ASONG Data Analysis Software Manual

Laura Schreiber

First release: 31/08/2022

Contents

Change records

Issue	Date	Section Affected	Comments
1.	31/08/2022	All	First document release
1.1	12/09/2022	All	Second document release
1.2	29/09/2022	Section 8	Added Subsection 8.1.5
		Appendix A/B	Corrected issues referring to Section 8
1.3	30/09/2022	Section 7	Added Section 7.2
		Section 6	Corrected variables order in program calling sequence according to modification in the code
		Appendix A/B	Corrected issues according to updates in Section 6

Acronyms

ASONG	AnalySeur de Front d'Onde de Nouvelle Generation
DAS	Data Analysis Software
GTF	Gradient Transmission Filter
IDL	Interactive Data Language
MLA	Mini Lens Array
PtV	Peak to Valley
RMS	Root Mean Square
SVD	Singular Value Decomposition
WFE	Wavefront Error
WFS	Wavefront Sensor

1 Purpose

The ASONG Data Analysis Software (DAS) is an high level IDL® code for Wavefront slopes computation and Wavefront reconstruction from pupil images. This software is not intended to control any hardware device. However, the code is transparent to the image input source, so it can accept as an input both stored images or images directly coming from the ASONG camera. This option would require to implement an additional coding level that is out of the scope of this software release.

2 ASONG principle and conventions

The ASONG Wavefront Sensor (WFS), also know as crossed-sine WFS, belongs to that class of WFS where an image of the pupil of the tested optical system is formed on a detector array, like the Pyramid WFS, thus enabling high spatial resolution of the measured Wavefront Error (WFE). The basic principle of the crossed-sine WFS, represented in Figure 1, consists in acquiring four pupil images simultaneously, each image being observed from different points located behind a Gradient Transmission Filter (GTF). The GTF is characterised by a transmission function built from a product of two sine functions rotated by 45 degrees around the optical axis.

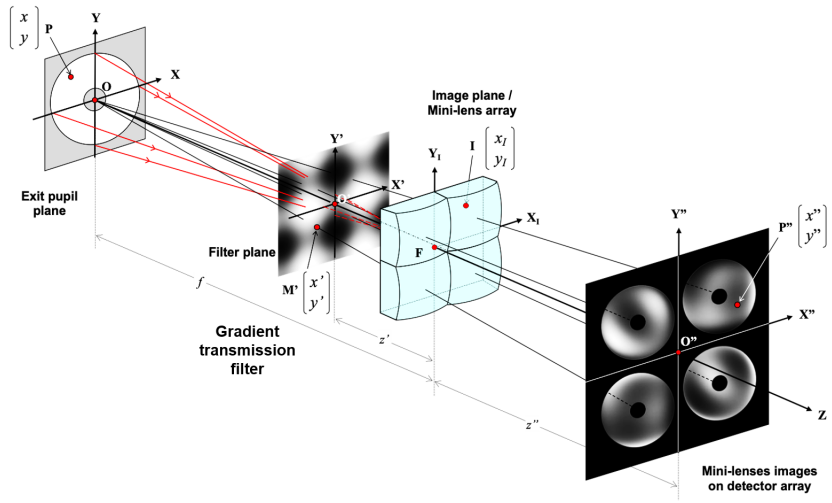

Figure 1: ASONG Principle

The Wavefront derivatives can be retrieved by combining the four pupil images intensity as follows:

$$\frac{\partial \Delta}{\partial x}(x,y) = \frac{1}{g} \arcsin \frac{I_4(x,y) - I_3(x,y) + I_2(x,y) - I_1(x,y)}{\sum_{i=1}^{4} I_i(x,y)} \tag{1}$$

$$\frac{\partial \Delta}{\partial y}(x,y) = \frac{1}{g} \arcsin \frac{I_4(x,y) + I_3(x,y) - I_2(x,y) - I_1(x,y)}{\sum_{i=1}^{4} I_i(x,y)} \tag{2}$$

where g is defined as the WFS gain:

$$g = 2\pi\sqrt{2}\frac{f}{p} \tag{3}$$

f is the distance between the pupil and the focal plane and p is the GTF period. Figure 2 represents an example of ASONG image to be analysed by the present software. The pupil order convention is also reported with 1 to 4 numbering.

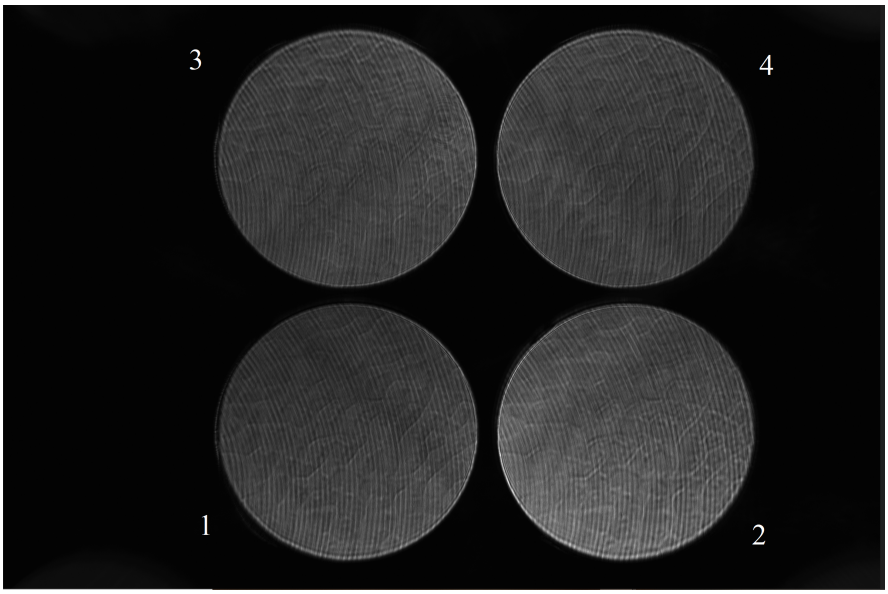

Figure 2: Example of ASONG pupil images and pupil numbering convention

3 Code description

3.1 General Description

The ASONG DAS v.1.0 has no GUI support. The main procedures need to be run by command line. A few operations require an interaction with the user.

3.2 Code Implementation

The basic Wavefront analysis procedure consists in three phases:

- Numerical Pupil Mask definition;
- Wavefront slope computation;
- Wavefront reconstruction from slopes.

The first item can be done once and saved for all the measurements in which the pupil mask does not need to be changed. This step might require an interaction with the user. I recommend to perform the mask computation on images acquired for this purpose, even if this operation can be easily performed directly on ASONG measurement images.

The core of the ASONG program is indeed represented by the Wavefront slopes computation through the algorithm presented in the refereed paper [4] and in the patent [3].

Once the slopes in X and Y are computed by the software, these can be used to reconstruct the Wavefront. The present software offers two different reconstruction methods (see Section 8), but other methods can be implemented if required.

The routines are organised in the folder as shown in Figure 3.

The three main procedures required to perform the Wavefront computation and analysis are in the main folder (`asong_mask_main.pro`, `asong_slope.pro`, `asong_wfe_reconstruction.pro`). The libraries are contained in three packages (`mask_package`, `slope_computation_package`, `wfe_reconstruction_package`) while a folder named 'TEST' contains an example of image and a few files for testing and training. Each procedure contains an header containing the description of the inputs, outputs, keywords and call sequence example. The headers of the main routines are contained in this document

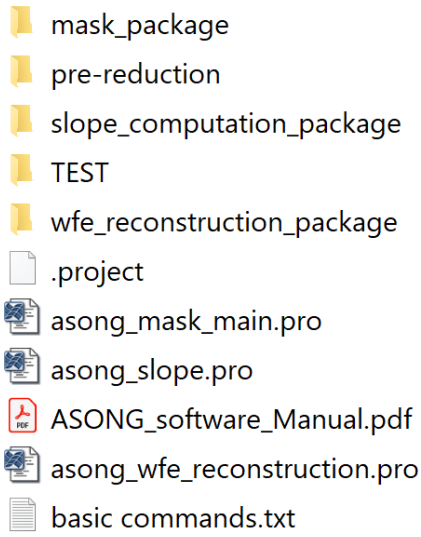

Figure 3: Code organisation in the software folder.

Appendix. A few procedures for averaging, flat-fielding and back-ground subtracting are contained in the folder called 'pre-reduction'. These procedures are not called by the software and they are not part of it. The user is free to use them or take inspiration from them. A sequence of quick basic commands to run the software are contained in the txt file called "basic commands.txt" The file "ASONG_software_MANUAL.pdf" contains this manual.

4 Setup

4.1 Requirements

Since the software is written in IDL® language, it requires IDL® properly installed with a valid licence. The software has been tested using IDL 8.8.0.

This software works under Windows OS, but it can be adapted to other OS with minor changes.

