

Article

Using Spherical-Harmonics Expansions for Optics Surface Reconstruction from Gradients

Juan Manuel Solano-Altamirano ^{1,*}, Alejandro Vázquez-Otero ^{2,*}, Danila Khikhlikha ^{3,*}, Raquel Dormido ⁴ and Natividad Duro ⁴

¹ Facultad de Ciencias Químicas, Benemérita Universidad Autónoma de Puebla, 14 Sur y Av. San Claudio, Col. San Manuel, Puebla 72520, Mexico

² MSD IT Global Innovation Center s.r.o., Svornosti 3321/2, 150 00 Prague 5, Czech Republic

³ ELI Beamlines, Institute of Physics ASCR, Za Radnicí 835, 252 41 Dolní Břežany, Czech Republic

⁴ Department of Computer Sciences and Automatic Control, UNED, C/Juan del Rosal, 16, 28040 Madrid, Spain; raquel@dia.uned.es (R.D.); nduro@dia.uned.es (N.D.)

* Correspondence: jmsolanoalt@gmail.com (J.M.S.-A.); alejandro.vazquez.otero@merck.com (A.V.-O.); danila.khikhlikha@eli-beams.eu (D.K.)

Received: 12 October 2017; Accepted: 25 November 2017; Published: 30 November 2017

Abstract: In this paper, we propose a new algorithm to reconstruct optics surfaces (aka wavefronts) from gradients, defined on a circular domain, by means of the Spherical Harmonics. The experimental results indicate that this algorithm renders the same accuracy, compared to the reconstruction based on classical Zernike polynomials, using a smaller number of polynomial terms, which potentially speeds up the wavefront reconstruction. Additionally, we provide an open-source C++ library, released under the terms of the GNU General Public License version 2 (GPLv2), wherein several polynomial sets are coded. Therefore, this library constitutes a robust software alternative for wavefront reconstruction in a high energy laser field, optical surface reconstruction, and, more generally, in surface reconstruction from gradients. The library is a candidate for being integrated in control systems for optical devices, or similarly to be used in ad hoc simulations. Moreover, it has been developed with flexibility in mind, and, as such, the implementation includes the following features: (i) a mock-up generator of various incident wavefronts, intended to simulate the wavefronts commonly encountered in the field of high-energy lasers production; (ii) runtime selection of the library in charge of performing the algebraic computations; (iii) a profiling mechanism to measure and compare the performance of different steps of the algorithms and/or third-party linear algebra libraries. Finally, the library can be easily extended to include additional dependencies, such as porting the algebraic operations to specific architectures, in order to exploit hardware acceleration features.

Keywords: wavefront reconstruction from gradients; surface reconstruction from gradients; spherical harmonics; zernike-polynomials; algorithm

1. Introduction

One of the main motivations to improve surface reconstruction techniques is to enhance wavefront sensors capabilities, which are instruments used to measure aberrations of incident wavefronts. In high-energy lasers production, these aberrations, i.e., imperfections, reduce the wavefront quality, and may also add undesirable effects to the produced beams, which, in turn, may affect the quality of the experimental outcomes wherein the laser is used. Furthermore, a wide range of technical fields benefit from accurate and efficient surface reconstruction algorithms, for instance the astronomical community, pioneering in developing wavefront reconstruction techniques for telescopes [1], measurements of eye aberrations [2], optical devices manufacturers, wherein the reconstruction plays a key role by identifying lens manufacturing errors, adaptive optics (AO) such as in microscopy [3] or data communication

through the Earth's atmosphere [4], lateral shearing interferometry [5–7], shape from shading [8], high energy laser (HEL) beam production control [9], or ophthalmology in refractive surgery [10].

In high-energy lasers production, opticians strongly prefer the Zernike polynomial set to reconstruct wavefronts and to decompose imperfections into well-known aberration components [11]. However, other sets have been proposed to be used for reconstructing surfaces (see, for instance, [12]). On the bright side, the Zernike-based reconstruction has been shown to outperform the iterative Fourier when reconstructing wavefront aberrations from slope data. However, on the other hand, noncircular pupils has posed a challenge to Zernike-based reconstruction, as the performance of this set for this pupils is questionable [13]. Furthermore, although Zernike polynomials constitute a complete set, hence any wavefront aberration can be decomposed in terms of them, and advantages of orthogonality are lost in noncircular cases. Moreover, the coefficients lose their physical meaning since the circle polynomials do not represent balanced aberrations for the noncircular pupil. In the library distributed along with this work, we provide a simple reconstruction algorithm based on the Legendre polynomial set for reconstructing surfaces in squared domains.

In addition, in producing high-energy lasers, the adaptive optics loop is critical, as in this process the quality of the wavefront is determined. The basic goal of the adaptive optics is easily stated: to measure the aberrations of an incoming wavefront and then cancel these out by applying compensating aberrations, all in real time. Therefore, it is desirable for the reconstruction algorithm to be as fast and accurate as possible. Unfortunately, high accuracy usually implies higher computational cost. The algorithm that we propose in this work, using Spherical Harmonics, may offer an alternative to the classical Zernike-based reconstruction algorithms, especially for automated processes wherein human intervention and analysis are not needed, but speed is, e.g., in adaptive optics [14].

The paper is organized as follows. In Section 2, we roughly describe how a Shack–Hartmann wavefront sensor works. In Section 3, we provide an overview of the modal reconstruction algorithm, and the generalities of our implemented version in OpenWavefrontReconstructor. The mathematical details of our new algorithm, using the Spherical Harmonics for circular domains, to reconstruct surfaces from gradients are given in Section 4. We also provide a few implementation details relative to the reconstruction algorithm that use classical Zernike polynomials for circular domains, and Legendre for square domains, in Sections 5 and 6. In Section 7, we describe some general features of OpenWavefrontReconstructor. Numerical results and discussion are given in Section 8 and finally we close with conclusions and future work in Section 9.

2. Shack–Hartmann Wavefront Sensor

We will focus on improving the algorithms used internally by Shack–Hartmann wavefront sensors (the reader is referred to [2] and references therein for technical details on early designs). Roughly speaking, a wavefront sensor consists of three main parts: (a) an opto-mechanical device intended to provide measurements of the original light wavefront to be reconstructed; (b) its associated processing electronics; and (c) the software responsible for the wavefront reconstruction.

The optomechanical part of a Shack–Hartmann wavefront sensor consists of a lenslet array and a light-sensing device (usually a charge-coupled device (CCD) or complementary metal-oxide-semiconductor (CMOS) camera). When a light wavefront hits the lenslet array (see Figure 1a), it generates a grid of small light spots (aka the spotfield, see Figure 1b) that are recorded by the light-sensing device. Through the intensity and shape of each light spot, the position of each center of mass can be found. In addition, if the wavefront is not plane, the produced spots deviate from the vertical position a distance Δx (see Figure 1a). Therefore, the wavefront slopes, dw/dx and dw/dy , can be evaluated through the distance between the lenslets and the sensor device, z (see Figure 1a), and Δx . These slopes constitute the information used to determine the shape of the wavefront through a reconstruction algorithm [2].

