Title

Modeling the pyramidal sensor by a ZEMAXTM user defined surface.

Journal

Adaptive Optics for Extremely Large Telescopes 4 – Conference Proceedings, 1(1)

Authors

Antichi, Jacopo
Munari, Matteo
Magrin, Demetrio
et al.

Publication Date

2015

DOI

10.20353/K3T4CP1131618

Modeling the pyramidal sensor by a ZEMAX^{TM} user defined surface.

Jacopo Antichi[1, *], Matteo Munari[2], Demetrio Magrin[3], Armando Riccardi[1]

[1]INAF-OAA, Largo Enrico Fermi 5, 50125 Firenze, Italia
[2]INAF-OACT, Via Santa Sofia, 95123 Catania, Italia
[3]INAF-OAPD, Via Vicolo Osservatorio 5, 35100 Padova, Italia
[*]antichi@arcetri.inaf.it

Abstract: After 20 years of wavefront sensors based on pyramid (PWFS), there are no straightforward ways to model such device in standard sequential ray-tracing software: modeling strategies tend to be oriented to the needs of the single user only and, in general, are unsatisfactory due to lack of flexibility. To overcome this problem, we have exploited the possibility of ZEMAX™ – one of the ray-tracing software mostly in use nowadays – to develop a user defined surface (UDS), whose properties are described in a dynamic link library (DLL) written in C language. The pyramid UDS approach greatly improves the versatility during the design and simplifies both quality and tolerance analysis. In order to prove the potentiality of our UDS-DLL surface, referred as PAM2R, we reproduced the optical layout of two PWFS systems already installed at LBT: the single-conjugate system FLAO, and the ground-layer system GWS of LINC-NIRVANA. In this proceeding we will highlight the main characteristics of the PAM2R surface, showing various results we obtained on the above case studies with the aim to establish a common design playground for the PWFS in the AO community.

1. Introduction

Modeling PWFS is a complex task in ray-tracing due to the nature of its main optical component. This component consists of a refractive pyramidal surface with its vertex located in a focal plane pointing in the direction of the incoming light (Ragazzoni 1996). It can consist also of two consecutive refractive pyramidal surfaces, the first one having vertex located in a focal plane pointing in the direction of the incoming light, while the second one pointing against the direction of the incoming light and acting as an optic relay of the beam emerging by the first pyramidal optic (Tozzi et al. 2008). Finally, it can consist of a reflective pyramidal surface with vertex located in a focal plane as (Ragazzoni 1996), but in this case slicing the input optical beam in four output optical beams (Wang et al. 2010).

So far, ray-tracing of a pyramidal surface has been addressed with different methods, but none of them is fulfilling the goal of a complete consistency with true PWFS layouts. These methods, in fact, address only some aspects of the propagation throughout a pyramidal optics, not the whole set, which limits the realm of applications of such kind of modeling. For example, a dioptric pyramidal surface is feasible exploiting four prismatic surfaces in the multi-configurations mode, or, alternatively, using a unique non-sequential object mimicking the pyramidal surface itself. However, the multi-configurations approach may be useful for the geometrical design purposes. Instead, any attempt of a physical optical propagation will be meaningless in this case just because it is incompatible with the multi-configuration tool, independently by the adopted ray-tracing software. On the contrary, the use of a non-sequential object to define a pyramidal surface allows exploiting the physical propagation tool, which is useful to look e.g. at the diffraction effects throughout a pyramidal optic, but it may fail because a non-sequential surface is represented – generally in ray-tracing software – by a large number of discrete components. Depending on the adopted spatial sampling, the incoming beam may not hit them all, degrading the accuracy of the Fourier transforms used to retrieve the phase map along the optical path. Moreover, a non-sequential object has to be defined for each considered pyramid configuration, reducing the flexibility and making unfeasible any attempt of tolerance analysis.