4.2 Installation

The ASONG DAS needs no particular installation procedure. After downloading the code as a compressed archive, decompress it and put its content in a new directory. The second (and last) step is to tell IDL® where the new directory containing the software is located modifying the IDL® system variable !Path. This can be done very easily within the IDL® Development Environment. Since the compressed folder contains a few sub-folders, do not forget to check the box beside the ASONG software path (see Figure 4) Oth-

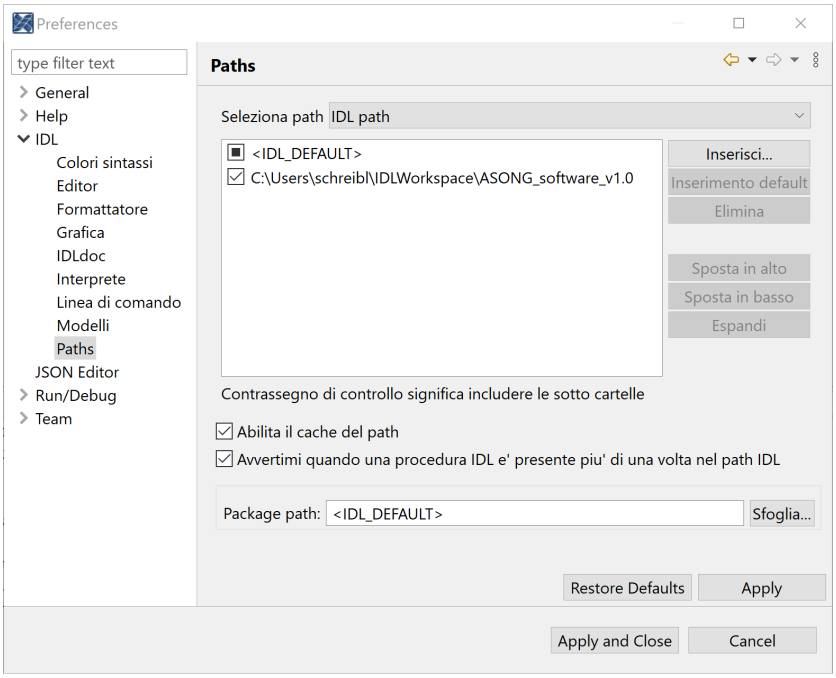

Figure 4: How to set the correct IDL® path for proper installation.

erwise the path can be modified from the IDL® command line: the way to do it is described in the IDL® Online Help, under the topic !Path. Once the IDL® session is open and the path is set, the procedures will be automatically compiled when called by the main procedures. Each time a new version of the code is released and download, delete or remove the old version from the IDL® path. Reboot the IDL® if necessary.

The code uses routines included in different public and private libraries. For simplicity and to assure the stability of the program, only the routines

used by the ASONG Data Analysis Software of the different libraries have been copied in the ASONG library.

In the following, a list of libraries used in the code:

- standard IDL® libraries;
- some modules of the IDL® Astronomy library
- MPFIT package
- Starfinder package[2, 10]

Please, check for disclaimers / copyrights before re-using pieces of the code.

5 Input Image variable

The software works with a 2D array stored in an IDL® variable. This variable needs to be defined and the image needs to be copied into it. Depending on the extension of the saved image, different IDL® functions/procedures are available in the IDL® standard libraries. A short list of examples is reported in the following:

- `image = READ_TIFF(''ASONG image.tif'')` ; IDL® function to read the .tif extension. The string must contain the image location. The image is compied in the `image` IDL® variable.
- `READ_JPEG, ''ASONG image.jpg'', image` ; IDL® procedure to read the .jpg extension. The string must contain the image location. The image is compied in the `image` IDL® variable.
- `RESTORE,''asong image.sav''` ; IDL® procedure to restore the ASONG image once saved in IDL® extension .sav. The string must contain the image location. The variable containing the image becomes visible in the 'Variables' list with its name. To visualise it on the command line, the command to type is `help`.

5.1 Image pre-reduction

The ASONG DAS does not include image pre-reduction operation (i.e. flat fielding, back ground subtraction, Rolling shutter effect correction). A few useful procedures can be found in the relative folders. Please refer to the procedures header for help.

6 Pupil mask definition

In order to compute the Wavefront derivatives assuring the correct combination of corresponding pixels belonging to different pupil images, the four pupils region of interest needs to be precisely defined. This mask should be re-defined every time some change that could affect the pupil position and size occurs in the setup. This mask will be used for the successive steps. Together with the four pupils mask to be applied to the ASONG output image, the procedure gives in output the pupil mask `pup_mask` for the slopes computation and for the final wavefront computation and a 2D array (2 lines $\times$ 4 columns) containing the four pupil centres. These coordinates can be used for MLA alignment verification.

6.1 How to create a pupil mask

The procedure to call in the IDL® command line to create the masks is called `asong_mask_main`. This procedure calls the procedures / functions

`asong_define4quad.pro` for quadrants definition (Sect. 6.1.2);
`asong_pupil_edge_fit.pro` for pupils edge fitting (Sect. 6.1.3);
`asong_mask.pro` for mask creation (Sect. 6.1.4);
`asong_check_mask.pro` for quick checking (Sect. 6.1.5)

Most of the times, this procedure requires the user to click on IDL® windows. Check the IDL® console for interactive instructions.

6.1.1 Calling Sequence

The basic calling sequence is the following:

IDL > `asong_mask_main,image, centres, quad_mask, pup_mask`

where `image` is a 2D array containing the pre-reduced image of the four pupils and `centres, quad_mask, pup_mask` are the outputs necessary for the successive steps. For a more precise definition of the pupil mask, it is recommended, when possible, to acquire an image with uniform pupil illumination. The most precise way is to mount a mask in the pupil plane with a pinhole producing an image on the detector of about 1 pixel. A dedicate procedure can be used in this case (see Section 6.1.6). See previous Section 5 for help on input variable image.

6.1.2 Four quadrants definition

The actual version of the ASONG DAS is not able to detect the four pupil images automatically to define the four quadrants and it requires an interaction with the user. We recall that in multiple measurements, the mask does not need to be updated. Once called the 'asong_mask_main' procedure, the user is asked to individuate the four quadrant by clicking approximately in the center of the four pupil images. In the IDL® console will appear the message:

Figure 5: The IDL® window displays the image and the user is asked to click approximately at the center of the four pupils

```
 ‘‘Select the center of the mask by clicking with the left
button of your mouse...  Push right button to exit’’
```

and an IDL® window titled “quad definition” will automatically open (Figure 5).

When the user clicks using the left button of the mouse, a cross centered in that point appears in the window as shown in Figure 6.The cross should not touch the four pupil images. The user can click until the cross is in the good position. Clicking the right button of the mouse, the position of the last click is used for further calculations.

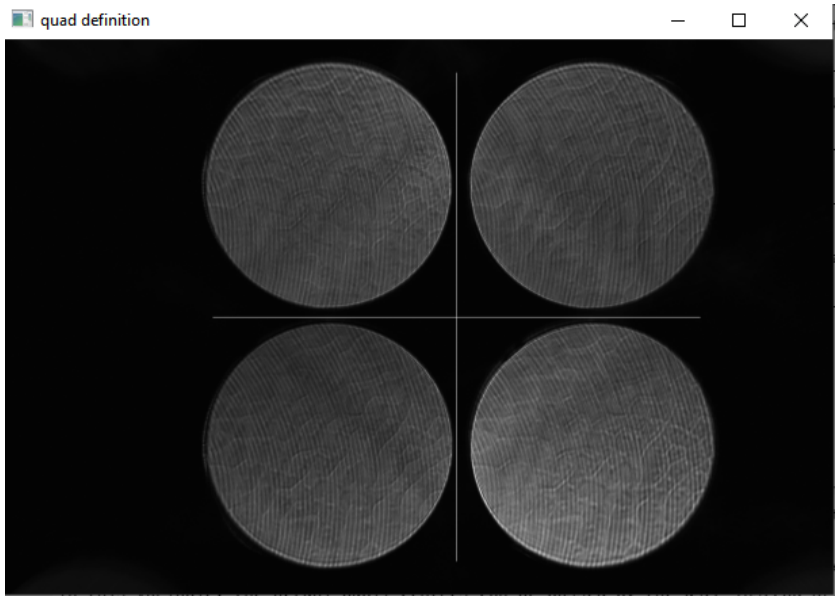

Figure 6: Each time the user clicks on the image, a cross appears in the window

6.1.3 Pupil edge fitting

The pupil images edge detection can be done manually by clicking on the images (DEFAULT option) or automatically by IDL® SOBEL function adding the keyword `/AUTO` to the call. The program fits circular curves to the pupil edges points through Levenberg-Marquardt least-squares minimization (see

`mpfit` NASA libraries and [5, 6]). The computed four pupil centres and radii are displayed in the IDL® Console in pixel units and used by the main program to compute the mask.