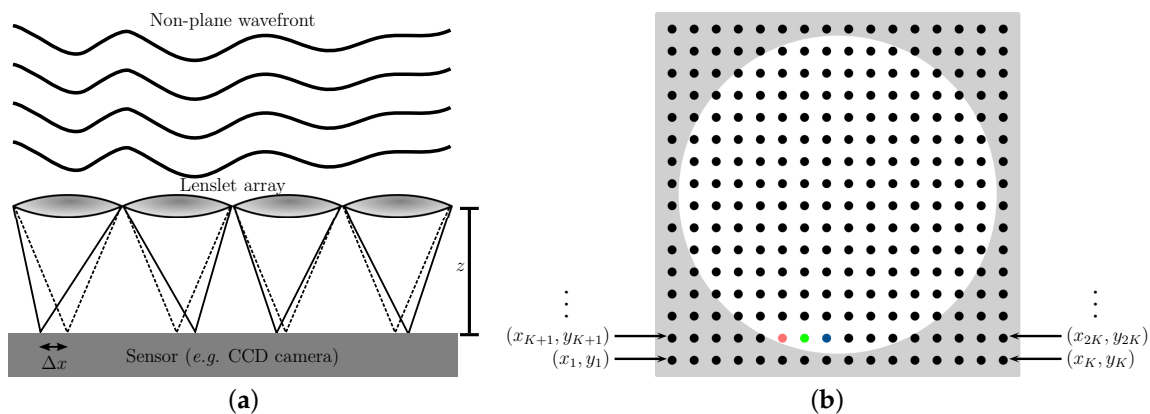


Figure 1. (a) Side view of a simplified Shack–Hartmann wavefront sensor scheme; (b) top view of the spotfield produced by a perfectly plane wavefront hitting a lenslet array.

In this work, we will be focused on contributing to the development of such wavefront reconstruction algorithms. However, the methods can also be used to reconstruct any surface whose slopes (or gradients) are known at a set of points. For this, one simply replaces the focal spots' coordinates by the coordinates where the slopes are measured.

The spotfield produced by a wavefront sensor is commonly trimmed to fit a circular shape, such as the white zone depicted in Figure 1a. This is so because, in optics, most of the lenses used to focus light are circular; hence, the area illuminated by a focused light beam is circular as well. Nonetheless, the sensing device usually can record information in square shapes, such as the complete spotfield shown in Figure 1a, and even on rectangular shapes. Furthermore, noncircular shapes occur frequently; for instance, the cross section of a laser beam in high-power laser facilities is often square [15].

Since there exists a relation between the shape of the illuminated area and the domain whereupon polynomials sets are defined, in this paper, we will generically denote all shapes with the symbol S , and we will refer to both shape and actual domain indistinctively as domains. These domains are relevant to us because any set of orthogonal polynomials is complete only on a given domain.

3. Wavefront Reconstruction Method

There exist several methods to reconstruct a wavefront (or any surface) from slopes. In this paper, we will use the modal reconstruction method, as this is commonly used in Shack–Hartmann wavefront sensors. The modal reconstruction method is built upon the idea that a wavefront incident on a sensing surface, S , can be recovered from slopes sampled at a set of points.

Furthermore, it is assumed that the wavefront, here denoted by $w(x, y)$, can be decomposed into a polynomial series truncated to the term J :

$$w(x, y) = \sum_{\alpha=1}^J A_{\alpha} \Xi_{\alpha}(x, y). \quad (1)$$

Here, $\{\Xi_{\alpha}(x, y)\}$ denotes a generalized set of functions (which is usually a set of polynomials orthogonal on the domain S), A_{α} the coefficients of the expansion, and α an ordering index that depends on the specific set $\{\Xi_{\alpha}(x, y)\}$. Traditionally, if S is a square, then

$$\Xi_{\alpha}(x, y) = P_k(x)P_l(y), \quad (2)$$

where $P_l(t)$ are the Legendre Polynomials, and $\alpha = \alpha(k, l)$ is some ordering function of the indices k and l . In the same fashion, if S is a circle, then

$$\Xi_{\alpha}(x, y) = Z_{l,m}(r, \theta). \quad (3)$$

Here, $Z_{l,m}$ are the classical Zernike polynomials [11], whose radial degree is l , and $\alpha = \alpha(l, m)$ is the respective ordering function of the indices l and m (see [16]). The specific form of α depends specifically on the type of polynomials used to expand the wavefront, and there is no universal convention (see Sections 4–6 for our specific implementation).

If Equation (1) is valid for a wavefront, $w(x, y)$, its gradient at any point (x, y) is given by:

$$\nabla w(x, y) = \sum_{\alpha} A_{\alpha} \left(\frac{\partial \Xi_{\alpha}(x, y)}{\partial x} \hat{i} + \frac{\partial \Xi_{\alpha}(x, y)}{\partial y} \hat{j} \right), \quad (4)$$

where \hat{i} and \hat{j} are unit vectors in x and y directions, respectively. Therefore, if $\nabla w(x, y)$ is known at a set of points, then the coefficients of the polynomial expansion can be recovered (see below). These slopes are precisely the information measured by a wavefront sensor.

3.1. Matrix Assembly

From the spot coordinates, (x_i, y_i) (see Figure 1b), we can assemble an array to contain the slopes of the wavefront at all sampled points (x_i, y_i) , as follows:

$$\mathbf{G} \equiv \begin{bmatrix} \frac{\partial w(x_i, y_i)}{\partial x} \\ \frac{\partial w(x_i, y_i)}{\partial y} \end{bmatrix} = \begin{bmatrix} \sum_{\alpha} A_{\alpha} \frac{\partial \Xi_{\alpha}(x_1, y_1)}{\partial x} \\ \vdots \\ \sum_{\alpha} A_{\alpha} \frac{\partial \Xi_{\alpha}(x_1, y_1)}{\partial x} \\ \sum_{\alpha} A_{\alpha} \frac{\partial \Xi_{\alpha}(x_1, y_1)}{\partial y} \\ \vdots \\ \sum_{\alpha} A_{\alpha} \frac{\partial \Xi_{\alpha}(x_I, y_I)}{\partial y} \end{bmatrix}. \quad (5)$$

In Equation (5), the coefficients A_{α} can be fitted to match the experimental slopes, and i is an index related to each focal spot (see Figure 1b). In the rest of this section, we will assume that our expansion has J terms, i.e., as in Equation (1), and that there are I sampled points whose coordinates are (x_i, y_i) , $i = 1, \dots, I$.

Similarly as for the ordering index α , there is no universal convention regarding the order of the coordinates of the focal spots. For square domains, the order of the nodes may be given as indicated in Figure 1b (here the grid has $I = K \times K$ spotlights), and for circular domains, the numbering may be given as (x_1, y_1) [red point], (x_2, y_2) [green point], (x_3, y_3) [blue point], etc. Fortunately, reconstruction algorithms do not depend on the specific ordering system of the focal spots; however, they depend on how exactly the coordinates of each node are known.