Another important issue concerning a pyramidal optics simulation is the sequential ray-tracing of the vertex and the edges between its adjacent surfaces. Indeed, a nominal sequential ray should start from an input plane and should end onto an output plane, propagating between the two with a unique path following the refraction/reflection laws. Hence, the emerging ray will belong to the plane containing the incident ray and the surface normal vector at the intersection ray-surface. This is problematic for the vertex and the pyramid surface edges, because they represent mathematical singularities for the normal definition. Commonly, the user avoids to consider such particular points (i.e. the ones of non-derivability of the surface sag function) in the sequential propagation, just because standard ray-tracing fails the correct propagation. This fact may even result not as vignetting of such rays at the pyramidal surface output, but as incorrect propagation of rays after the pyramidal surface. Practically, in the case of a pyramidal surface, this obliges the user to figure out tricks to skip rays passing through the pyramid vertex and surface edges.

With these problematics in mind, we have developed a user-defined dynamic-link-library surface whose ray-tracing properties are fully described on a proper C code, and working as standard surface of the ray-tracing software ZEMAX™ (Antichi et al. 2015). Pyramid vertex and edges are accounted in the code with explicit rules of propagation, and they are considered without any wideness just to perform the propagation through an ideal pyramidal surface. Consistency checks, based on ray propagation statistics, certify our recipe deals with vertex and edges in a way compliant with refraction/reflection propagation rule once large numbers of rays are running in the ray-tracing software. This UDS-DLL surface – we named PAM2R – is defined by an independent set of parameters explicit to the user. This method permits both (i) to manage the pyramidal surface during the ray-tracing modeling, allowing control of the whole set of dependent parameters, (ii) to lower error propagation, which is of benefit to any subsequent sensitivity analysis. We tested PAM2R to model both geometrical and physical propagations of existing PWFS layouts at the Large Binocular Telescope (Esposito et al. 2003 & Farinato et al. 2008). PAM2R DLL can be downloaded from the web site http://pam2r.arcetri.astro.it/#services.

2. Algorithms defining surface rendering and ray-tracing with PAM2R

In this section we give the formalism adopted in the PAM2R source code to define a pyramidal surface and trace rays through it.[1]

2.1 Definitions

Rays propagation is from $z < 0$ toward $z > 0$. Pyramid vertex is defined on the $z = 0$ plane. In equations, scalar quantities are set lower case, no bold. Vector quantities are set lower case, bold. The symbol $|$ indicates the scalar product in the Euclidean tri-dimensional affine space with origin at V.

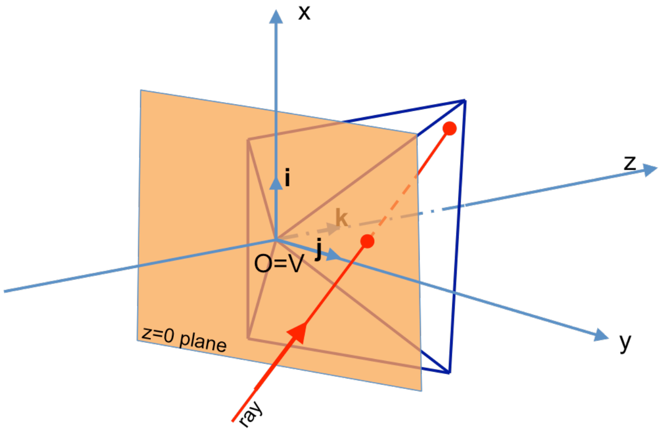


Fig. 1:Scheme of a pyramidal surface as defined in the PAM2R source code. Orange plane is the z = 0 plane at the vertex (V) of the pyramdial surface. A single ray intercepts such plane and the pyramidal surface onto 1 of its 4 faces. Origin (O) of the cordinate system is defined to be coincident to V. Space orientation follows the right hand rule.

---

[1] We emphasize that our formalism allows ray tracing through a catoptric pyramidal surface also.