6.1.3.1 Pupil edge fitting by clicking on the image

As soon as the four quadrants are defined, the message

```
 ‘‘Select at least three points of the mask edge by clicking
with the left button of your mouse...
Push right button to exit’’
```

will be displayed in the IDL® console. One by one the four pupils will be displayed in different IDL® windows. The user is required to click on the edge of the pupils using the left button of the mouse (Figure 7). The program will use these points to fit the edges of the four pupils to circles. When the user is satisfied with the number of points to define the pupil edge, by clicking the right button of the mouse can continue with the following pupil image until the end. The minimum of required points is three, but to have a more precise mask, 10-15 clicks should be preferable.

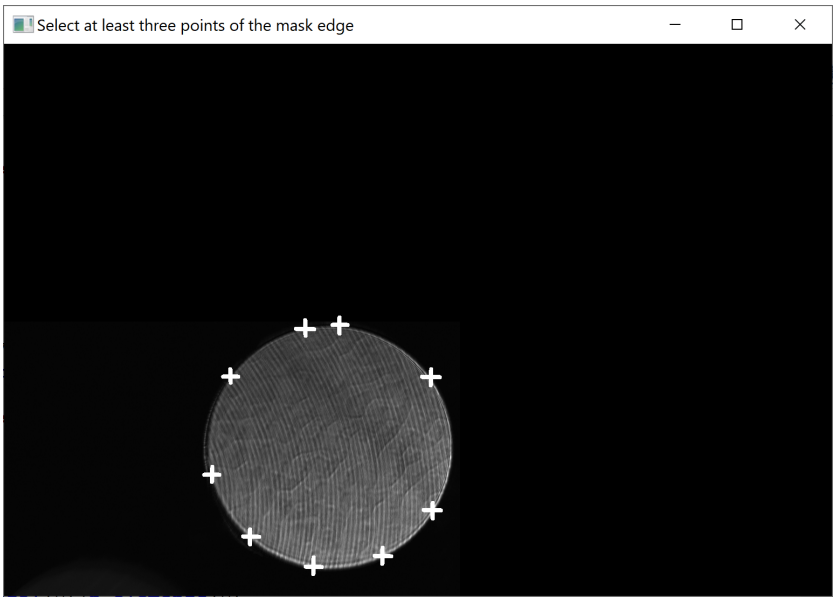

Figure 7: Example of pupil 1 edge fitting procedure by user clicking

6.1.3.2 Pupil edge fitting through automatic edges detection

IDL > `asong_mask_main,image, centres, quad_mask, pup_mask, /AUTO, THRESHOLD = 0.15`

When the pupil images are well illuminated and defined, the keyword `/AUTO` can be used. When this KEYWORD is set in the `asong_mask_main` call, the IDL® `SOBEL` function for pupil edge detection is used. In this case, no interaction with the user is required to detect the pupil edges and windows displaying the automatically detected edges will appear for visual check (Figures 8 and 9). When using the automatic procedure, possible errors

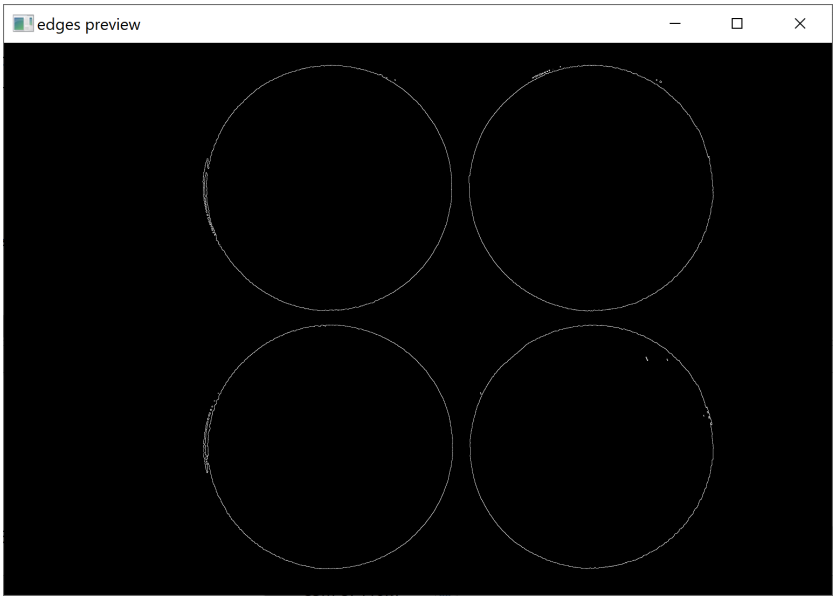

Figure 8: Pupil edges preview when using `SOBEL`

can occur due to image light distribution. Using the keyword `THRESHOLD = THRESHOLD` a different normalised threshold value for the edge detection can be used. The default value is 0.1. A typical example is represented in Figure 10 where regions external to the pupil images has an intensity value grater than the default threshold value. By setting an higher threshold (i.e. `THRESHOLD = 0.15`) a better result can be obtained.

Figure 9: Pupil edges preview over-plotted to pupil images

Figure 10: Example of possible edge detection error when using `SOBEL`

6.1.4 Mask creation

Once computed the centres of the four pupils through edge fitting, the program builds the binary mask for the signal computation and the pupil mask. The four pupil images mask is a 3D array composed by four layers containing each the binary mask relative to one pupil image. They are ordered as the convention introduced in Figure 2. The default pupil radius is `RAD = 450` pixels. This size can be changed by setting the keyword `RAD = RAD` to a different value.

```
IDL > asong_mask_main,image, centres, quad_mask, pup_mask, RAD =
480
```

6.1.5 Mask checking

Once the masks are created, a visual checking is performed automatically. This procedure allows the user to quick check the quality of the result and also to possibly refine some mask additional input parameter, like the radius and the pupil centers coordinates. Three different kinds of visual checks are displayed in IDL® windows:

- masked input image visualisation (Figure 11 - Top): to check for correct edge masking and proper radius size;
- complementary of the masked input image visualisation (Figure 11 - Middle): to check for correct pupil centering and proper radius size;
- visualisation of a mask created with the fitted radius values (Figure 11 - Bottom): sanity check of fit circumferences parameter.

Figures 11 and 12 show an example of successful and unsuccessful pupil mask definitions.