Equation (5) can be rewritten as:

$$\mathbf{G} = \mathbf{M}\mathbf{A}. \quad (6)$$

Here, \mathbf{A} is the array of the expansion coefficients A_{α} :

$$\mathbf{A} = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_J \end{pmatrix}, \quad (7)$$

and the matrix \mathbf{M} (whose dimension is $2I \times J$) is constructed as:

$$\mathbf{M} = \begin{pmatrix} \frac{\partial \Xi_1(x_1, y_1)}{\partial x} & \cdots & \frac{\partial \Xi_J(x_1, y_1)}{\partial x} \\ \vdots & \ddots & \vdots \\ \frac{\partial \Xi_1(x_I, y_I)}{\partial x} & \cdots & \frac{\partial \Xi_J(x_I, y_I)}{\partial x} \\ \frac{\partial \Xi_1(x_1, y_1)}{\partial y} & \cdots & \frac{\partial \Xi_J(x_1, y_1)}{\partial y} \\ \vdots & \ddots & \vdots \\ \frac{\partial \Xi_1(x_I, y_I)}{\partial y} & \cdots & \frac{\partial \Xi_J(x_I, y_I)}{\partial y} \end{pmatrix}. \quad (8)$$

3.2. Least-Squares Method and Singular Value Decomposition

The coefficients A_α can be found by the least-squares method, applied to Equation (6). For this, in the C++ library distributed along with this paper, we use the Singular Value Decomposition theorem [17,18], which states that any $m \times n$ real matrix \mathbf{M} can be decomposed as

$$\mathbf{M} = \mathbf{U}\mathbf{S}\mathbf{V}^T, \quad (9)$$

wherein \mathbf{U} ($= [\mathbf{u}_1 \cdots \mathbf{u}_n]$) is an $n \times n$ matrix that orthogonally diagonalizes $\mathbf{M}^T\mathbf{M}$, \mathbf{V} is an $m \times m$ orthogonal matrix, and the non-zero diagonal elements of the $m \times n$ matrix \mathbf{S} are the non-zero eigenvalues of $\mathbf{M}^T\mathbf{M}$ corresponding to the column vectors (\mathbf{u}_i) of \mathbf{U} . Therefore, the coefficients of the expansion are given by

$$\mathbf{A} = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T\mathbf{G}, \quad (10)$$

and the wavefront at any point is thus given by Equation (1).

In version 1.0.0 of OpenWavefrontReconstructor, we provide an implementation that uses the armadillo library [19] for solving Equation (10). The user may re-implement some of the functions of the library, should he/she desire to use a different linear algebra library. Optionally, the user may implement another third-party linear algebra library of his/her choice.

3.3. Wavefront Retrieving

Once the coefficients of the expansion, A_α , are known, the values of the wavefront (or surface) at the sampled points, $\{w(x_i, y_i)\}$, can be recovered through the following matrix equation:

$$\mathbf{W} = \begin{pmatrix} w(x_1, y_1) \\ \vdots \\ w(x_I, y_I) \end{pmatrix} = \mathbf{R}\mathbf{A} = \begin{pmatrix} Z_1(x_1, y_1) & \cdots & Z_J(x_1, y_1) \\ \vdots & \ddots & \vdots \\ Z_1(x_I, y_I) & \cdots & Z_J(x_I, y_I) \end{pmatrix} \begin{pmatrix} A_1 \\ \vdots \\ A_J \end{pmatrix}. \quad (11)$$

Here, if there are I sampled points, (ξ_i, η_i) , the matrix \mathbf{R} has dimensions $I \times J$. Of course, the wavefront can also be obtained at any point through Equation (1).

A performance remark: once the matrices $\mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T$ and \mathbf{R} are generated during the first reconstruction, subsequent reconstructions can be carried out with a much smaller computational cost, using the same matrices. This is valid only if the focal spots' coordinates do not change from reconstruction to reconstruction, which is true for wavefront sensors. For instance, the coefficients of a new wavefront expansion, \mathbf{A}' , whose slopes are given by a new vector \mathbf{G}' , are computed as $\mathbf{A}' = \mathbf{V}\mathbf{S}^{-1}\mathbf{U}^T\mathbf{G}'$, and therefore the respective wavefront values are given by $\mathbf{W}' = \mathbf{R}\mathbf{A}'$.

4. Half Circular Harmonics

In this section, we describe the design and implementation of the new algorithm for reconstructing wavefronts using Spherical Harmonics as the orthogonal polynomial set mapped onto a circle. For this, we apply four consecutive mappings χ_0, χ_1, χ_2 , and χ_3 , which transforms the original coordinates

until the coordinates are suitable for using half of the domain of the Spherical Harmonics (see Figure 2). The complete mapping from (x, y) to (μ, φ) is given by $\chi_3 \circ \chi_2 \circ \chi_1 \circ \chi_0$, and we will refer to this mapped basis set as Half Circular Harmonics.

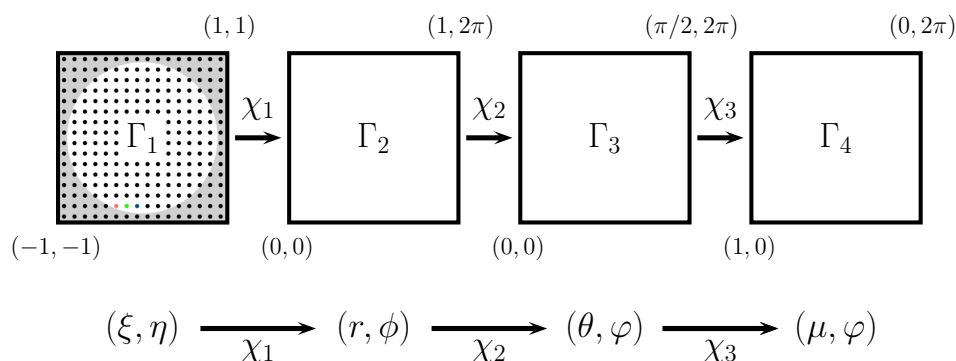


Figure 2. The mappings to transform the coordinates from the original Cartesian set $\Gamma_1 = \{(\xi, \eta) | \xi^2 + \eta^2 < 1\}$ to the target set $\Gamma_3 = \{(\mu, \varphi) | 1 > \mu > 0 \ \& \ 0 < \varphi < 2\pi\}$.

To use the library, the user must provide its own implementation of χ_0 , i.e., the library performs the map $\chi_3 \circ \chi_2 \circ \chi_1$, and all their related transformations. The coordinates of the grid must be normalized, i.e., the spot centers need to have $0 < r < 1$ (which implies $\xi \in [-1, 1]$, $\eta \in [-1, 1]$, and $(\xi^2 + \eta^2) < 1$).

