BeamTuner Optimization Program, version 1.0.6

User guide

July 9, 2014

1 Introduction

The BeamTuner Windows application implements a number of optimization algorithms for adaptive compensation of static and quasi-static aberrations in optical systems. BeamTuner does not need a wavefront sensor, instead it uses a feedback based on some quality metric of the image, registered by a computer-controlled camera. The software implements a range of well-known multidimensional algorithms including Nelder-Mead simplex optimization, parallel perturbation gradient descent, simulated annealing, coordinate descent, as well as some of variants, such as random walker and irregular simplex optimization. A wide choice of performance metrics, based on the beam size, image brightness, beam shape, image sharpness, etc., suitable for different experimental situations, is provided by the software.

The program is designed to work together with adaptive mirrors produced by Flexible Optical B.V., however deformable mirrors from other manufacturers also can be used. The deformable mirrors are driven by digital-to-analog converter (DAC) followed by high voltage amplifier with appropriate number of control channels. Two models of DAC unit, both from OKO Tech, are currently supported: DAC40USB (40-channel, 12-bit, USB interface) [2] and EDAC40 (40-channel, 16-bit, Ethernet interface) [3]. For detailed hardware description please refer to the corresponding DAC manual. Camera interface is organized via so called "video plug-ins". These plug-ins are compatible with FrontSurfer software [4]. Plug-ins for several popular scientific cameras are included with the software, modules for other cameras can be programmed according to the published interface description.

A "new style" video interface, implemented specially for BeamTuner has several advantages, including the possibility of program control of the area-of-interest and exposure in the process of optimization. Currently only IDS cameras are supported in this new framework.

2 Installation

The installation package can be found in the "BeamTuner" directory on the supplied CD. In some cases it comes pre-installed on a laptop computer as well. Please run the installer by double-clicking its icon, then follow the instructions. Some third-party software components listed below should be installed prior to the running of the application. The distribution packages for these components can be found on the CD or downloaded from the manufacturers' websites.

1. Driver for the camera (e.g. directory `/drivers/ueye/` for IDS uEye camera series). The most recent drivers are available from IDS Imaging Development Systems GmbH web site [6].

2. FTDI driver for DAC40USB (connect the device to the computer with USB cable, wait till Windows asks for the driver, direct it to `/drivers/ftdi/` directory). Please check for updated drivers on the website of Future Technologies Devices International Ltd.[5]

3. Driver for KEYLOK piracy prevention hardware key (please install it *before* connecting the key to the computer from the directory `/drivers/keylok2/`). The manufacturer's website has the most recent drivers [7].

3 Typical optical setup

Typical system for testing the functionality of the system is shown in Fig. 1. Output of a single-mode fiber, coupled to a laser diode, serves as the points source which is collimated to a parallel beam by the lens L1. After reflecting from the deformable mirror (DM), the beam is focused on the CMOS sensor of the imaging camera (IC) by the lens L2. The MMDM mirror is driven by a DAC connected to the high voltage amplifier (HVA). The DM compensates the system aberrations and the aberrations introduced with the "aberration simulator" (ABS). A circular plexiglass plate mounted on the axis of geared DC motor can be used as a simple ABS. Fig. 2 illustrates the results of optimization of the beam diameter in the lens focus.

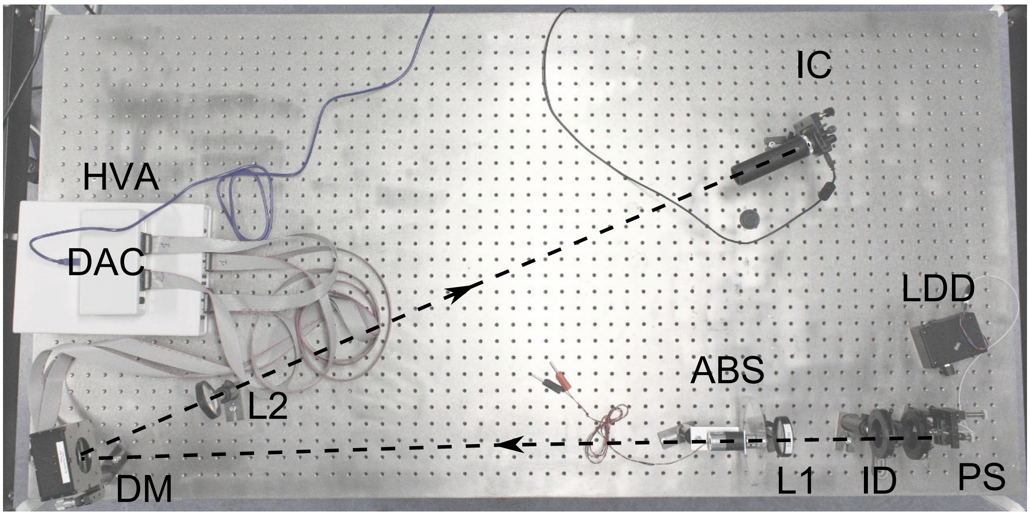

Figure 1: Testing optical setup. Main components: point source (PS), iris diaphragm (IS), collimating lens (L1, f = 200mm), aberration simulator (ABS, optional), deformable mirror (DM), focusing lens (L2, f = 1m), imaging camera (IC), laser diode driver (LDD), digital-to-analog converter unit (DAC), high voltage amplifier (HVA). The long black tube attached to the front of the camera serves as a hood protecting the sensor from ambient light.