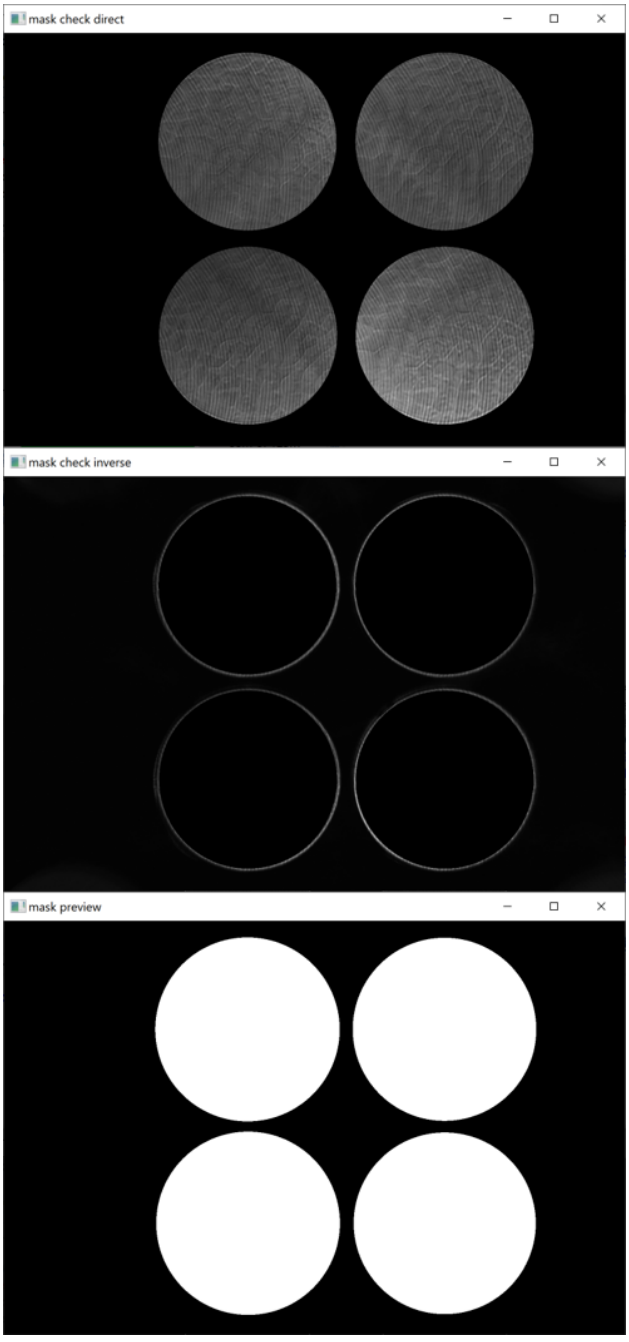

Figure 11: Pupil mask visual checking windows

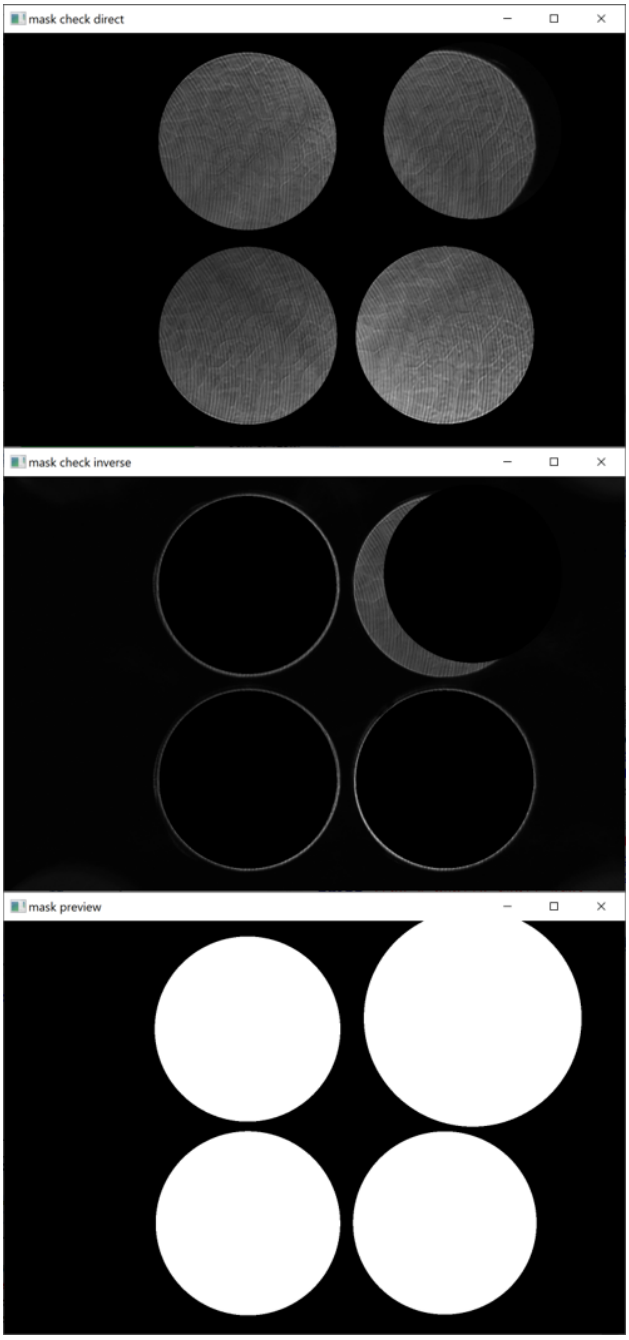

Figure 12: Mask error detection thanks to pupil mask visualisation

6.1.6 Mask definition using a pinhole

In order to make precise centering of the four pupil mask, the use of a pinhole is preferred. In the `asong_mask_main` procedure, this option is not yet implemented, but the `FUNCTION` called `asong_center_pupil` is already available in the folder `mask_package`. When the keyword `/CLICK` is set, this function computes the centroid of the pupil mask pinhole images by clicking on them through center of gravity algorithm or 2D gaussian fit (if the `/GAUSS` keyword is set).

```
IDL > centres = asong_center_pupil(pinhole_image, /CLICK, GAUSS =
GAUSS
```

This Function returns the 2D array of pupil centres in the user clicking order. Please, refer strictly to the convention shown in Figure 2. It is then possible to create the masks by calling the `asong_mask` procedure as follows:

```
IDL > asong_mask,centres, quad_mask, RAD = RAD
```

Once the variables centres, quad_mask and pup_mask are available, the pupil radius can be changed as explained in Section 6.2

6.2 How to change the pupil radius

It is possible to change the pupil radius (see Figure 13) without performing the pupil edge fit and the quadrants definition just recalling the main program, passing the centres coordinate using the variable `centres` as an input and setting the keywords `/CEN` and `RAD = RAD`

```
IDL > asong_mask_main,image, centres, quad_mask, pup_mask, RAD =
400, /CEN
```

When the keyword `CEN = CEN` is set, the `centres` variable must be defined. If not, an error message appears and the procedure returns. This option might be useful also in case of low quality pupil edges detection. Through this call it is also possible to force the single pupils shifting by acting on the centres array containing the coordinates of the centres.

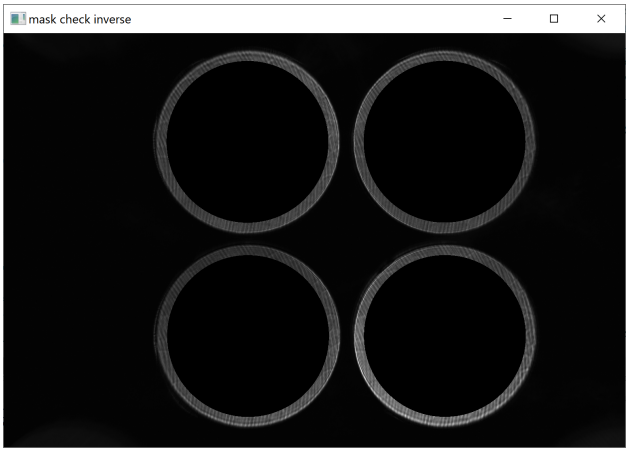

Figure 13: Pupil inverse mask in case of smaller pupil

6.3 How to save a pupil mask

The outputs of the `asong_mask_main` procedure can be saved automatically if the keyword `/SAVE_FILE` is set in the call. When setting the keyword `/SAVE_FILE`, the directory and the file name can be also set through the keywords `DIR = DIR` and `NAME = NAME`. The two output files "ASONG_maskNAME.sav" and "ASONG_maskNAME_preview.bmp" contain respectively the variables `quad_mask, pup_mask, centres, rad, dimpup` and a preview of the four pupils mask.

```
IDL > asong_mask_main,image, centres, quad_mask, pup_mask, /SAVE_FILE,
DIR = ''my dir'', NAME = ''my namefile''
```

If the keywords `DIR = DIR` and `NAME = NAME` are not set, the outputs files named "ASONG_mask.sav" and "ASONG_mask_preview.bmp" are saved in the current IDL®.

6.4 How to restore a pupil mask

When the mask and the other outputs of the `asong_mask_main` procedure are saved in the IDL® .sav format, they can be used whenever needed simply restoring the file using the IDL® `RESTORE` procedure.

IDL > `restore,''my dir'' + ''my namefile'' + ''.sav''`

6.5 Display options

A few keywords are available in the `asong_mask_main` procedure to set the window size and to operate on the displayed image contrast.

6.5.1 How to set the IDL® window size

The keyword `WSCALE = WSCALE` can be used to adapt the window size to the actual monitor resolution. The default value is 0.5. Setting, for instance, `WSIZE = 0.3`, the displayed windows will be smaller, while setting `WSIZE = 0.7` the displayed windows will be bigger.

6.5.2 How to change the image contrast

A very basic option is available to change the image display scale form linear to exponential by using the keyword `LOG = LOG`. The value passed in the `LOG` keyword is the value of the exponent.

6.6 Algorithms description

The main algorithms used in the code for the mask definition are listed below, including references used for code implementation:

- SOBEL algorithm for pupil edge enhancement. Mathematical expression at this link: `https://www.l3harrisgeospatial.com/docs/sobel.html`
- Levenberg-Marquardt least-squares minimization for pupil circular fitting. References can be found in the `mpfit` NASA libraries and in [5, 6]
- Center of gravity computation for pinhole image centroid computation. The mathematical description of the algorithm can be found in [11]

7 Wavefront Slopes computation

The computation of the X and Y slopes represents the core of the ASONG DAS. Given the pre-reduced ASONG WFS image (Section 5) properly masked (6), the code returns the X and Y wavefront derivatives defined over the pupil mask. No interaction with the user is required. The returned slopes are multiplied by the detector pixel size (or sub-aperture size) projected in the pupil plane for the radians to microns transformation. This multiplication is done in the slopes computation step even if formally speaking the slopes should return in radians unit. This arbitrary choice is a simple convention. The DEFAULT value of this parameter is set using theoretical magnification factors, but a calibration in the lab is strongly recommended. See Section 7.1.3 for more details. In this piece of code, all the physical parameters of the prototype, like the GTF periods in mm (`DXF = DXF`, `DYF = DYF`), the distance between the pupil of the tested optical system and the focal plane in mm (`FOC = FOC`), the camera pixel size in microns (`DET_PX = DET_PX`) and the system magnification (`PUP_SCALE = PUP_SCALE`) are set and they can be changed by using the dedicated keywords.