2.2 Ray parametrization

A ray is defined by the following set of parameters:

- its position coordinates referred to the z = 0 plane $(x_0, y_0, 0)$
- its direction cosines (α, β) with respect to x and y axis respectively
- the distance from the position vector $\mathbf{r_0}$ and the point where the ray intercepts the pyramidal surface: t. Hence, the equation of the ray is given by its position vector **r**:

$$\mathbf{r} = \mathbf{r}_0 + t\,\mathbf{e} \qquad (1),$$

where the the intersection point $\mathbf{r_0}$ and the unitary vector **e** and are defined as:

$$\begin{aligned} \mathbf{r}_0 &= \left[x_0, y_0, 0\right] \\ \mathbf{e} &= \left[\alpha, \beta, \sqrt{1-\alpha^2-\beta^2}\right] \end{aligned} \qquad (2).$$

2.3 Intersection point of a ray with a face and membership of this point with such face

The equation of a plane is defined exploiting the vector orthogonal to such plane and one point belonging to the plane. Being **n** the normal to a face and $\mathbf{r_V}$ the position vector of the piramid vertex V, the equation of the plane containing a face is:

$$(\mathbf{r} - \mathbf{r}_V) \,|\, \mathbf{n} = 0 \qquad (3).$$

Combining it with eq. (1) we found the intersection point **r** as:

$$\mathbf{r} = \mathbf{r}_0 - \frac{\mathbf{r}_0 \,|\, \mathbf{n}}{\mathbf{e} \,|\, \mathbf{n}}\,\mathbf{e} \qquad (4).$$

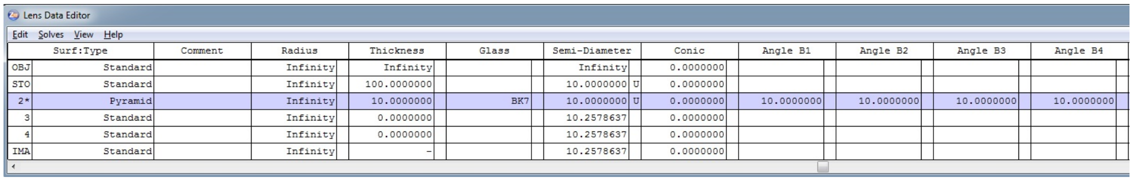
This is valid only when the ray and the plane containing the face are not parallel to each other.

3. Implementation of PAM2R in ZEMAX^{TM} as UDS-DLL surface

PAM2R is implemented as a UDS surface exploiting a DLL file using the coding conventions described in the ZEMAX^{TM} manual, following the example file: `usersurface.c`. In detail, for each ray, defined by its director cosines (α,β), the position vector $\mathbf{r}_0$ is evaluated. Thus, pyramidal surface *base* and *tilt angles* fix the pyramid geometry. Hence, the propagation through the pyramidal surface is output following the described formalism.

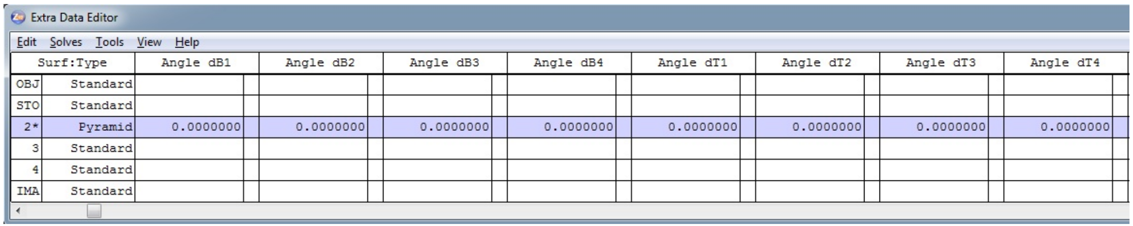
3.1 Pyramidal surface input parameters and rendering of its base and tilt angles