The simple direct and indirect maps χ_i are given by:

$$\chi_0 = \begin{cases} \xi = \frac{2x - x_M - x_m}{x_M - x_m} \\ \eta = \frac{2y - y_M - y_m}{y_M - y_m} \end{cases}, \quad \chi_0^{-1} = \begin{cases} x = \frac{x_M - x_m}{2} \xi + \frac{x_M + x_m}{2} \\ y = \frac{y_M - y_m}{2} \eta + \frac{y_M + y_m}{2} \end{cases}, \quad (12)$$

$$\chi_1 = \begin{cases} r = \sqrt{\xi^2 + \eta^2} \\ \phi = \text{atan}(\eta/\xi) \end{cases}, \quad \chi_1^{-1} = \begin{cases} \xi = r \cos \phi \\ \eta = r \sin \phi \end{cases}, \quad (13)$$

$$\chi_2 = \begin{cases} \theta = \frac{\pi}{2} r \\ \varphi = \phi \end{cases}, \quad \chi_2^{-1} = \begin{cases} r = \frac{2}{\pi} \theta \\ \phi = \varphi \end{cases}, \quad (14)$$

$$\chi_3 = \begin{cases} \mu = \cos \theta \\ \varphi = \phi \end{cases}, \quad \chi_3^{-1} = \begin{cases} \theta = \arccos \mu \\ \phi = \varphi \end{cases}, \quad (15)$$

thus the composite direct and indirect maps $\chi_3 \circ \chi_2 \circ \chi_1$ are

$$\chi_3 \circ \chi_2 \circ \chi_1 = \begin{cases} \mu = \cos \left(\frac{\pi}{2} \sqrt{\xi^2 + \eta^2} \right) \\ \varphi = \text{atan}(\eta/\xi) \end{cases}, \quad (16)$$

$$\chi_1^{-1} \circ \chi_2^{-1} \circ \chi_3^{-1} = \begin{cases} \xi = \frac{2}{\pi} \arccos \mu \cos \varphi = \frac{2}{\pi} \theta \cos \varphi \\ \eta = \frac{2}{\pi} \arccos \mu \sin \varphi = \frac{2}{\pi} \theta \sin \varphi \end{cases}, \quad (17)$$

For our purposes, we will also need the chain rule derivatives relative to the maps χ_0 and $\chi_3 \circ \chi_2 \circ \chi_1$, which are given by:

$$\begin{aligned} \frac{\partial}{\partial \xi} &= \frac{x_M - x_m}{2} \frac{\partial}{\partial x'} \\ \frac{\partial}{\partial \eta} &= \frac{y_M - y_m}{2} \frac{\partial}{\partial y'} \end{aligned} \quad (18)$$

$$\begin{aligned}\frac{\partial}{\partial \xi} &= -\frac{\pi\sqrt{1-\mu^2}}{2}\cos\varphi\frac{\partial}{\partial\mu}-\frac{\pi\sin\varphi}{2\arccos\mu}\frac{\partial}{\partial\varphi}, \\ \frac{\partial}{\partial\eta} &= -\frac{\pi\sqrt{1-\mu^2}}{2}\sin\varphi\frac{\partial}{\partial\mu}+\frac{\pi\cos\varphi}{2\arccos\mu}\frac{\partial}{\partial\varphi}.\end{aligned}\quad (19)$$

Since `OpenWavefrontReconstructor` uses the circular domain Γ_1 , the user must re-scale the gradients using the chain rule (18); here, we assume that the Cartesian coordinates inside Γ_1 are known. In the rest of the section, we describe the `OpenWavefrontReconstructor`'s internal procedure to reconstruct a wavefront (or a surface) defined on the domain Γ_1 .

The sensor should provide (possibly after applying the map χ_0) the gradients of the wavefront of a finite set of points contained in Γ_1 , which must be ordered into an array of the type \mathbf{G} (see Equation (5)). In the library, this ordering is performed by a function whose arguments are the following two array:

$$\mathbf{G}_\xi = \left[\frac{\partial w(\xi_i, \eta_i)}{\partial \xi} \right] = \left[\frac{x_M - x_m}{2} \frac{\partial w(x_i, y_i)}{\partial x} \right], \quad (20)$$

and

$$\mathbf{G}_\eta = \left[\frac{\partial w(\xi_i, \eta_i)}{\partial \eta} \right] = \left[\frac{y_M - y_m}{2} \frac{\partial w(x_i, y_i)}{\partial y} \right]. \quad (21)$$

Here, the index i is the ordering index, and x_m, x_M, y_m , and y_M are the minimum and maximum values of x and y (see Figure 1b).

Internally, `OpenWavefrontReconstructor` assembles the matrix \mathbf{M} (see Equation (8)), applying the chain rules, Equation (19), and using the pairs $(\mu_i(\xi_i, \eta_i), \varphi_i(\xi_i, \eta_i))$. Subsequently, it solves the equation system (10) for the coefficients of the expansion (see Equation (7)). Afterwards, the user can retrieve the coefficients, request the wavefront reconstruction, and retrieve the values of the wavefront at the coordinates (ξ_i, η_i) (with which the matrix \mathbf{M} was originally generated). For the latter, `OpenWavefrontReconstructor` internally applies the inverse map $\chi_1^{-1} \circ \chi_2^{-1} \circ \chi_3^{-1}$ to the coordinates and returns the wavefront, using Equation (1), in the original coordinates (ξ, η) .

After the first reconstruction, the user may request additional reconstructions if the coordinates of the spots do not change (see Section 3.3). This procedure would considerably decrease the computation time if the reconstruction is performed in a loop, which is quite common in adaptive optics, and in high-energy lasers production (especially in the so-called adaptive optics loop, which is a coupled system of a wavefront sensor and a deformable mirror, whose purpose is to correct aberrations originated during the production of the laser beams). After the first reconstruction (which requires assembling the matrix \mathbf{M} , the singular value decomposition shown in Equation (9), and generating the matrix \mathbf{R} defined in Equation (11)), the computational cost of subsequent reconstructions reduces to perform a matrix-vector product (i.e., the product $\mathbf{R}\mathbf{A}$ given in Equation (11)).

Each term of the matrix $(R_{\alpha,i})$ is of the form:

$$R_{\alpha,i} \equiv Y_\alpha(\xi_i, \eta_i) = Y_n^m(\mu_i(\xi_i, \eta_i), \varphi_i(\xi_i, \eta_i)). \quad (22)$$

In Equation (22), n is the principal order of the Spherical Harmonics, and the ordering index $\alpha = \alpha(n, m)$ reproduces the following order: $\alpha(0, 0) = 1, \alpha(1, -1) = 2, \alpha(1, 0) = 3, \alpha(1, 1) = 4, \alpha(2, -2) = 5, \alpha(2, -1) = 6, \dots$ `OpenWavefrontReconstructor` uses this convention. In addition, for the internal numerical computations of `OpenWavefrontReconstructor`, we implemented the Real Spherical Harmonics (or Tesseral Spherical Harmonics), and we use the following normalization convention:

$$Y_n^m(\mu, \varphi) = \begin{cases} \frac{1}{\sqrt{\pi}} P_n^{|m|}(\mu) \sin(|m|\varphi), & m < 0, \\ \frac{1}{\sqrt{2\pi}} P_n^0(\mu), & m = 0, \\ \frac{1}{\sqrt{\pi}} P_n^m(\mu) \cos(m\varphi), & m > 0. \end{cases} \quad (23)$$

Therefore,

$$\int_{-1}^1 d\mu \int_0^{2\pi} Y_n^m(\mu, \varphi) Y_{n'}^{m'}(\mu, \varphi) d\varphi = \delta_{nn'} \delta_{mm'}. \quad (24)$$

5. Classical Zernike Decomposition

In the library, we implemented the classical Zernike polynomials to reconstruct wavefronts defined on circular domains [11]. We provide this in order both to use this reconstruction as a benchmark (i.e., to measure the quality of the reconstruction using Half Circular Harmonics below) and to provide the user with well-known reconstruction methods.