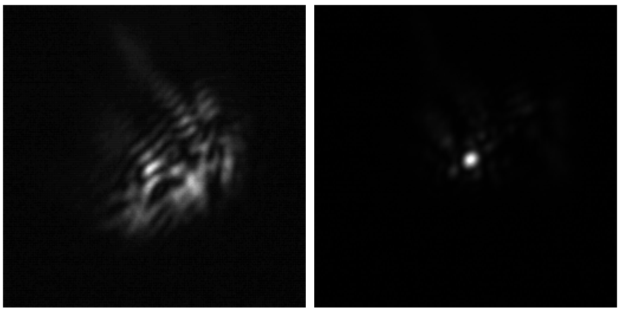

Figure 2: Spot image before (left) and after (right) optimization. Before the experiment the mirror was set to zero state (middle of its operational range, corresponding to some offset), then the beam was focused on the sensor surface by manual adjusting of the camera position. After that aberration simulator was introduced, then optimization started. Intensity is scaled, otherwise defocused light is barely visible.

4 Operating software

The GUI is comprised of the Main application window (Fig. 3) which allows the user to control optimization process and the "Spot View" window (Fig. 4) representing the image used for the feedback. The parameters are split into several groups, controlling different aspects of the system (see table 1 for summary). Most parameters are self-explained. Depending on the settings and the state of the system, the controls that are irrelevant to the situation, are automatically disabled.

The spot view window (Fig 4) shows the image captured from the camera in real time. It can be used for the adjustments of the optical setup, and for monitoring of the optimization process. The spot position (green hair cross), spot size (blue circle) and the intensity cross-section (blue curves at the window boundaries) through the beam center along two axes, are shown in the window. The behaviour of those marking depends on the state of "Follow spot" check box: center marker either automatically moved to the spot center, or can be positioned manually. It is possible to select the area-of-interest (indicated with red dashed line), the desired spot position (small red cross) and the position for the cross-section with the mouse (see table 2 for a list of mouse actions). Parts of the image with intensity values above threshold could be shown with semi-transparent color overlay by checking "Visualize threshold" check box. Some additional information, such as maximum intensity, spot width etc could be shown in the same window during optimization.

