**International Conference on Space Optics—ICSO 2018**

Chania, Greece

9–12 October 2018

*Edited by Zoran Sodnik, Nikos Karafolas, and Bruno Cugny*

# *Freeform optics design tool for compact spectrometers*

*M. David Nijkerk*

*J. Michael Gruber*

*Bart Boonacker*

International Conference on Space Optics — ICSO 2018, edited by Zoran Sodnik, Nikos Karafolas, Bruno Cugny, Proc. of SPIE Vol. 11180, 1118022 · © 2018 ESA and CNES · CCC code: 0277-786X/18/$18 · doi: 10.1117/12.2535993

# Freeform Optics Design Tool for Compact Spectrometers

M. David Nijkerk, J. Michael Gruber*, Bart Boonacker
TNO, Stieltjesweg 1, 2628 CK, Delft, The Netherlands

## ABSTRACT

We present a novel optical design tool that makes use of an evolutionary global optimization algorithm. The algorithm has several characteristics that make it well-suited for freeform optics design. With the design tool it is no longer necessary to make the distinction between paraxial degrees of freedom and degrees of freedom related to freeform surface description. The design process, which typically involves a multi-stage scheme consisting of finding an optimal paraxial starting layout, optimization, gradually including freeform degrees of freedom to yield an optimal nominal design, and finally a step in which the as-built design is optimized, is shortened because optimal paraxial starting point and optimal freeform shapes are combined to a single optimization step. Optionally, as-built performance can be included in this step as well. The design tool is applied to the design of a compact spectrometer.

**Keywords:** Freeform Optics, Optical Design, Optimization, Spectrometers, CMA-ES, Evolutionary Strategies

## 1. INTRODUCTION

For next generation space missions optical instruments need to be compact, lightweight and cost-effective, but should show a higher performance in terms of optical performance, stability and robustness at the same time. Solutions can be found by using freeform optics. The technological benefits of using free-forms are discussed in literature[1] and usually a set of advantages are given, of which the most relevant for the present case are:

- Less optics can be used in the opto-mechanical system and therefore a decrease in the amount of optical surfaces occurs. Since every surface is a reduction of light intensity (e.g. by scattering), a higher throughput of the optical system is the result.
- Less optics also means a reduction in mass and size. This benefit is in particular of interest for space-based instruments, as the usage of free-forms would reduce the required volume for the same instrument and thereby reducing the costs of a space mission.
- An improvement in optical quality (e.g. spherical aberration, coma, distortion)
- A more favourable position of the optical components is possible

With the advent of new developments in optical manufacturing technologies, such as single point diamond turning, opto-mechanical systems are no longer limited by the shapes realizable by manufacturing, but by the solutions proposed by the optical designers.

The general process of optimization begins with the selection of the starting configuration, which is done, traditionally, on the basis of patents and lens databases, first-order layouts, previous research, experience or global searches in the merit function space. The choice of the initial configuration is critical for the result of local optimization. Poor starting configurations often lead to solutions with bad imaging quality. Once the starting point is defined, the optical designer optimizes the layout of the system. The purpose of the optimization is to minimize the merit function of the optical system and to reach a nominal solution. If the solution satisfies the originally stated requirements, opto-mechanical multidisciplinary studies are performed to check that the system can be realized. A tolerance analysis and analyses related to manufacturing, assembly, integration and test (MAIT) are performed. The goal of these analyses is to predict the as-built performance of the system. If this performance is compatible with the instruments requirement the system design is finished. The traditional design process can be illustrated with the following diagram:

*michael.gruber@tno.nl

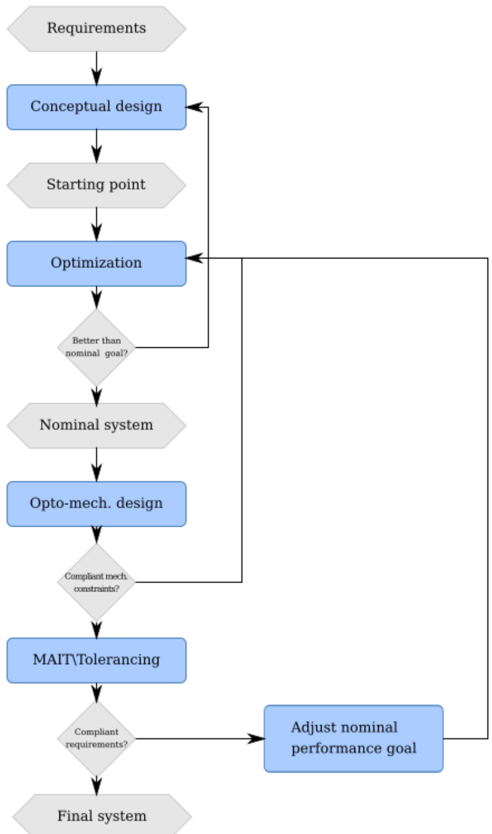


Figure 1. Traditional optical design scheme.

Using freeform optics in optical design adds to the cost and complexity of the design and engineering process for several reasons. General freeforms have many parameters, and consequently, degrees of freedom. For the latter reason they may naively seem attractive from the point of view of an optical designer, providing many knobs to optimize the performance; however, the overwhelming number of parameters may also cause a hotchpotch of local optima in merit function landscape that are difficult to escape from with commonly used optimization means[2]. Many of those will not lead to an improved design because e.g. the particular freeform representation does not provide the correct handles, the location in the optical train is unsuitably chosen, or the mathematical description of the surface is ill-suited for the optimization process. The many different possible mathematical representations for freeform surfaces may lead to very different performance of the optimized solution[3]. Yet another difficulty is the problem of the creation of starting configurations. An appropriate conventional (i.e. non-freeform) design can often be obtained as an optimization from a paraxial system that can be defined semi-analytically with just a few parameters. Then, the paraxial design is a good starting configuration for an optimization algorithm as is used in optical design software in the hunt for an optimum design. In contrast, for a design that includes non-symmetric optics, the parameters that provide the departure from symmetry are not so easily obtained with an analytically determined first guess. This makes it difficult to coax the optimizer into the right direction. The generation of starting configurations for non-symmetric designs can be dealt with in a number of ways, using e.g. construction methods and aberration theories, like simultaneous multiple surfaces[4], construction-iteration[5], and nodal aberration theory[6].

Thus, the benefits of using freeforms in optical design come with difficulties to determine appropriate starting points and suitable freeform representations, encountering optimization pitfalls, and dealing with more complex tolerancing analyses and alignment procedures.

We present a novel design tool for optimization of designs containing freeforms, that mitigates the difficulties related to optimization and mathematical surface representations, and combines starting configuration generation and tolerancing schemes in a single step with the final optimization. The tool makes use of a state-of-the-art global optimization strategy and using commercial optical design software as ray tracing engine.

## 2. DESIGN PRINCIPLES AND IMPLEMENTATION

In this section we will describe the implementation of the freeform optics optimization (FFOO) tool. The basic building blocks are an optimizer, and a merit function that drives a ray tracing engine. A Python-based interface is used to control the design steps and the optimization procedure.