7.1 How to compute XY Wavefront Slopes

The procedure to call in the IDL® command line to compute the X and Y slopes is called `asong_slope`. This procedure does not call other procedures / functions, a part from those of the standard IDL® libraries. This procedure implements the reconstruction formula reported in [4] and recalled in Section 1.

7.1.1 Calling Sequence

The basic calling sequence is the following:

```
IDL > asong_slope,image, quad_mask, pup_mask, centres, slope_x,
slope_y
```

where `image` is a 2D array containing the pre-reduced image of the four pupils (see Section 5), `quad_mask`, `pup_mask`, `centres` are respectively a 3D array containing the four pupil masks in the correct order, as assumed in the convention represented in Figure 2, the pupil mask for slopes computation

and the 2D array containing the coordinate centres of the pupil images. These variables are all returned by the `asong_mask_main` procedure in the format expected by the `asong_slope` procedure.

7.1.2 Slopes computation

The X and Y slopes are computed combining the intensities of corresponding pixels of the four pupil images as described in Section 2. Both the `GAIN` and the `PUP_SCALE` are computed from theoretical values, but measured values can be passed to the procedure in the form of keywords. Once the slopes are computed, if the keyword `/DISPLAY` is set, an IDL® window opens automatically and displays the result (see Figure 14).

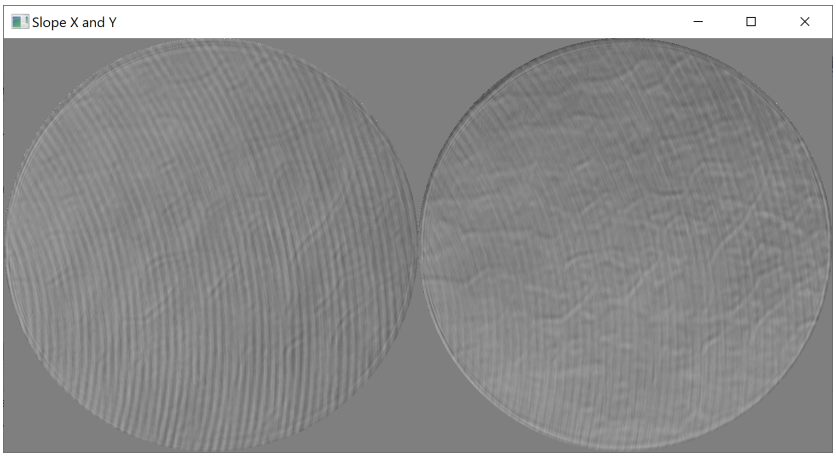

Figure 14: Computed X and Y slopes

7.1.3 Pupil scale factor

The DEFAULT value of the `PUP_SCALE` variable is computed by using nominal focal lengths, but the true value can be measured in the laboratory by putting in the pupil plane an object of known size (i.e. a transparent ruler) and measuring its size on the ASONG detector plane. The IDL® that perform the measurement is not included in the ASONG DAS (for the moment).

7.2 Error messages

In the process of computing the slopes from the four pupils intensities combination, infinite values can might occur. In the slope computation the arcsinus function is computed[4]. Whenever a value of the argument is found to be outside the interval $[-1 : +1]$, this value is automatically substituted to the `signum` function and an error message appear:

```
   WARNING! Detected n infinite values in the slope computation'
Automathic sobstitution of infinite values...'
```

7.3 How to save XY Wavefront Slopes

The outputs of the `asong_slope` procedure can be saved automatically if the keyword `/SAVE_FILE` is set in the call. When setting the keyword `/SAVE_FILE`, the directory and the file name can be also set through the keywords `DIR = DIR` and `NAME = NAME`. The two output files "ASONG_slopeNAME.sav" and "ASONG_slopeNAME_preview.bmp" contain respectively the variables `slope_x, slope_y, pup_mask` and a preview of the slopes X and Y.

```
IDL > asong_slope,image, quad_mask, pup_mask, centres, slope_x,
slope_y, /SAVE_FILE, DIR = ''my dir'', NAME = ''my namefile''
```

If the keywords `DIR = DIR` and `NAME = NAME` are not set, the outputs files named "ASONG_slopes.sav" and "ASONG_slopes_preview.bmp" are saved in the current IDL®.

7.4 How to restore XY Wavefront Slopes

When the slopes computed by the `asong_slope` procedure are saved in the IDL® .sav format, they can be used whenever needed simply restoring the file using the IDL® `RESTORE` procedure.

```
IDL > restore,''my dir'' + ''my namefile'' + ''.sav''
```

7.5 Display options

A few keywords are available in the `asong_slope` procedure to set the window size and to operate on the displayed image contrast.

7.5.1 How to set the IDL® window size

The keyword `WSCALE = WSCALE` can be used to adapt the window size to the actual monitor resolution. The default value is 0.5. Setting, for instance, `WSIZE = 0.3`, the displayed windows will be smaller, while setting `WSIZE = 0.7` the displayed windows will be bigger.

7.5.2 How to change the image contrast

A very basic option is available to change the image display scale form linear to exponential by using the keyword `LOG = LOG`. The value passed in the `LOG` keyword is the value of the exponent.

7.6 Algorithms description

The main algorithm implemented in the `asong_slope` procedure is the one presented in [4] and recalled in Section 1.

8 Wavefront reconstruction

After the computation of the X an Y slopes, it is possible to reconstruct the Wavefront calling the procedure `asong_wfe_reconstruction`. This piece of code represents the most complex of the ASONG DAS and it could take some computation time and could require quite a lot of memory. In the actual version of the ASONG DAS, this procedure and sub-procedures have not been optimised neither accelerated. Given the slopes `sx` and `sy` of the Wavefront and the pupil mask `pup_mask`, the code returns the map of the reconstructed Wavefront through iterative reconstruction (see Section 8.4 for algorithm references and description) and the map of the first Zernike modes (20 by default), with Zernike coefficients values (see Section 8.4). The results are returned in microns units unless the keyword `/WAVES` is set. The number of reconstructed Zernike modes can be set using the keyword `N_ZER = N_ZER`; the DEFAULT value is 20. The computation of the modal

reconstruction matrix can be particularly time consuming and this matrix could also occupy quite a lot of memory, if the number of settled modes is high. The reconstruction matrix is in fact a 2D matrix with size [2 * number of sub-apertures in the pupil × number of modes]. The number of sub-apertures corresponds to the number of pixels in which the pupil is defined. If the pupil image diameter D is around 1000 pixels, the number of sub-apertures is around $\pi * D^2/2 \approx 1.6 \times 10^6$. For this reason, the keyword `/NOZER` can be set. In this case, the modal reconstruction will be skipped and the slopes will be processed only by the iterative method. A calibration coefficient can be passed through the keyword `CALIBRATION = CALIBRATION` in case the slopes `PUP_SCALE` needs further correction.

8.1 How to compute the Wavefront from Slopes

The procedure to call in the command line is `asong_wfe_reconstruction`. This procedure calls numerous procedures / functions included in the library called `wfe_reconstruction_package`. This procedure does not require interaction with the user.

8.1.1 Calling sequence

The basic calling sequence is the following:

IDL > `asong_wfe_reconstruction,slope_x, slope_x, pup_mask, wfe_it, wfe_zer, zer_coeff`

where `slope_x`, `slope_x` are 2D arrays containing the wavefront slopes computed in the `pup_mask` mask, `wfe_it` and `wfe_zer` are the 2D arrays containing the WFE map reconstructed using respectively the Fourier iterative method (Section 8.1.2) and the modal approach (Section 8.1.3).