At the software graphical user interface (GUI), PAM2R exhibits parameters both in the lens data editor (LDE) and in the extra data editor (EDE). In the LDE GUI, aside from typical parameters proper to surfaces (radius of curvature, material, diameter, etc.) PAM2R allows indicating the four *base angles* in degrees named: B1, B2, B3, B4. In the EDE GUI, height extra parameters permit to further customize the pyramidal surface. Specifically, from angle dB1 to angle dB4, PAM2R allows to introduce errors on the pyramid *base angles* set, letting their actual values becoming: B1+dB1, B2+dB2, B3+dB3, B4+dB4. Four further parameters: dT1, dT2, dT3, dT4 fix the variation of the *tilt angles*, whom referring values: T1, T2, T3, T4, are set equal to zero.



	Surf:Type	Comment	Radius	Thickness	Glass	Semi-Diameter	Conic	Angle B1	Angle B2	Angle B3	Angle B4
OBJ	Standard		Infinity	Infinity		Infinity	0.0000000				
STO	Standard		Infinity	100.0000000		10.0000000 U	0.0000000				
2*	Pyramid		Infinity	10.0000000	BK7	10.0000000 U	0.0000000	10.0000000	10.0000000	10.0000000	10.0000000
3	Standard		Infinity	0.0000000		10.2578637	0.0000000				
4	Standard		Infinity	0.0000000		10.2578637	0.0000000				
IMA	Standard		Infinity	-		10.2578637	0.0000000				

Fig. 2: Example of a PAM2R prescription row within the ZEMAX^{TM} LDE GUI. *Base angles* default is 10 degrees.

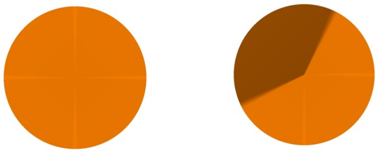


Extra Data Editor

Edit Solves Tools View Help

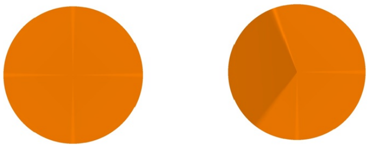
Surf:Type		Angle dB1	Angle dB2	Angle dB3	Angle dB4	Angle dT1	Angle dT2	Angle dT3	Angle dT4
OBJ	Standard								
STO	Standard								
2*	Pyramid	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000	0.0000000
3	Standard								
4	Standard								
IMA	Standard								

Fig. 3: Example of a PAM2R prescription row within the ZEMAX^{TM} EDE GUI. Variation to both *base* and *tilt angles* default is 0 degrees.



(1) (2)

Fig. 4: Pyramidal surface comparison: top view. (1) perfect pyramidal surface. (2) imperfect pyramidal surface showing an exaggerated 30 degrees error of the *base angle* on its upper-left face.



(1) (2)

Fig. 5: Pyramidal surface comparison: top view. (1) perfect pyramidal surface. (2) imperfect pyramidal surface showing an exaggerated 30 degrees error of the *tilt angle* on its upper-left face.

3.2 Pyramidal surface vertex and edge treatment

The easiest possibility to treat with rays passing close to the pyramid vertex or edges is simply to terminate them, so that they were considered as vignetted by the ray-tracing. This approach even if correct, would lead to problems like: i) a quite large number of rays terminated during the ray-tracing, ii) some glitches in the visualization of the surface (given that the DLL is used by ZEMAX^{TM} to generate plots and images of the simulated system).

To resolve this problem, PAM2R adopts a different strategy: if a ray is vignetted by the edges or the vertex of the pyramidal surface, PAM2R perturbs its initial position coordinates (x_0,y_0,0) by a small random amount (some times the `FLT_MIN`), and the ray is traced again. In this way, the ray may land on one of the faces of the pyramidal surface. It is possible that again that the ray (even if perturbed) hits no face of the pyramidal surface. To resolve this issue, the adopted rule is to not to repeat the procedure and to accept a small fraction of vignetted rays. This fraction is less meaningful as the total member of rays hitting the pyramidal surface increases, and it becomes irrelevant for large numbers of traced rays, as Figure 6 explains.