### 2.1 Optimizer

The core of the design tool is the merit function optimizer. We consider a state-of-the-art stochastic global optimizer based on evolutionary strategies: Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES). An evolutionary strategy is an optimization technique based on the ideas of evolution. The optimizer starts with a population consisting of random possible optical designs, and converges to the best optical design (as indicated by the merit function) by introducing “mutations” in the population. CMA-ES is an optimization algorithm invented at INRIA[7,8] . It is currently considered as one of the best non-linear optimization algorithms available[9]. It has several characteristics that makes it promising for free-form optics design. CMA-ES does not evaluate the gradient of the merit function. This gradient evaluation is responsible for instabilities during optimization of commercial optical design software such as Zemax and CodeV and might be a reason for the difficulty to optimize free-form optics systems[2]. The algorithm is parameter-free: it does not need tuning of the algorithms parameter to reach good results. Being stochastic and evolutionary, it makes no fundamental distinction between local and global optimizations, and is capable of optimizing non-continuous functions. It is invariant for linear transformations of the search space and therefore is expected to be less sensitive to a particular choice of the freeform surface representation. It has the characteristic of a robust optimizer. Robust optimization allows to take into account tolerances (from manufacturing, alignment and integration processes) directly as input for the optimization activities. This capability will drastically reduce the needed iterations between the two “decision” steps of the optical design scheme flowchart (see Figure 1). All these characteristics make CMA-ES a very promising algorithm for freeform design.

### 2.2 Merit function

CODE V is used to compute the error function that represents a figure of merit for the performance of the optical design. The optimization capabilities of CODE V are not applied in the design tool. CODE V is predominantly used as ray tracing engine. For the design problem presented here, only geometric ray tracing is needed and no advanced features like beam synthesis are used. However, in principle any feature of CODE V needed to compute the merit function can be used.

One other advantage of evolutionary strategies worth mentioning in relation to the merit function is the following: CMA-ES is capable of jumping over, ignoring, optical configurations that are not ray traceable (the result is simply tagged as a solution that cannot be evaluated). By using it in this way the method is not dependent on continuously linked, ray-traceable systems.

### 2.3 Interface

To create the initial “blank” template lens the standard CODE V user interface or scripting capabilities are used.

In principle the CMA-ES algorithm could be ported to the CODE V scripting language. However, we chose to make use of an existing Python implementation of the CMA-ES algorithm. The interface between CODE V and CMA-ES has been realized in a Python main script via the Windows COM interface that allows inter-process communication as illustrated in Figure 3. Thus, CMA-ES controls the parameters of individuals in the population (corresponding to the parameters of the optical design, like surface curvatures, thicknesses, freeform parameters, et cetera), the main script feeds the parameters into CODE V, CODE V returns a merit function value back to Python which is in turn fed into the CMA-ES algorithm.

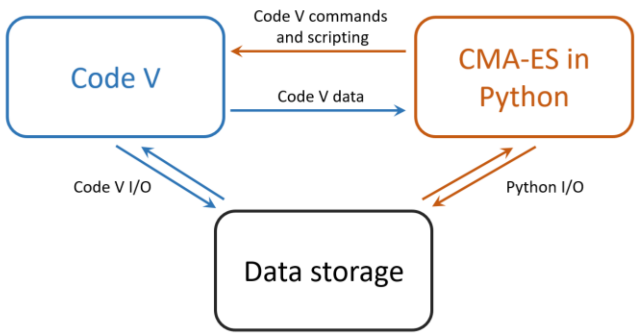


Figure 2. Schematic of the interface between CODE V and CMA-ES

Any graphical output of the optical design (view of the lens design, spot diagrams, et cetera) is taken care of by CODE V.

Like most population-based algorithms, CMA-ES can be easily run on parallel architectures, using coarse grain parallelism of the computation of the merit function for all points of the population. Thus, per iteration, the population is distributed over different processor cores. The gain here is almost linear, up to the number of processors cores, only limited by the communication time for sending the points coordinates (parameters of the optical designs together with the corresponding merit function values) to the CMA-ES algorithm and vice versa using the Python interface.

# 3. WORK FLOW

## 3.1 Preparing the optical design

The optical design is prepared in CODE V. That is the most convenient way to define for example the number and type of surfaces and a variety of other system settings such as apertures, dimensions, wavelengths, fields and more. Similar to the CODE V internal optimization approach, the parameters that should be included into the optimization by the freeform optics design tool are set as variables within CODE V. In addition, external parameters can be included that can be applied e.g. to control a starting configuration algorithm. The parameter values will later be altered by the CMA-ES algorithm in order to optimize the merit function.