8.1.2 Wavefront Error reconstruction

By default, the WFE is reconstructed using the Fourier iterative reconstruction method. This method has been formalised by Roddier & Roddier in 1991 and is described in Section 8.4 and in reference [9]. This method allows us to have high order modes reconstructed without the need of a big modal

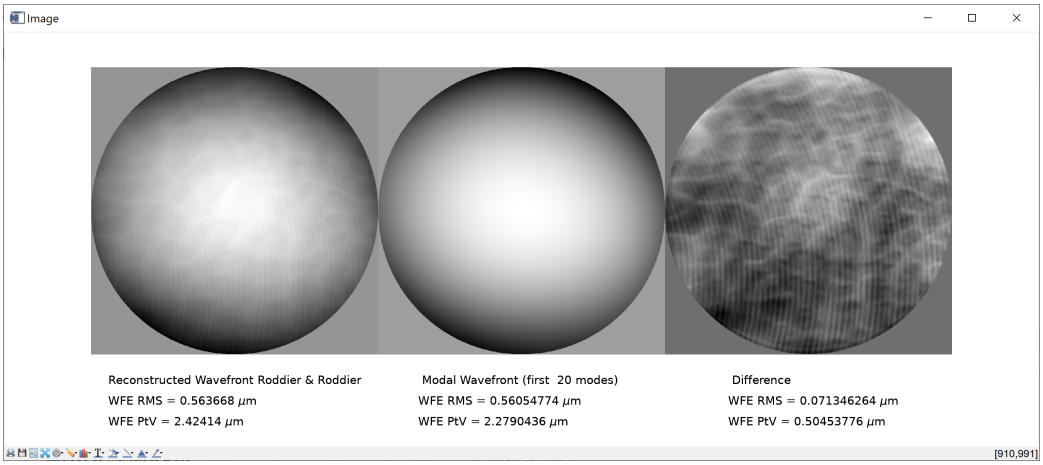

Figure 15: WFE display output in microns

reconstruction matrix computation. However, no information on spacial frequency distribution of the WFE is available. For this reason, if nothing is specified, the procedure computes also a small reconstruction matrix and the 20 first Zernike modes are returned. Using the keyword `N_ZER = N_ZER` it is possible to set a different number of Zernike modes. The modal reconstruction can be also excluded by setting the keyword `/NOZER`. Once launched the `asong_wfe_reconstruction` procedure, the PC where the ASONG DAS is running will take some times to perform the computation. Both the Fourier iterative method and the SVD for interaction matrix inversion are time consuming (1 minute for standard parameters on a Intel(R) Core(TM) i7-7Y75 CPU @ 1.30GHz 1.60 GHz). If the keyword `DISPLAY = DISPLAY` is not set to `0` (DEFAULT value = 1), two IDL® graphical windows will automatically appear reporting the representation of the WFE maps reconstructed using the two methods and the difference of the two and the plot of the first N = 20 (or more) Zernike coefficients. The RMS, PtV and Zernike coefficients values are in microns (or in waves if the keyword `/WAVES` is set, see Figure 15).

By default, the TIP-TILT order is removed by subtracting the average value from the WFE X and Y slopes (Figure 16). This can be avoided setting the keyword `NOTILT = 0` (DEFAULT value = 1, (Figure 17)). The values of the WFE RMS and PtV, as well as the value of the first N = 20 (or N_ZER) Zernike modes are printed in the IDL® console (an example is shown in Figure 18)

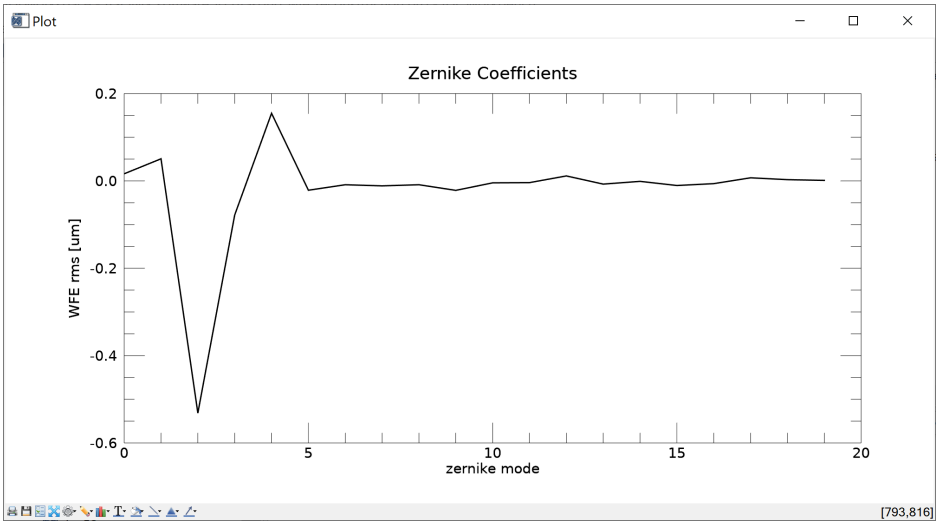

Figure 16: Plot of the first N = 20 Zernike modes (`NOTILT = 1`)

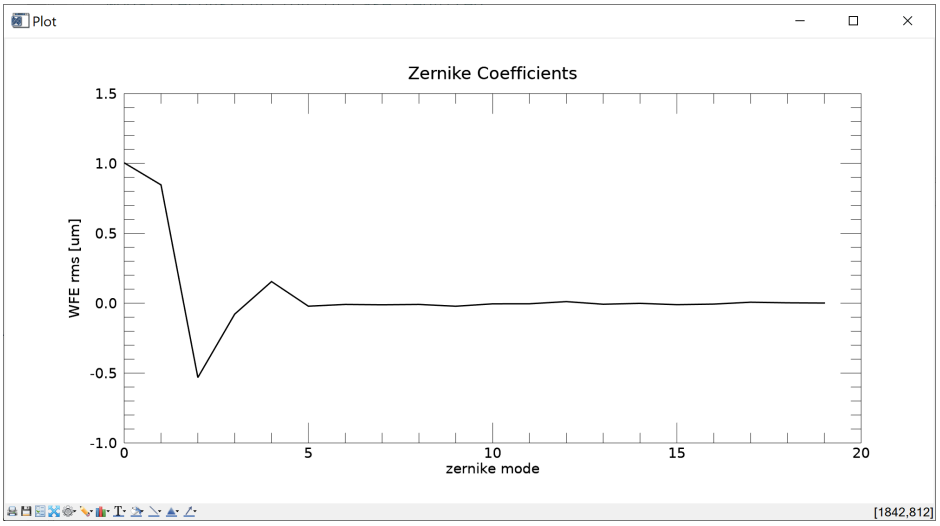

Figure 17: Plot of the first N = 20 Zernike modes (`NOTILT = 0`)

```
number of zernike modes computed:       20
WFE RMS = 1.42686 microns
WFE PtV = 5.81262 microns
zernike coefficient  0 =  1.0045211
zernike coefficient  1 =  0.84665792
zernike coefficient  2 =  -0.53162270
zernike coefficient  3 =  -0.078046328
zernike coefficient  4 =  0.15479426
zernike coefficient  5 =  -0.021346145
zernike coefficient  6 =  -0.0087702072
zernike coefficient  7 =  -0.011325993
zernike coefficient  8 =  -0.0087526566
zernike coefficient  9 =  -0.021654668
zernike coefficient  10 =  -0.0044362391
zernike coefficient  11 =  -0.0039441263
zernike coefficient  12 =  0.011386235
zernike coefficient  13 =  -0.0073260804
zernike coefficient  14 =  -0.0010388564
zernike coefficient  15 =  -0.010506743
zernike coefficient  16 =  -0.0062658663
zernike coefficient  17 =  0.0071096062
zernike coefficient  18 =  0.0028807373
zernike coefficient  19 =  0.0011837097
IDL>
```

Figure 18: The WFE RMS and PtV and the Zernike Coefficients are also printed in the IDL® console

8.1.3 Using a pre-computed reconstruction matrix

The reconstruction matrix must be computed every time that the pupil mask or the number of Zernike modes to compute change. If none of these two variable changes, a pre-computed reconstruction matrix can be used. The reconstruction matrix can be passed to the `asong_wfe_reconstruction` call by using the keyword `RMAT = RMAT`. Together with this keyword, the matrix of Zernike polynomials defined on the proper pupil mask must be passed using the `zermat` optional output variable. The typical call is:

```
IDL > asong_wfe_reconstruction,slope_x, slope_x, pup_mask, wfe_it,
wfe_zer, zer_coeff, zermat, RMAT = RMAT
```

This call can be used even if the `RMAT` is not defined. In this case, it will be computed by the procedure and returned as an output in the variable `RMAT`. WARNING!!! If called a second time with the same syntax and without re-setting the IDL® session, the `RMAT` variable will be passed to the procedure as an optional input and the matrix will not be re-computed. In case the reconstruction matrix stored in the RMAT IDL® variable has a size not compatible with the working pupil size or with the number of required Zernike modes, the reconstruction matrix is re-computed and a message appear in the IDL® console:

```
‘‘Incompatible reconstruction matrix encountered.
Re-computing RMAT...’’
```

A similar message appears if the Zernike cube is not passed to the procedure together with the reconstruction matrix. Another possibility is represented by restoring the reconstruction matrix and the Zernike polynomials array from a .sav file and use the same syntax. See Section 8.3 for more details.

8.1.4 Excluding the modal reconstruction computation

Since the computation of the reconstruction matrix is time consuming, it is possible to skip the modal reconstruction computation by setting the keyword `/NOZER`. In this case, only the reconstructed wavefront through iterative Fourier method will be displayed and no information about Zernike modes

will be displayed.

8.1.5 Excluding the iterative reconstruction computation

Sometimes might be necessary to compute only wavefront low order modes and no information on the whole wavefront is required. In order to avoid the computation of the iterative wavefront error and save time and memory, it is possible to skip the iterative reconstruction computation by setting the keyword `/ONLY_ZER`. In this case, only the reconstructed modal wavefront will be displayed and the relative variables will be saved.