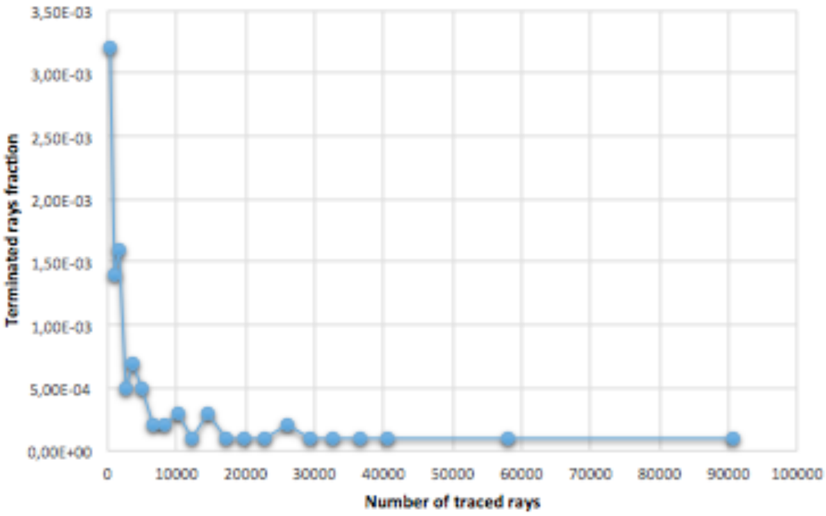


Fig. 6: Fraction of terminated rays (vignetted) vs. traced rays for an optical setup where the pyramidal optic is illuminated by a collimated beam. The fraction of terminated rays due to vignetting by the edges or the vertex converges toward 0 with the growing number of traced rays, as expected by the PAM2R prescription.

4. Modeling existing PWFS at the Large Binocular Telescope with PAM2R

In this section, we present some tests showing that PAM2R is able to model a pyramidal optics system both by geometrical optical propagation and by physical optical propagation tools provided in standard ray-tracing software like ZEMAX^{TM}. These verifications apply to two PWFS operating at LBT: the single conjugate FLAO (Esposito et al. 2003) and the GWS of LINC-NIRVANA (Farinato er a. 2008) wavefront sensors.

4.1 Modeling the FLAO PWFS

The LBT FLAO PWFS is the wavefront sensor for the Fizeau interferometer (LBTI) and the multi-object spectrographs (LUCI1 and LUCI2). In this case, we have verified that PAM2R reproduces almost perfectly the FLAO design obtained by the recipe adopted in (Tozzi et al. 2008), which makes use of two non-sequential objects to design the two adjacent pyramidal surfaces shaping the pyramidal optics of this wavefront sensor.

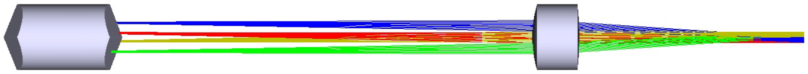


Fig. 7: Rendering of the PWFS sub-unit of FLAO consisting of the double-pyramid optics and the relay optics, allowing the reimaging of the four pupils. The beam line goes from left to right and enters on the double-pyramidal optic at its 1st vertex laying at a f/45 focus . 1st surface *base angles* all equal to 30.00 degrees, 2nd surface all equal to 28.31 degrees. Colors are field points.

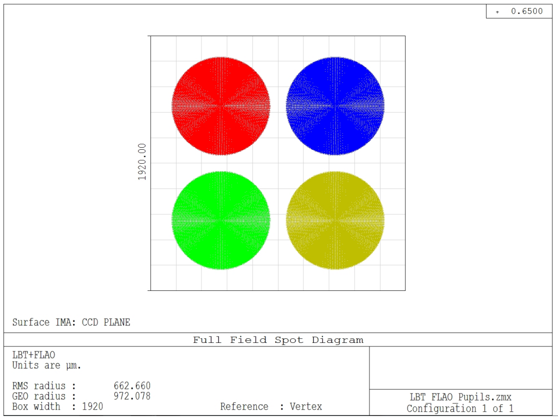


Fig. 8: Geometrical re-imaging at the FLAO PWFS pupil plane. The square side is in microns and corresponds to 80 pixels (with size equals 0.024 mm). According to the FLAO design (Esposito et al. 2003), single pupil size is 30 pixels and the distance between adjacent pupils is 36 pixels.