Briefly, the Zernike polynomials are defined as

$$Z_l^m(\rho, \phi) = \begin{cases} R_l^m(\rho) \cos(m\phi), & l \text{ even}, \\ R_l^m(\rho) \sin(m\phi), & l \text{ odd}. \end{cases} \quad (25)$$

Here, l and m are nonnegative integers, $l \geq m$, $0 \leq \rho \leq 1$, $0 \leq \phi \leq 2\pi$, and $R_l^m(\rho)$ is given by

$$R_l^m(\rho) = \sum_{k=0}^{(l-m)/2} \frac{(-1)^k (l-k)!}{k! ((l+m)/2 - k)! ((l-m)/2 - k)!} \rho^{l-2k}. \quad (26)$$

In Equation (26), it has been assumed, by definition, that $l - m$ is even. If $l - m$ is odd, then $R_l^m = 0$, also by definition. For further details, the interested reader may consult, for instance, Ref. [11].

Our implementation uses the Noll's sequential indices [16] for α , which reproduces the following sequence: $\alpha(0,0) = 1$, $\alpha(1,1) = 2$, $\alpha(1,-1) = 3$, $\alpha(2,2) = 4$, $\alpha(2,0) = 5$, $\alpha(2,-2) = 6$, $\alpha(3,3) = 7$, $\alpha(3,1) = 8$, ... We will refer to the radial degree l (see Equations (3), (25) and (26)) as the principal order of the Zernike polynomials.

In version 1.0.0 of OpenWavefrontReconstructor, the user is responsible for renormalizing the focal spot coordinates, such that they are defined on circular domain whose radius is <1 , which can be done using Equations (12) and (18).

Internally, OpenWavefrontReconstructor uses the same procedure as described in Section 4, but exchanging the Half Circular Harmonics by Zernike polynomials of the form shown in Equation (3).

6. Legendre Polynomials for Square and Rectangular Domains

To reconstruct wavefronts defined on square domains, in OpenWavefrontReconstructor, we implemented the classical Legendre polynomials as described by Equation (2).

The user of OpenWavefrontReconstructor must provide a custom implementation of the scaling transformations, using Equations (12) and (18). The rest of the reconstruction procedure is carried out automatically, and there is no required order for the focal spot coordinates (see Figure 1b).

Wavefronts defined on rectangular domains can also be reconstructed using the direct map χ_0 , i.e., Equations (12) and (18), which maps the rectangle into a unit square. The reconstruction would render the square wavefront, and the user would just use the inverse map χ_0^{-1} , i.e., Equation (12), in order to recover the original coordinates. However, since the order in which the coordinates are arranged does not change during the reconstruction algorithms, the user does not need to recompute the original coordinates, but only associate the wavefront values through the index of the arrays (let us recall that the arrays are ordered, see Equations (20) and (21)).

Internally, OpenWavefrontReconstructor uses the same procedure as described in Section 4, but replacing the Half Circular Harmonics by products of Legendre polynomials of the form shown in Equation (2). The ordering index function, $\alpha(k, l)$, for this set reproduces the sequence $\alpha(0, 0) = 1, \dots, \alpha(0, N) = N + 1, \alpha(1, 0) = N + 2, \dots$

7. OpenWavefrontReconstructor Implementation Details

The source code of the herein described wavefront reconstruction algorithms has been made available for public download at [20] under the GPLv2 license. The library OpenWavefrontReconstructor is written in C++ with a high degree of configuration, and it is aimed to facilitate future research in the field, as well as to provide a *ready-to-go* framework for being used in control systems. The library contains the following capabilities:

1. A mock-up generator of various types of incident wavefronts (see Section 8.1 for specific functions).
2. Implementation of different wavefront reconstruction algorithms: Zernike polynomials and Half Circular Harmonics for circular domains, and Legendre polynomials for square domains.
3. Runtime selection of the linear algebra library to perform the algebraic computations
4. Time profiling of the linear algebraic operations (mainly the CPU time (*aka* the process time) needed to generate the matrices M , and R defined in Equations (8) and (11), and also to record the time of the matrix-vector product for performing the operation described in Equation (11)).

All software entities are fully decoupled, therefore OpenWavefrontReconstructor can be easily modified or extended. This also applies to the *class* in charge of performing the algebraic computations. Additionally, porting the linear algebra functions to specific hardware architectures is part of our future work, which is aimed to decrease the CPU times below the millisecond for the operations associated with Equations (10) and (11). According to our experience, this will make the OpenWavefrontReconstructor an interesting option in control systems where the reconstruction is required to be nearly real-time, such as in the adaptive optics [4,14].

The diagram depicted in Figure 3 provides the different options offered by the library, and it shows a typical work flow that can be described as follows:

1. The user can choose the input wavefront as a simulated optical field generated by the mock-up generator, or as a direct input from a sensor. Usually, the sensor vendors provide libraries that can be used to retrieve information such as the focal spots coordinates, when the slopes are measured, and obviously these also provide the slopes. These coordinates and slopes are the inputs received by OpenWavefrontReconstructor, and are used to configure the matrices M and R defined by Equations (8) and (11).
2. Similarly, the user can choose the particular algorithm that will perform the wavefront reconstruction, which can be instantiated as an object from the available classes representing the different approaches (see the previous enumeration list—item 2—for the list of available polynomials).
3. Finally, the user can also select, at runtime, the linear algebra library to perform the linear algebraic operations. In version 1.0.0 of OpenWavefrontReconstructor, only the armadillo library is merged in the code; however, we plan to include more options, and the user can also implement the libraries of his/her preference. OpenWavefrontReconstructor's design is intended to provide an easy-to-follow (nearly copy-paste) environment to implement new linear algebra libraries.

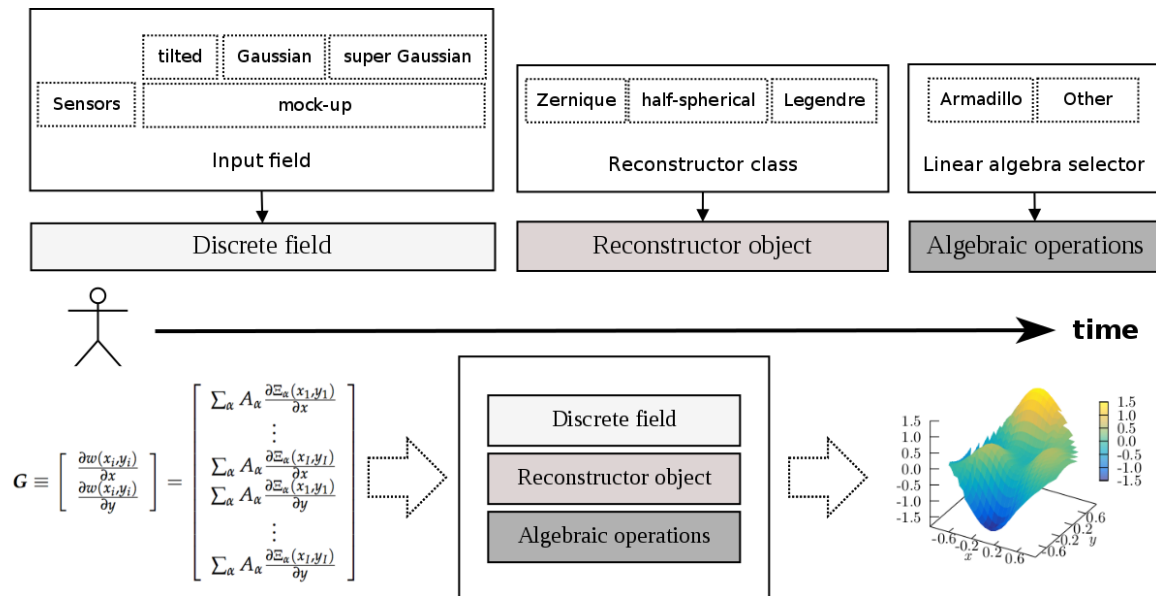


Figure 3. A visual representation of the different possibilities offered by the library for wavefront reconstruction.