## 3.2 Preparing the merit function

The error function contains all the information about the desired performance of the optical system. Typical examples are ray aberrations, wavefront error, spot size and MTF. However, any other performance metric that can be evaluated within CODE V can be added to the error function. It furthermore contains all required constraints apart from the upper and lower boundaries of the variables. Those are directly handled by the CMA-ES algorithm.

Using the CODE V error function infrastructure (by running the CODE V optimizer with zero iterations) is a very powerful way of implementing the merit function Utilizing the error function infrastructure makes it much easier to include weighted and inequality constraints, although it is possible to program one’s own error function. It is advised that when possible, the CODE V error function should be used.

Constraints related to volume, obscuration, clearance, et cetera, can be included by penalty terms in the merit function when they are soft constraints. Hard constraints are typically implemented by returning a merit function value that flags the system as invalid. The CMA-ES algorithm then ignores the optical configurations that are not valid.

# 4. ASSESSMENT OF THE DESIGN TOOL

For the assessment of the design tool the development of an imaging spectrometer was defined as a test case. The spectrometer can be used for hyperspectral imaging in a push broom remote sensing instrument.

### 4.1 Test case

The specifications of the imaging spectrometer are as follows:

Table 1. Specifications of the spectrometer

| | |
|---|---|
| Entrance slit length | 60 mm |
| Magnification | 1/3 |
| Entrance NA | 0.1 |
| Operating wavelength | 300 – 500 nm |
| Pixel pitch | 25 µm |
| Dispersion | > 1 nm / 100 µm |
| MTF @ Nyquist | > 0.3 |
| Smile | < 0.1 pixel |
| Keystone | < 0.1 pixel |

For the present test case of applying the design tool, we approach the problem of designing a spectrometer from basic principles, expressly with little expert knowledge. An imaging spectrometer is typically divided into three functional parts: a collimator that shapes the optical beam before the diffractive grating, the grating as a dispersion element and an imager that produces an image of both the spatial and spectral field of view. A concept design is shown in Figure 4. The entrance slit length of 60 mm can be used straight away and multiplying it with the spatial magnification of m = 1/3 results in an image size in the spatial direction of 20 mm. The slit width is determined by the spectral oversampling on the detector and the magnification in the spectral dimension. Selecting 4 pixels for spectral oversampling (4 · 25 µm = 0.1 mm) and assuming an identical magnification in spatial and spectral dimension of m = 1/3 yields a slit width of 0.3 mm. For this test case we assume that the system is telecentric at the entrance slit. The spectral range from 300 nm to 500 nm is best implemented by a fully reflective design due to better transmission and the absence of chromatic aberrations when using mirrors. A field flattener can later be used in front of the detector to improve the optical performance.

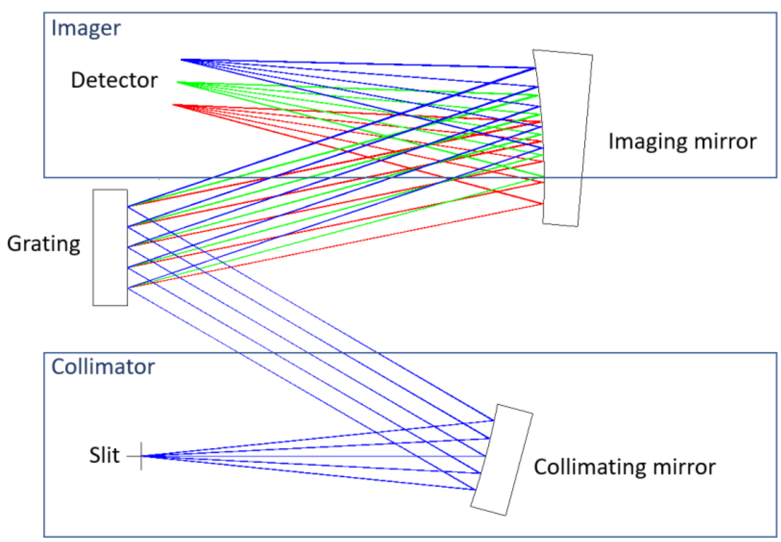


Figure 3. Concept design of an imaging spectrometer

### 4.2 Starting configuration

The following free parameters define the paraxial design: the focal length of the collimator which fully determines the spatial dimension. The focal length and NA of the imager are then determined by the magnification; the grating constant that determines the spectral dispersion and therefore influences the FOV in the spectral dimension for the imager; the angle of incidence on the grating, that together with the grating constant determines the focal length of the imager in spectral direction.