4.2 Modeling the LINC-NIRVANA GWS

LINC-NIRVANA is a near infrared camera for LBT devoted to interferometry. Its AO correction is based on multi conjugate adaptive optics (MCAO) techniques by means of two separated systems: the ground layer wavefront sensor (GWS) and the mid-high layers wavefront sensor (MHWS). GWS (Farinato et al. 2008) takes advantage of the layer oriented AO technique by optically superimposing on the detector the light coming from the reference stars, thanks to up to twelve fully deployable star enlargers (SE). In detail, each SE is made of two achromatic doublets that re-focus and enlarge the reference star image, which forms at the LBT focal plane, on the vertex of a single pyramid. Then, a pupil re-imaging camera creates four on the detector for each reference star. Due to the restored telecentricity downstream the SE optic, any single image superimposes to the others at the detector plane. Figures 9, 10 and 11 explain this optical concept. In this case, we have verified that PAM2R reproduces the same results as the original non-sequential design based on a single-pyramid approach. Thanks to its flexibility and compatibility with the multi-configurations tool of ZEMAX$^{\mathrm{TM}}$, we have also verified performances for several SE configurations across the whole GWS field of view, which is an annulus having inner diameter equal to 2 arcminutes and outer diameter equal to 6 arcminutes. It has to be underlined that with PAM2R, each pyramid can be easily modeled separately allowing to evaluate the impact on the performances due to manufacturing errors on both *base* and *tilt angles*. Especially in case of multiple PWFS like GWS, this is mandatory to establish more reliable error budgets describing this kind of PWFS.

Fig. 9: Rendering of a single star enlarger of GWS made of two achromatic doublets and a single pyramid (red color). The beam line goes from left (entrance focal plane delivered by LBT) to right, where the pyramidal optic lies. In the model, the pyramidal surface vertex angle is 0.5526 degrees. Colors are different field points. Note that GWS exploits a unique pyramidal optics, not a double one like in the FLAO design.

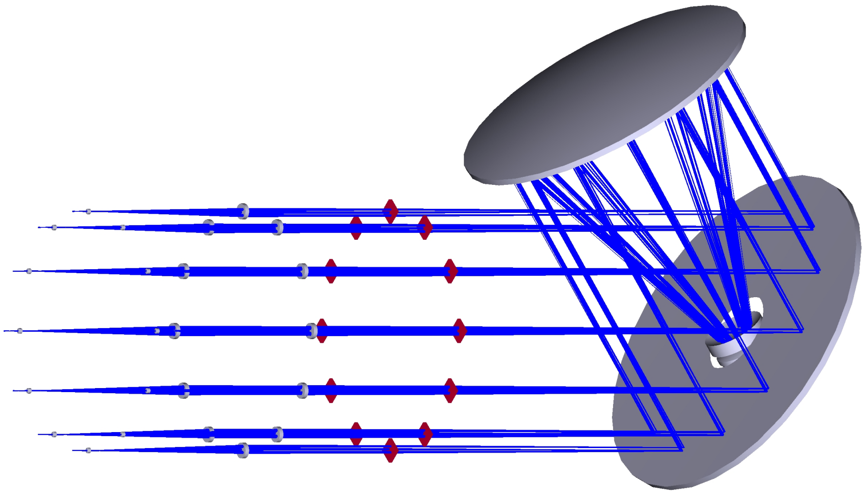


Fig. 10: Rendering of GWS module made of twelve star enlargers in circular configuration and the pupil re-imaging camera. The beam line goes from left (entrance focal plane delivered by LBT) to right. Pyramids are red colored.