8. Numerical Results and Discussion

In this work, we present detailed numerical experiments exclusively for the Half Circular Harmonics, which are proposed by the authors. We compare the accuracy of the reconstruction against the results obtained with classical Zernike polynomials.

8.1. Testing Functions

For the tests and results presented in this section, we use the wavefronts depicted in Figure 4. Most of them were chosen because they are frequently encountered in the laser wavefront reconstruction problem. Figure 4a shows a tilted plane, both around the x - and y -axis. Figure 4b shows a combination of three terms, each of which is a polynomial times an off-centered Gaussian. We will refer to this function as the test f_1 function, and it is defined as follows:

$$f_1(x, y) = \frac{1}{5} \left\{ 3(1 - 2x)^2 \exp(-4x^2 - (2y + 1)^2) - \frac{1}{3} \exp(-(2x + 1)^2 - (2y)^2) \right. \quad (27)$$

$$\left. - 10 \left[\frac{2x}{5} - (2x)^3 - (2y)^5 \right] \exp(-(2x)^2 - (2y)^2) \right\}. \quad (28)$$

Figure 4c shows a centered Gaussian function, i.e., $w(x, y) = A \exp(-a(x^2 + y^2))$, and Figure 4d shows an off-centered Gaussian, i.e., $w(x, y) = A \exp(-a[(x - x_0)^2 + (y - y_0)^2])$. Figure 4e,f show Super Gaussian wavefronts of orders 4 and 6 (a super Gaussian of order n is defined as $w(x, y) = A \exp(-a(x^n + y^n))$, with n even).

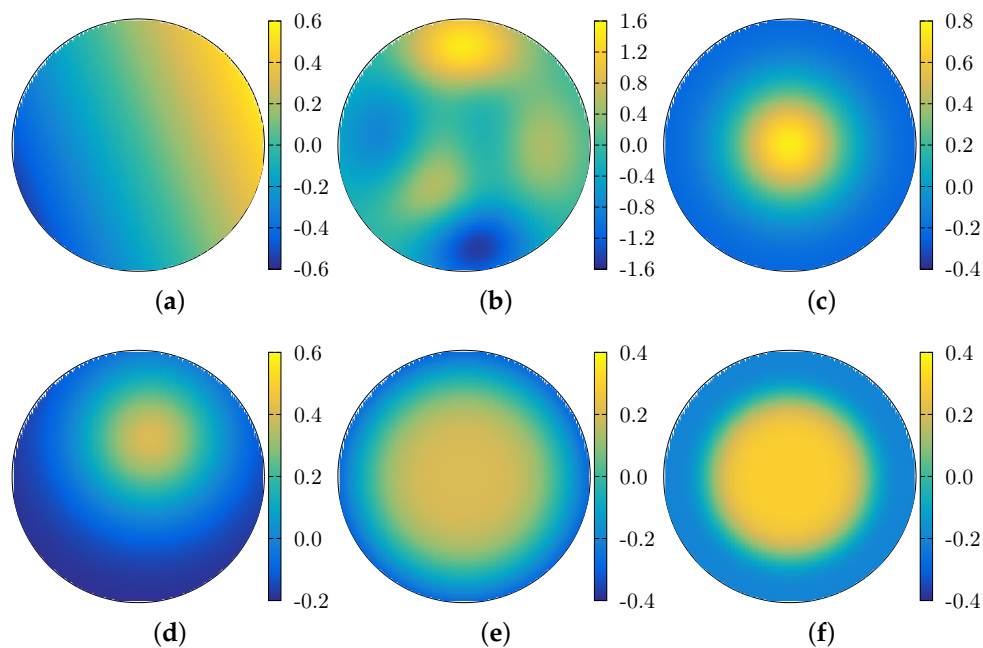


Figure 4. Topview of the wavefronts used to measure the reconstruction algorithm’s accuracy. The functions depicted are as follows. (a) a tilted plane, both around the x - and y -axis; (b) the function f_1 (see Equation (27)); (c) a centered Gaussian; (d) and off-centered Gaussian; (e) a centered super-Gaussian of order 4; and (f) a centered super-Gaussian of order 6.

8.2. Qualitative Reconstruction

In Figure 5, we qualitatively compare the reconstruction of several wavefronts. The grids were generated using a wavefront simulator (i.e., the class `MockWavefrontGenerator` of `OpenWavefrontReconstructor`). The circular grid was obtained from an initial 30×30 square grid, from which only the points inside the unit circle are kept. This grid size corresponds to approximately the number of focal spots measured by a wavefront sensor. Neither the mock generator nor the reconstruction algorithms are exclusive to this size, but the user can setup any grid size.

We used $J = 81$ polynomial terms to perform the reconstruction. This number was obtained by setting the maximum value, N , of the principal order of the Half Circular Harmonic to be $N = 8$. The principal order can be easily identified by comparing it with the corresponding Spherical Harmonic, $Y_n^m(\mu, \phi)$ (see also the text after Equation (22)).

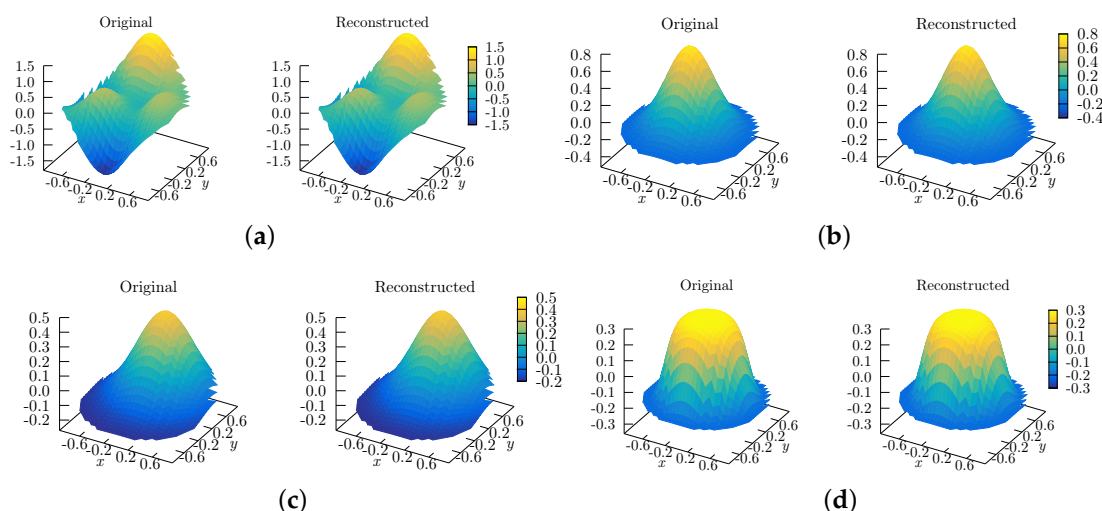


Figure 5. Qualitative reconstruction of several wavefronts, using Half Circular Harmonics, with $J = 81$. The functions depicted are: (a) the test function f_1 (see Equation (27), and Figure 4b for the top view); (b) a centered Gaussian (see Figure 4c); (c) an off-centered Gaussian (see Figure 4d); and (d) a centered super-Gaussian of order 6 (see Figure 4f).