The focal length of the collimator (which will be the distance from the slit to the first mirror) was varied between 150 and 500 mm and chosen at 250 mm. It is the result of a tradeoff between scale/size of the optical system and image quality and results in a FOV of about 13.8 degrees in the spatial direction.

The grating constant and angle of incident were chosen to be 1100 lines/mm and 29 degrees in order to reach the required spectral resolution of better than 10 nm/mm. This is the result of a survey in the available free parameters to fulfill the requirements and clearance limits as well as minimize the FOV to limit aberration effects.

For this test case we chose one mirror for the collimator (M1) and three mirrors for the imager (M2, M3, M4) because aberration correction in the imager is usually considerably more complex than for the collimator due to the difference in f-number of a factor of 3. That provides ample degrees of freedom for the design of the imager and limits the total number of mirrors to four. The mirrors are positioned in a Z configuration (no crossed beams). Starting powers of the mirrors are chosen to comply to the paraxial requirements.

### 4.3 Degrees of freedom and merit function

All four freeform mirrors are represented with XY polynomial surfaces. In general, this is not the best choice surface representation because of the non-orthogonality of the polynomials which makes life difficult for general optimizers. Nevertheless this representation is selected here for ease of use, and from the point of view of the assessment it can serve as a demonstration that the evolutionary optimizer can deal with non-orthogonal polynomials. The surface shapes are defined using fourteen polynomial terms (no base curvature was used; anamorphic power of the mirrors is implemented via $x^2$ and $y^2$ terms. Together with decenter and tilt in the yz plane, distance to the next mirror, and orientation of the imager and collimator with respect to the grating, the number of degrees of freedom in this system amounts to 70.

The merit function is based on the Code V internal transverse ray aberration error function and contains additional weighted constraints:

- The maximum smile distortion of the sampled wavelength range
- The keystone distortion between the wavelengths
- Size constraints (distances between M2, M4 and detector to slit)
- Size of the spatial and spectral image on the detector to implement focal length constraints
- Constraint on collimation (reference ray angles)
- Constraint on the slit image size

The merit function value is extracted by reading the error function value from the Code V database error function value (we emphasize again that CODE V is used for ray tracing, but not for optimization).

Finally, a clearance function sets the boundary for the minimum clearance between optical elements (Slit, M1, Grating, M2, M3, M4 and Detector) and the neighboring beam. The output of that function is binary, either 0 for a compliant system or -1 for violation of the clearance, which triggers CMA-ES to ignore this sample in the population.

### 4.4 Results

Despite the large number of degrees of freedom, the design tool manages to come up with a design that is close to compliant with the requirements, without putting any expert knowledge into the starting configuration. This is a very promising outcome, as normally (i.e. for standard design tools) when many degrees of freedom are concerned and bad starting configuration, optimizations will go haywire. Indeed, in comparison, the standard CODE V optimizer was also used. It finds a solution only when gradually including more freeform degrees of freedom, otherwise CODE V is not converging on feasible solutions. By that time, it is very difficult to get to the more promising parts of the merit function landscape.