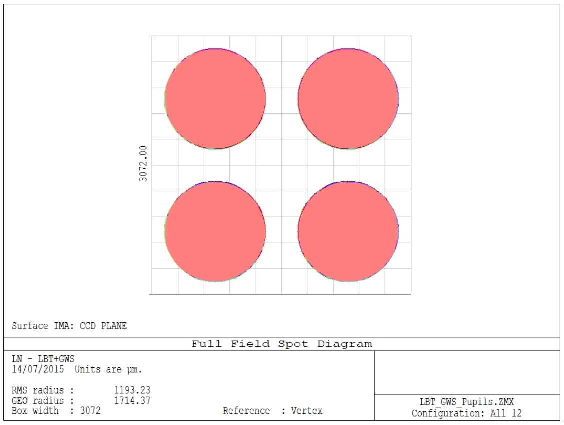


Fig. 11: Geometrical re-imaging at the GWS pupil plane of the reference stars shown in Fig. 10. Colors represent the whole set of twelve configurations. The square side is in microns and corresponds to 128 pixels (with size equals 0.024 mm). According to the GWS design (Farinato et al. 2008), single pupil diameter is 48 pixels and the distance between side pupils centers is 64 pixels.

5. Conclusions

This paper describes a user-defined dynamic-link-library surface named PAM2R. Its source code is procedural and written in C. Its output is the optical beam propagation throughout a pyramidal surface. Consistency tests were made using ZEMAX™ on existing pyramidal wavefront sensor layouts working on AO systems in operation at LBT. Specifically, i) both geometrical and physical propagations obtained by PAM2R match the ray-tracing output of FLAO, ii) the multi-PWFS design proper to the LINC-NIRVANA GWS we obtained with PAM2R matches with its original design, this latter is based on a rather complex non-sequential ray-tracing prescription for the twelve pyramidal optics. Hence, PAM2R represents a suited tool to design pyramidal wavefront sensors, as *easily* done for other wavefront sensor types.

References

Ragazzoni, R., “Pupil plane wavefront sensing with an oscillating prism”. J. Modern Opt. 43, 289-293 (1996)

Tozzi, A., Stefanini, P., Pinna, E., Esposito, S., “The Double Pyramid wavefront sensor for LBT”. Proc. SPIE 7015 701558 (2008)

Wang, A., Yao, J., Cai, D., Ren, H., “Design and fabrication of a pyramid wavefront sensor”. Optical Eng. 49 (7) 073401 (2010)

Antichi, J., Munari, M., Magrin, D., Riccardi, A., “Modeling pyramidal sensors in ray-tracing software by a suitable user defined surface ”. JATIS - submitted (2015).

Esposito, S., Tozzi, A., Ferruzzi, D., Carbillet, M., Riccardi, A., Fini, L., Vérinaud, C., Accardo, M., Brusa, G., Gallieni, D., Biasi, R., Baffa, C., Biliotti, V., Foppiani, I., Puglisi, A., Ragazzoni, R., Ranfagni, R., Stefanini, P., Salinari, P., Seifert, W., Storm, J., “First Light Adaptive Optics System for Large Binocular Telescope”. Proc. SPIE Vol. 4839 0277 (2003)

Farinato, J., Ragazzoni, R., Arcidiacono, C., Brunelli, A., Dima, M., Gentile, G., Viotto, V., Diolaiti, E., Foppiani, I. Lombini, M., Schreiber, L., Bizenberger, P., De Bonis, F., Egner, S., Gässler, W., Herbst, T., Küster, M., Mohr, L., Rohloff, R-R., “The Multiple Field of View Layer Oriented wavefront sensing system of LINC-NIRVANA two arcminutes of corrected field using solely Natural Guide Stars”. Proc. SPIE Vol. 7015, 70155J (2008)