum R. *et al* , 2006, MNRAS, 367, 611

[20] Massey R. & Refregier A., 2005, MNRAS, 363, 197

[21] Massey R. & Refregier A., Shapelets Web Page, http://www.astro.caltech.edu/∼rjm/shapelets/

[22] Massey R., Refregier A. & Bacon D., 2004, astro-ph/0408458,To be published in "Impact of Gravitational Lensing on Cosmology", IAU Symposium 225, eds. Mellier & Meylan

[23] Mellier Y., 1999,*Annu. Rev. Astron. Astrophys.*, 37:127

[24] Press W., Flannery B., Teukolsky S. & Vetterling W., 2002, Numerical Recipes in C/C++, Cambridge University Press

[25] Refregier A., 2003,*Annu. Rev.Astron. Astrophys.*,41:645

[26] Refregier A., 2003, MNRAS, 338,35

[27] Refregier A. & Bacon D.,2003, MNRAS, 338,48

[28] Rhodes J., Refregier A. & Groth E.,2000,ApJ 536,79

[29] Smith D., Bernstein G., Fisher P. & Jarvis M., 2001, ApJ 551, 643

[30] Soucail G., Fort B., Mellier Y. & Picat J.B.,1987a,*Astron. Astrophys.*,172,L14

[31] Soucail G.,Mellier Y., Fort B. *et al* , 1987b,*Astron. Astrophys.*,184,L7

[32] Subaru Collaboration, 2004, http://www.subarutelescope.org

[33] van Waerbeke L., Mellier Y., Pelló R. *et al* ,2000,*Astron. Astrophys.*,393:369

Index