Table 1. Performance values of the nominal design obtained by the CMA-ES algorithm compared with standard optimizer

| **Performance metric** | **FFOO** | **CODE V** | **Requirement** |
|---|---|---|---|
| Minimum MTF at 20 lines/mm | 0.66 | 0.05 | 0.3 |
| Maximum Smile error [µm] | 4.4 | 220 | 2.5 |
| Maximum Keystone error [µm] | 1.9 | 50 | 2.5 |

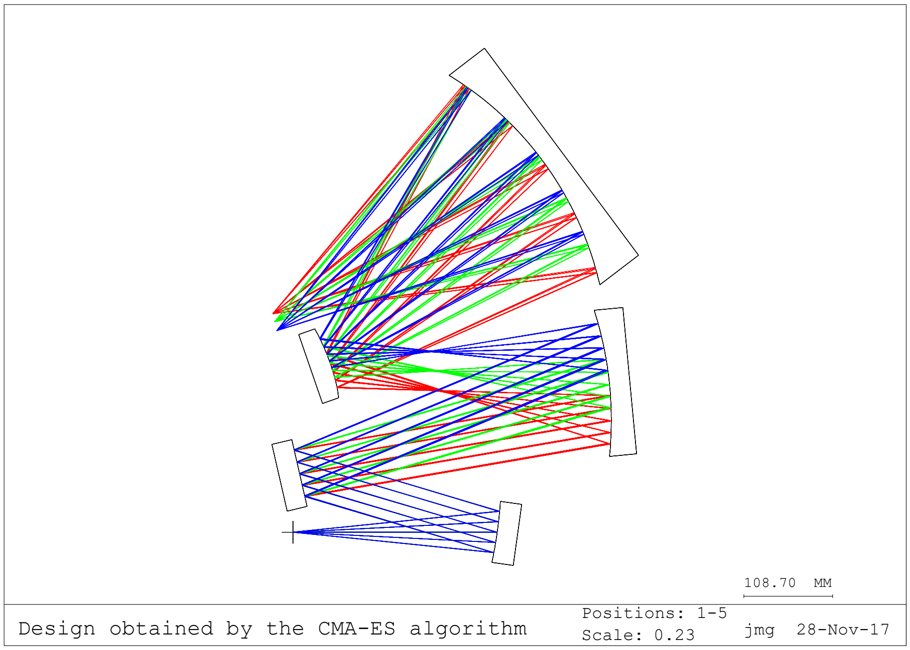


Figure 4. Design obtained directly by the CMA-ES algorithm with 70 degrees of freedom