8.3. Accuracy

In Figure 6, we show the accuracy of the reconstruction algorithm using Half Circular Harmonics, via the coefficient of determination, C , which is defined as

$$C \equiv 1 - \frac{S_{\text{res}}}{S_{\text{tot}}} = 1 - \frac{\sum_i (y_i - f_i)^2}{\sum_i (y_i - \bar{y})^2}. \tag{29}$$

Here, S_{res} is known as the sum of squares of residuals, and S_{tot} is the total sum of squares. In addition, in Equation (29), $\{y_i\}$ is the set of known values (i.e., our known test function), \bar{y} the mean value of the set $\{y_i\}$, and $\{f_i\}$ the predicted (i.e., the reconstructed) values (see also [21]). Since the coefficients of determination are close to 1, we actually plot $1-C$ in semi-log scale. As a general trend, a smaller number of polynomial terms, J , is needed to obtain a given accuracy (compared to using Zernike polynomials). The curve for the Tilted Plane using Zernike polynomials, TP*, does not appear because for this case $1 - C$ is close to the machine precision epsilon.

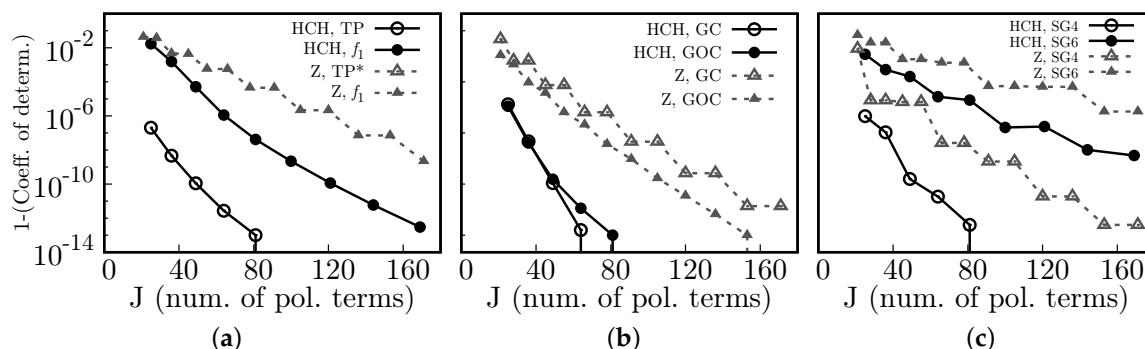


Figure 6. Compared accuracy ($1 - \text{Coefficient of determination}$) vs. J (number of polynomial terms) between Half Circular Harmonics (HCH) and the Zernike (Z) polynomials, for the test functions (a) Tilted Plane (TP/TP*), f_1 , (b) a centered Gaussian (GC), an off-centered Gaussian (GOC), and (c) super-Gaussians of order 4 (SG4) and 6 (SG6).

In Figure 7, we show the R.M.S. for the same test set as in Figure 6. Here, we used the following definition:

$$(\text{R.M.S.})^2 = \frac{\sum_i (y_i - f_i)^2}{\sum_i y_i^2}, \quad (30)$$

where $\{y_i\}$ and $\{f_i\}$ are the set of known and predicted values, respectively.

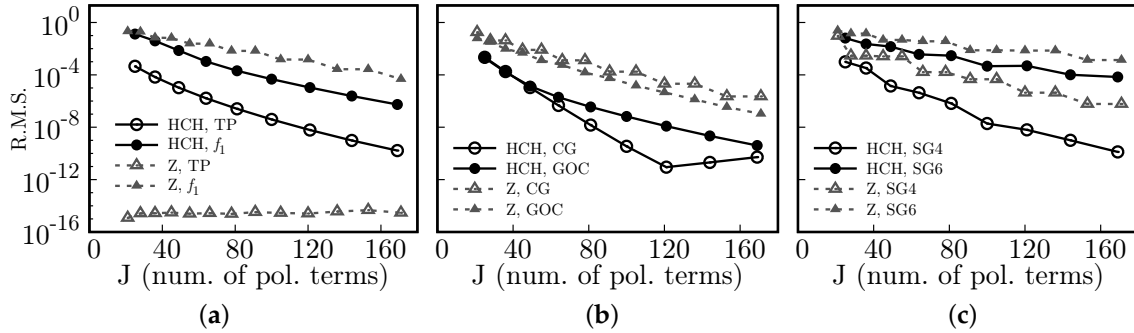


Figure 7. Compared Root Mean Square (R.M.S.) vs. J (number of polynomial terms) between Half Circular Harmonics (HCH) and the Zernike (Z) polynomials, for the test functions (a) Tilted Plane (TP), f_1 , (b) a centered Gaussian (CG), an off-centered Gaussian (GOC), and (c) super-Gaussians of order 4 (SG4) and 6 (SG6).

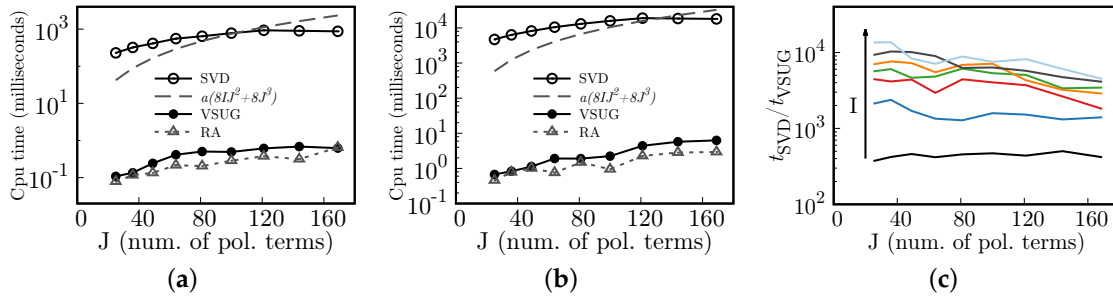


Figure 8. Processing times (CPU times) for different grid sizes (I), number of polynomial terms (J), and matrix operations. SVD is Singular Value Decomposition, VSUG represents the product $VS^{-1}U^TG$ (Equation (10)), and RA the product RA (Equation (11)). (a) matrix operations for a system with $I = 648$; (b) matrix operations for a system with $I = 2724$; (c) the ratio between SVD and VSUG CPU times, $t_{\text{SVD}}/t_{\text{VSUG}}$; the arrow indicates increasing I , and I takes the values $I = 276, 648, 1184, 1876, 2724, 3720, \text{ and } 4872$. In (a,c), $a(8I^2 + 8J^3)$ is the theoretical flops times a scaling constant a .

Except for the tilted plane, both accuracy measures indicate that using HCH increases the quality of the reconstructed surfaces. The tilted plane is perfectly described by Zernike because one of the Zernike polynomials is precisely a tilted plane.