8.2 How to save the computed Wavefront

The outputs of the `asong_wfe_reconstruction` procedure can be saved automatically if the keyword `/SAVE_FILE` is set in the call. When setting the keyword `/SAVE_FILE`, the directory and the file name can be also set through the keywords `DIR = DIR` and `NAME = NAME`. The output files "ASONG_wfeNAME.sav" and "ASONG_Zernike_coeffNAME.txt" contain respectively the variables listed in Table 1 and the first `N_ZER` Zernike modes values.

```
IDL > asong_wfe_reconstruction,slope_x, slope_y, pup_mask, wfe_it,
wfe_zer, zer_coeff, /SAVE_FILE, DIR = ''my dir'', NAME =
''my namefile''
```

If the keywords `DIR = DIR` and `NAME = NAME` are not set, the outputs files named "ASONG_wfe.sav" and "ASONG_Zernike_coeff.txt" are saved in the current IDL® folder.

`wfe_it`	Iterative WFE map
`wfe_zer`	Modal WFE map
`zer_coeff`	first N_ZER Zernike values
`mask_pup`	pupil mask where the maps are computed
`wfe_zer_rms`	Modal WFE map RMS
`wfe_it_rms`	Iterative WFE map RMS
`wfe_zer_PtV`	Modal WFE map Peak-to-Valley,
`wfe_it_PtV`	Iterative WFE map Peak-to-Valley

Table 1: Variables stored in "ASONG_wfeNAME.sav"

8.2.1 How to save graphical displayed windows

By default, IDL® graphical windows will pop-up to display WFE maps and Zernike coefficients plots. These window are automatically saved when the keyword `/SAVE_FILE` is set. The two windows are interactive and a few operations can be done clicking on the displayed menu.

8.2.2 How to save the reconstruction Matrix for future use

The modal reconstruction matrix can be particularly time demanding to compute, especially if a number of modes `N_ZER` $\gg 100$ is set. For this reason, it is possible to save the reconstruction matrix `recmat` together with the 3D matrix containing the Zernike modes normalised on the used pupil `mask_pup` setting the keyword `/SAVE_RMAT`.

```
IDL > asong_wfe_reconstruction,slope_x, slope_y, pup_mask, wfe_it,
wfe_zer, zer_coeff, /SAVE_FILE, /SAVE_RMAT, DIR = ''my dir'',
NAME = ''my namefile''
```

This reconstruction matrix can be later restored passed to the `asong_wfe_reconstruction` procedure through the keyword `RMAT = RMAT`.

8.3 How to restore the computed Wavefront

When the WFE maps and main statistics are computed by the `asong_wfe_reconstruction` procedure are saved in the IDL® .sav format, they can be used whenever needed simply restoring the file using the IDL® `RESTORE` procedure.

```
IDL > restore,''my dir'' + ''my namefile'' + ''.sav''
```

8.3.1 How to restore the pre-computed reconstruction matrix

When the reconstruction matrix and the zernike cube 3D array computed by the `asong_wfe_reconstruction` procedure are saved in the IDL® .sav format, they can be used whenever needed simply restoring the file using the IDL® `RESTORE` procedure.

```
IDL > restore,``my dir'' + ``my namefile'' + ``.sav''
```

8.4 Algorithms and formalism description

The main algorithms and formalism used in the code for wavefront calculation are listed below, including references used for the code implementation:

- Iterative Fourier Wavefront reconstruction method [9]. This method computes the Fourier transform of the Wavefront derivatives, divides it by $(u^2 + \nu^2)$ and makes an inverse Fourier transform, to retrieve the initial Wavefront. The process is then iterated. When the error between two iterations becomes smaller than a preassigned level, the next Wavefront estimate is computed and the iterative process is stopped.
- The Modal Wavefront Reconstruction method is based on classical least square estimator [1].
- The inversion of the reconstruction matrix is performed through Singular Value Decomposition.
- For the Zernike polynomials definition, we assumed the Noll formalism [7] and the normalisation described in [8].

Appendix A: Main commands and small tutorial

In this Appendix, a basic sequence of commands to perform the entire Wavefront Analysis is reported. Please look into the specific sections for input, output and keywords definitions. In the folder "TEST" it has been saved a file named "test_image.sav". This image can be used for training and try the basic commands line. To restore the file, use the RESTORE IDL® procedure.
IDL > `restore,''my dir'' + ''test_image.sav''`
The contained variable is called `ima_avg` and it is a pre-reduced ASONG four pupil image of a phase screen.

Make the mask

Basic call:
IDL > `asong_mask_main, ima_avg, centres, quad_mask, pup_mask`
To change the window size and make them smaller:
IDL > `asong_mask_main, ima_avg, centres, quad_mask, pup_mask, WSCALE = 0.4`
To change the image display scale:
IDL > `asong_mask_main, ima_avg, centres, quad_mask, pup_mask, LOG = 0.5`
For automatic pupil edge detection:
IDL > `asong_mask_main, ima_avg, centres, quad_mask, pup_mask, /AUTO, THRESHOLD = 0.15`
To set a pupil radius different from 450 pixels (default value):
IDL > `asong_mask_main, ima_avg, centres, quad_mask, pup_mask, RAD = 480`
To change the pupil radius:
IDL > `asong_mask_main, ima_avg, centres, quad_mask, pup_mask, RAD = 400, /CEN`
To save the pupil mask:
IDL > `asong_mask_main, ima_avg, centres, quad_mask, pup_mask, /SAVE_FILE, DIR = ''my dir'', NAME = ''my namefile''`

Compute the slopes

Basic call:
IDL > asong_slope, ima_avg, quad_mask, pup_mask, centres, slope_x, slope_y
To display the preview of the slopes:
IDL > asong_slope, ima_avg, quad_mask, pup_mask, centres, slope_x, slope_y, /DISP
To save the slopes into a file:
IDL > asong_slope,ima_avg, quad_mask, pup_mask, centres, slope_x, slope_y, /SAVE_FILE, DIR = ‘‘my dir’’, NAME = ‘‘my namefile’’

Compute the Wavefront Error

Basic call:
IDL > asong_wfe_reconstruction, slope_x, slope_x, pup_mask, wfe_it, wfe_zer, zer_coeff
To save the reconstructed wavefront and its statistics:
IDL > asong_wfe_reconstruction, slope_x, slope_y, pup_mask, wfe_it, wfe_zer, zer_coeff, /SAVE_FILE, DIR = ‘‘my dir’’,
NAME = ‘‘my namefile’’
To save the reconstruction matrix RMAT and the Zernike cube matrix:
IDL > asong_wfe_reconstruction, slope_x, slope_y, pup_mask, wfe_it, wfe_zer, zer_coeff, /SAVE_FILE, /SAVE_RMAT, DIR = ‘‘my dir’’,
NAME = ‘‘my namefile’’
To use a pre-computed reconstruction matrix:
IDL > asong_wfe_reconstruction,slope_x, slope_x, pup_mask, wfe_it, wfe_zer, zer_coeff, zermat, RMAT = RMAT
To avoid display and modal reconstruction:
IDL > asong_wfe_reconstruction, slope_x, slope_x, pup_mask, wfe_it, wfe_zer, zer_coeff, DISPLAY = 0, /NOZER
To avoid iterative reconstruction:
IDL > asong_wfe_reconstruction, slope_x, slope_x, pup_mask, wfe_it, wfe_zer, zer_coeff, /ONLY_ZER

Appendix B: Main IDL® Procedures description

In this Appendix, the header of the main procedures of the ASONG DAS are reported.

asong_mask_main

ROUTINE NAME:
asong_mask_main

PURPOSE:
Given an image of the four pupils, it is used to define the quadrant mask for slope computation

CALLING SEQUENCE:
asong_mask_main, image, centres, qmask, pup

INPUT:
image: 2D array containing the image of the four pupils

OPTIONAL INPUT:

centres:	when the keyword /CEN is set, a 2D array containing the centres of the four pupils in the correct order can be passed as an input

OUTPUT:

qmask:	3D array with four planes, each representing a binary mask associated to one quadrant of the sensor
pup:	pupil mask, logical AND of four masks
centres:	4 pupils centres

KEYWORDS:

save_file: Variable type: LOGICAL. Use this keyword to save the mask and the output variables

dir: Variable type: STRING. Set this keyword to define the directory in which the variables must be stored.