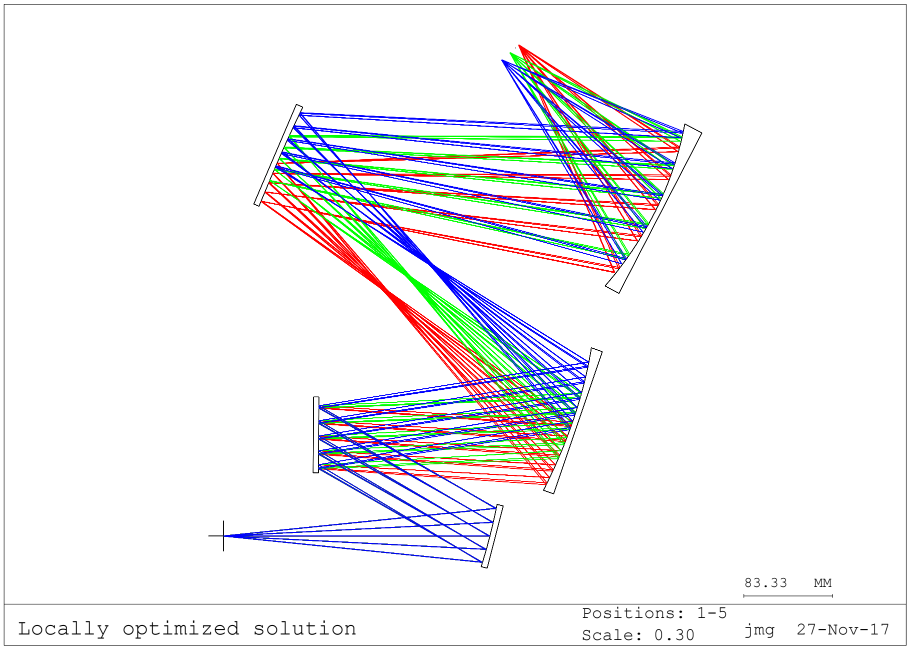


Figure 5. Design obtained by a local optimization procedure

The global optimizer of CODE V ("global synthesis", GS) is only capable of finding solutions when starting from an optimized system. It is important to note however that no exhaustive study of the GS option of CODE V was performed. This test case was more aimed at validating the new design tool than performing a complete quantitative assessment between the FFOO tool and CODE V. The results obtained with the design tool give confidence that FFOO is a very useful tool for designing freeforms optics with many degrees of freedom.

### 4.5 Tolerances

Owing to the robustness of the optimizer, solutions obtained with the new design tool are already less sensitive to tolerances than other solutions. As a final verification, tolerances can be incorporated in the design process by including the performance degradation into the merit function. This is easily done with the CODE V internal error function by using the SAB command in the merit function computation, or using the macro functions AS_BUILT_ABC and AS_BUILT_VAR when using one's own error function. It has been checked that the design is relatively tolerance insensitive, and the design optimized excluding tolerances closely resembles the design optimized including tolerances.

## 5. CONCLUSIONS

In this publication, we described the development of a novel design tool for freeform optics and application to the design of an imaging spectrometer. It was shown to be capable of yielding a good optical design solution, even when optimizing bad starting configurations, with many degrees of freedom. The design tool can conveniently be used in combination with the CODE V error function infrastructure. FFOO is a powerful tool to overcome the complexities accompanying freeform optics design. The importance of surface representations, starting configurations, careful managing of the degrees of freedom are considerably mitigated.

## ACKNOWLEDGEMENT

The authors wish to thank Marc Schoenauer, Nikolaus Hansen and Dejan Tušar (INRIA) for their support and helpful discussions in developing the freeform optics optimization tool.

## REFERENCES

[1] Thompson, K.P. and Rolland, J.P. "Freeform optical surfaces: A revolution in imaging optical design," Opt. Photon. News 23(6), 30 (2012).

[2] Van Turnout, M., Bociort, F. "Chaotic behavior in an algorithm to escape from poor local minima in lens design", Optics Express, 17(8), p. 6436 (2009).

[3] Brömel, A., Gross, H., Ochse, D., Lippmann, U., Ma, C., Zhong, Y., Oleszko, M.. , "Performance comparison of polynomial representations for optimizing optical freeform systems", SPIE 9626, 0W-1 (2015).

[4] Lin, W., Benítez, P., Miñano, J.C., Infante, J.M., Biot, G., and de la Fuente, M. "Sms-based optimization strategy for ultra-compact swir telephoto lens design", Optics express, 20(9) p. 9726–9735 (2012).

[5] Yang, T., Zhu, J., Wu, X., and Jin, G. "Direct design of free-form surfaces and free-form imaging systems with a point-by-point three-dimensional construction-iteration method", Optics Express, 23(8), p. 10233–10246 (2015).

[6] Thompson, K.P. "Description of the third-order optical aberrations of near-circular pupil optical systems without symmetry", J. Opt. Soc. Am. A. 22(7), p. 1389–1401 (2005).

[7] Ostermeier, A., Gawelczyk, A., and Hansen, N. "A derandomized approach to self-adaptation of evolution strategies", Evolutionary Computation, 2(4), p. 369–380 (1994).

[8] Hansen, N. and Ostermeier, A. "Completely derandomized self-adaptation in evolution strategies", Evolutionary Computation, 9(2), p. 159–195 (2001).

[9] Hansen, N., Auger, A., Ros, R., Finck, S., and Posik, P. "Comparing Results of 31 Algorithms from the Black-Box Optimization Benchmarking" BBOB-2009. In Proc. Workshops ACM-GECCO, Portland, United States, p 1689-1696 (2010).