8.4. Performance

Usually, matrix-vector products require $O(2rs)$ floating point operations (here, r is the number of elements of the vector, and the matrix has dimensions $r \times s$ (see [17])). Therefore, reducing either the number of sampled focal spots, I , or the number of polynomials used for the expansion, J , may impact the total CPU time required to reconstruct wavefronts. Furthermore, if the coordinates of the focal spots (i.e., the coordinates where the slopes are sampled) do not change between consecutive reconstructions, then the computational cost should be considerably reduced. This can be seen from the following rationale. Performing singular value decompositions require $O(8I^2 + 8J^3)$

flops, for a matrix of dimensions $2I \times J$ [17]. On the other hand, the products $VS^{-1}U^TG$ (Equation (10)) and RA (Equation (11)) require $O(4IJ)$ and $O(2IJ)$. Hence, after performing the SVD, subsequent reconstructions require $O(6IJ)$ flops.

In Figure 8, we show the CPU times of different matrix operations for different I and J . The Singular Value Decomposition CPU times include the computation of the $M_{\alpha j}$ and $R_{\alpha j}$ terms (see Equations (8) and (11)). As expected, after the first reconstruction, further coefficient estimations (VSUG) and wavefront retrieving (RA) are carried out $\sim 10^2$ to 10^4 faster (see Figure 6c), relative to the complete process (which takes approximately the combined $t_{SVD} + t_{VSUG} + t_{RA} \approx t_{SVD}$ CPU time). The ratio t_{SVD}/t_{RA} increases as the system's size, hence the performance improvement also increases with the system's size.

9. Conclusions

We proposed a new algorithm to reconstruct wavefronts, and more generally surfaces, from gradients using the Spherical Harmonics mapped to work for unit circle domains, which we call Half Circular Harmonics. Relative to the use of classical Zernike polynomials, the same numerical accuracy can be obtained using $\sim 1/2$ to $2/3$ the number of polynomial terms. This might decrease the computational time spent to reconstruct wavefronts from slopes, which may be of interest to designers and manufacturers of wavefront sensors.

Additionally, we implemented the proposed reconstruction algorithm in a library (OpenWavefrontReconstructor) released to the public under the GPLv2 license terms. The library is designed to be used as a part of the wavefront sensors software, hence, in addition to the new polynomials (i.e., the Half Circular Harmonics), we also provide an implementation for reconstructing wavefronts, defined on a unit circle, using the classical Zernike polynomials. On the other hand, for square domains, we implemented the classical Legendre polynomials. Currently, OpenWavefrontReconstructor uses the linear algebra library armadillo to perform the matrix operations; however, its design is intended to provide an easy-to-follow (nearly copy-paste) environment to implement new linear algebra libraries. Further additions to the library are planned as well, such as coupling other linear algebra libraries and additional polynomial sets for square domains. Finally, we also plan to port OpenWavefrontReconstructor to specific massively parallel hardware architectures, in order to decrease the reconstruction computing time.

Acknowledgments: This work was supported in part by the UNED project GID2016-6-1, the Spanish Ministry 317 of Economy and Competitiveness under Projects DPI2014-55932-C2-2-R and ENE2015-64914-C3-2-R and FEDER funds.

Author Contributions: J.M.S.-A., A.V.-O. and D.K. conceived and designed the algorithms, performed the experiments, and analyzed the data. The conceptual idea behind the paper was originated while these authors were working at ELI Beamlines (Dolní Břežany, Czech Republic). All authors contributed to analyzing data, discussing, and writing and reviewing the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Geary, J.M. *Introduction to Wavefront Sensors*; SPIE: Bellingham, WA, USA, 1995; p. 168.
2. Liang, L.; Grimm, B.; Goelz, S.; Bille, J.F. Objective measurement of wave aberrations of the human eye with the use of a Hartmann–Shack wave-front sensor. *J. Opt. Soc. Am. A* **1994**, *11*, 1949–1957.
3. Booth, M. Adaptive optics in microscopy. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* **2007**, *365*, 2829–2843.
4. Rukosuev, A.L.; Kudryashov, A.V.; Lylova, A.N.; Samarkin, V.V.; Sheldakova, Y.V. Adaptive optics system for real-time wavefront correction. *Atmos. Ocean. Opt.* **2015**, *28*, 381–386.
5. Bon, P.; Maucort, G.; Wattellier, B.; Monneret, S. Quadriwave lateral shearing interferometry for quantitative phase microscopy of living cells. *Opt. Express* **2009**, *17*, 13080–13094.
6. Chanteloup, J.C. Multiple-wave lateral shearing interferometry for wave-front sensing. *Appl. Opt.* **2005**, *44*, 1559–1571.

7. Shen, W.; Chang, M.W.; Wan, D.S. Zernike polynomial fitting of lateral shearing interferometry. *Opt. Eng.* **1997**, *36*, 905–914.
8. Bakshi, S.; Yang, Y.H. Shape from shading for non-Lambertian surfaces. In Proceedings of the 1st International Conference on Image Processing, Austin, TX, USA, 13–16 November 1994; Volume 2, pp. 130–134.
9. Geary, J.M. High-energy laser wavefront sensors. *Proc. SPIE* **1991**, *1414*, doi:10.1117/12.43664.
10. Doane, J.F.; Slade, S.G. An introduction to wavefront-guided refractive surgery. *Int. Ophthalmol. Clin.* **2003**, *43*, 101–117.
11. Zernike, v.F. Beugungstheorie des schneidenerfahrens und seiner verbesserten form, der phasenkontrastmethode. *Physica* **1934**, *1*, 689–704. (In German)
12. Trevino, J.P.; Gómez-Correa, J.E.; Iskander, D.R.; Chávez-Cerda, S. Zernike vs. Bessel circular functions in visual optics. *Ophthalmic Physiol. Opt.* **2013**, *33*, 394–402.
13. Mahajan, V.N.; Dai, G.M. Orthonormal polynomials in wavefront analysis: Analytical solution. *J. Opt. Soc. Am. A* **2007**, *24*, 2994–3016.
14. Nirmaier, T.; Pudasaini, G.; Bille, J. Very fast wave-front measurements at the human eye with a custom CMOS-based Hartmann-Shack sensor. *Opt. Express* **2003**, *11*, 2704–2716.
15. Gerard, A.; Mourou, G.K.; Sandner, W.; Collie, J.L. *ELI White Book*; Andreas Thoss: Berlin, Germany, 2011.
16. Noll, R.J. Zernike polynomials and atmospheric turbulence. *J. Opt. Soc. Am.* **1976**, *66*, 207–211.
17. Golub, G.H.; Van Loan, C.F. *Matrix Computations*, 4th ed.; The Johns Hopkins University Press: Baltimore, MA, USA, 2013.
18. Singular Value Decomposition. Available online: <http://mathworld.wolfram.com/SingularValueDecomposition.html> (accessed on 28 November 2017).
19. Sanderson, C.; Curtin, R. Armadillo: A template-based C++ library for linear algebra. *J. Open Source Softw.* **2016**, *1*, 26.
20. OpenWavefrontReconstructor's Github Repository. Available online: <https://github.com/SciCompDEV> (accessed on 28 November 2017).
21. Coefficient of Determination. Available online: https://en.wikipedia.org/wiki/Coefficient_of_determination (accessed on 28 November 2017).

Sample Availability: The source code of the library OpenWavefrontReconstructor can be found in <https://github.com/SciCompDEV>.



© 2017 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).