name: Variable type: STRING. Set this keyword to set the name of the file. DEFAULT VALUE = 'ASONG_mask'

dimpup: Variable type: SCALAR. The dimension of the array that will contain the pupil for the wavefront calculation. It can be slightly larger than the calculated pupil. DEFAULT VALUE = 900

log: Variable type: LOGICAL. Set this keyword to display more contrasted images to help some clicking operation

auto: Variable type: LOGICAL. Set this keyword to use SOBEL algorithm to find the pupils edges. **WARNING!!!** t is recommended to use this keyword only in case light distribution in the 4 pupils is homogeneous

threshold: Variable type: Scalar, float. Normalised value (less than 1). Set the threshold for the automatic mask definition. DEFAULT VALUE = 0.1 of the max

n_pupils: Variable type: Scalar, integer. DEFAULT VALUE = 4. Number of pupils in the image. 1 and 4 are acceptable

wscale: Variable type: Scalar. Graphical window scaling factor w.r.t. image size. DEFAULT VALUE = 0.5

silent: Variable type: LOGICAL. Set this keyword to avoid unecessary output in the IDL Console.

rad: Variable type: Scalar. DEFAULT VALUE = 450 image pixels. Pupil mask radius

dimpup: Variable type: Scalar. DEFAULT VALUE = 900 image pixels. The dimension of the array that will contain the pupil for the wavefront calculation. It can be slightly larger than the calculated pupil.

cen: Variable type: LOGICAL. If set, the centres variable is used as an input variable and the centers coordinates are used to compute the mask.

asong_slope

ROUTINE NAME:
asong_slope

PURPOSE:
Given a frame and the binary masks defining the four quadrants of the ASONG WFS, compute the wavefront derivatives in X and Y

CALLING SEQUENCE:
asong_slope, frame, qmask, pup, centres, sx, sy

INPUT:

image:	measured frame with subtracted background and divided by the flat (if required)
qmask:	3D array with four planes, each representing a binary mask associated to one quadrant of the sensor
pup:	footprint of sx and sy
centres:	4 X 2-elements vector of coordinates of pupils centroids

KEYWORDS:

gain:	when this keyword is set, the gain is not computed and the value set in the keyword is used instead.It can come from calibration procedures. DEFAULT VALUE = 0
pyr:	Set this keyword in case a Pyramid WFS is used in stead of ASONG. DEFAULT VALUE = 0
dxf, dyf:	GTF x and y period in mm. If only one of the two keywords is set, the period in X and Y is considered to be the same. DEFAULT VALUE = 9.05 mm
foc:	distance from the aperture and the Fourier plane
display:	set this keyword to visualise the computed slopes in a window
pup_scale:	System magnification. Theoretical value is given by the ratio between the focalising optic focal length and the MLA focal length. This parameter should be measured in the lab by re-imaging in the pupil an object of known length

det_px: detector pixel size in microns. DEFAULT VALUE = 2.4

OUTPUTS:

sx, sy: array of signals in X and Y

asong_wfe_reconstruction

ROUTINE NAME:
asong_wfe_reconstruction

PURPOSE:
Given the Wavefront X and Y slopes and the pupil mask, it reconstructs the phase through iterative method proposed by F. Roddier & C. Roddier ("Wavefront reconstruction using iterative Fourier transforms,"Applied Optics, 30, 1325, (1991)) and through (low order) modal reconstruction using Zernike modes.

CALLING SEQUENCE:
asong_wfe_reconstruction, , sy, pup, wfe_it, wfe_zer, zer_coeff

INPUTS:

sx : Wavefront slope X

sy : Wavefront slope Y

pup : footprint of sx and sy

OPTIONAL INPUTS:

rmat: See the keyword RMAT. Precomputed Reconstruction matrix

zermat: 3D Array containing the Zernike modes normalised on the pupil. To be visualised they have to be multiplied by the pupil array

KEYWORDS:

save_file: Variable type: LOGICAL. Use this keyword to save the mask and the output variables

dir: Variable type: STRING. Set this keyword to define the directory in which the variables must be stored.

name: Variable type: STRING. Set this keyword to set the name of the file. DEFAULT VALUE = 'ASONG_mask'

notilt: Variable type: LOGICAL. Set this keyword to subtract the average value to slope maps. DEFAULT VALUE = 1

diplay: Variable type: LOGICAL. Set this keyword to have displayed the wavefronts once computed. DEFAULT VALUE = 1

rmat: Variable type: 2D array. This keyword can be used to pass to the procedure the reconstruction matrix if already computed. If the reconstruction matrix RMAT is not available, this keyword can be used to have it as an output and use it for a second iteration.

n_zer: Variable type: SCALAR. Number of computed Zernike modes. DEFAULT VALUE = 20. WARNING!!!! Since the pupil resolution is very high (around 1000 X 1000 pixels) the nominal value of Zernike modes that can be used for the reconstruction is very high. The computation time and the memory required for an high number of Zernike modes can be prohibitive. If possible, use this only with low number of Zernike (less than few hundreds)

sigma_cut: NOT IMPLEMENTED!!! Data more than this number of standard deviations from the median is ignored. Suggested values: 2.0 and up. DEFAULT VALUE = 3

calibration : If set, the slopes are multiplied by a coefficient computed during calibration operations. DEFAULT VALUE = 1

save_rmat: Variable type: LOGICAL. Use this keyword to save the reconstruction matrix and the Zernike cube

lambda : Variable type: SCALAR. DEFAULT VALUE: 0.633 microns

waves: Variable type: LOGICAL. If set, returns the values in waves

nozer : Variable type: LOGICAL. If set, only iterative reconstruction will be performed and all concerning modal reconstruction will be ignored

only_zer : Variable type: LOGICAL. If set, only modal reconstruction will be performed and all concerning iterative reconstruction will be ignored

OUTPUTS:

wfe_it: Calibrated reconstructed WAVEFRONT throught the method of R&R

wfe_zer: Low order reconstructed WAVEFRONT throught modal reconstruction

zer_coeff: Zernike coefficients coming from modal reconstruction

OPTIONAL OUTPUTS:

rmat: See the keyword RMAT. Precomputed Reconstruction matrix

zermat: 3D Array containing the Zernike modes normalised on the pupil. To be visualised they have to be multiplied by the pupil array

9 Links

9.1 Libraries:

- https://idlastro.gsfc.nasa.gov/
- http://adsabs.harvard.edu/abs/2009ASPC..411..251M
- https://www.ict.inaf.it/gitlab/laura.schreiber/starfinder2

9.2 Patents:

https://patentscope.wipo.int/search/fr/detail.jsf?docId=WO2020156867

References

[1] R. Cubalchini. "Modal wave-front estimation rom phase derivative measurements". In: *Journal of the Optical Society of America (1917-1983)* 69.7 (July 1979), p. 972.

[2] E. Diolaiti et al. "Analysis of isoplanatic high resolution stellar fields by the StarFinder code". In: *Astronomy and Astrophysics Supplement* 147 (Dec. 2000), pp. 335–346. DOI: 10.1051/aas:2000305. arXiv: astro-ph/0009177 [astro-ph].

[3] François Henault and Alain Spang. "System for inspecting surfaces of an optical wave using a graduated density filter". In: WO2020156867 (2020).

[4] François Hénault et al. "Crossed-sine wavefront sensor for adaptive optics, metrology and ophthalmology applications". In: *Engineering Research Express* 2.1, 015042 (Mar. 2020), p. 015042. DOI: 10.1088/2631-8695/ab78c5.

[5] C. B. Markwardt. "Non-linear Least-squares Fitting in IDL with MPFIT". In: *Astronomical Data Analysis Software and Systems XVIII*. Ed. by D. A. Bohlender, D. Durand, and P. Dowler. Vol. 411. Astronomical Society of the Pacific Conference Series. Sept. 2009, p. 251. arXiv: 0902.2850 [astro-ph.IM].

[6] Jorge J. Moré. "The Levenberg-Marquardt algorithm: Implementation and theory". In: *Lecture Notes in Mathematics, Berlin Springer Verlag*. Vol. 630. 1978, pp. 105–116. DOI: 10.1007/BFb0067700.

[7] R. J. Noll. "Zernike polynomials and atmospheric turbulence." In: *Journal of the Optical Society of America (1917-1983)* 66 (Mar. 1976), pp. 207–211.

[8] F. Rigaut and E. Gendron. "Laser guide star in adaptive optics: the tilt determination problem." In: *Astronomy and Astrophysics* 261 (Aug. 1992), pp. 677–684.

[9] Francois Roddier and Claude Roddier. "Wavefront reconstruction using iterative Fourier transforms". In: *Applied Optics* 30.11 (Apr. 1991), pp. 1325–1327. DOI: 10.1364/AO.30.001325.

[10] Laura Schreiber et al. “Starfinder2: a software package for identification and analysis of point-like sources in adaptive optics images with spatially variable PSF”. In: *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*. Vol. 11448. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series. Dec. 2020, 114480H, 114480H. DOI: 10.1117/12.2564105.

[11] S. Thomas et al. “Comparison of centroid computation algorithms in a Shack-Hartmann sensor”. In: *MNRAS* 371.1 (Sept. 2006), pp. 323–336. DOI: 10.1111/j.1365-2966.2006.10661.x.

List of Figures