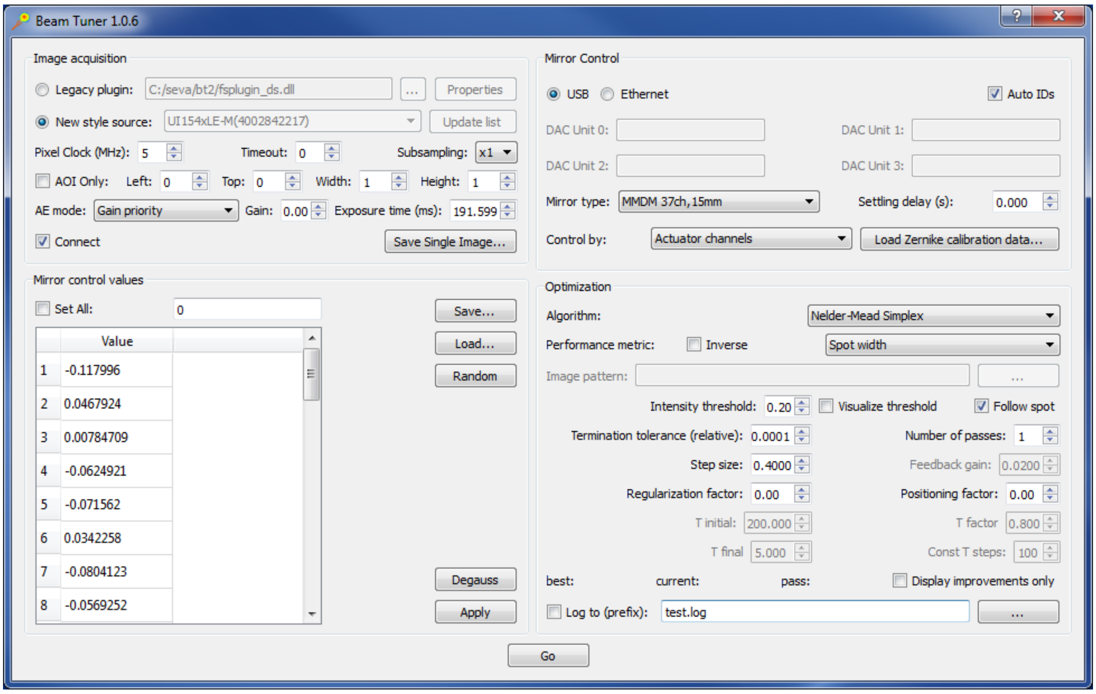

Figure 3: BeamTuner main window

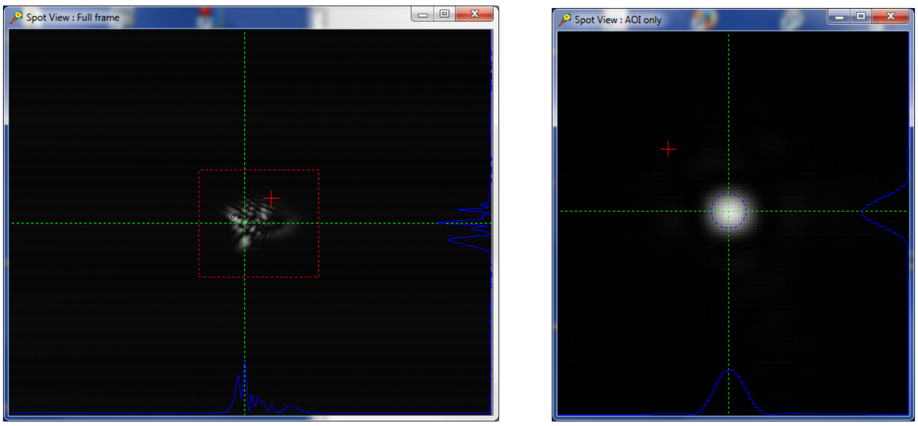

Figure 4: Spot view window in full-frame (*left*) and area-of-interest only (*right*) modes. The states before and after optimization are shown.

Table 1: Program controls available in the main application window

Image acquisition group:

Legacy plug-in	Use legacy FrontSurfer plugins. The name of currently selected video plug-in is shown and can be changed with "..." (Browse) button. Those plug-ins are .dll library files, for instance "ueye.dll" for IDS uEye series cameras or "fsplugin_ds.dll" for DirectShow cameras including most web-cameras and analog frame grabbers. See section 3.2.3 of FrontSurfer manual [4] for specification. It is recommended to use "New style source" whenever possible.
Properties	Invokes properties window for selected FrontSurfer plug-in. Available parameters, and appearance of this dialog determined entirely by the plug-in.
New style source	Select new style source for images. Particular camera can be selected from drop-down list of cameras detected in the system
Update list	Update list of available cameras manually. Automatic updates occurs every few seconds.
Pixel clock	Camera pixel clock in MHz. Needs to be lowered on the lower performance systems or when several USB devices especially cameras share the same bus (30MHz and below is usually a safe choice). Depends on camera model. Determines available range of exposure times.
Timeout	During this period of time (in seconds) response from the camera should be received before it was reported as an error condition. Specify 0 to use internal default.
AIO only	Only selected part of the image (Area-of-interest) is shown in the "Spot View" window and analyzed by the software. Size of the area affects performance drastically. It is strongly recommended to keep the AOI reasonably small.
Left,Top,Width,Height	Numeric values in pixels, specifying area-of-interest rectangle.
AE mode	Auto exposure mode. Possible values are *Manual*, *Gain priority* and *Exposure time priority*. In the manual mode both exposure (gain and exposure time) parameters can be set independently. In two other modes one parameter is set manually by user while other is dynamically adjusted be the software to keep intensity maximum withing the range of the sensor. In this way maximum dynamic range of the sensor can be achieved. For most applications it is recommended to keep gain at minimum (0) and use *Gain priority* mode.
Gain	Defines sensitivity of the camera by controlling of amplification gain. Possible range is $0 \dots 1$. Noise level drastically increases for higher values.
Exposure time	Exposure time in ms. Depends on camera model and current *Pixel clock* settings.

Connect	The checkbox allows connect-disconnect-reconnect camera (for example, in the case of changing of the device or some hardware problem). At the start the program tries to connect to the first available camera.
Save single image	Allows to save last captured image to .PGM file. Upon pressing the button file selecting dialog appears.
Mirror Control group:	
USB / Ethernet	Type of the modules used for controlling of the mirror.
DAC Unit 0. . .4	Depending on the number of actuators, one to four 40-channel DAC modules are used. Each module has a fixed unique hardware identifier (it coincides with MAC address in the case of modules with Ethernet interface) labeled on the bottom of unit. The modules connected to the mirror could be specified either manually by their IDs (up to three units in corresponding text fields) or automatically
Auto IDs	if the box is checked, modules are selected automatically, in ascending order of their IDs. The user should take care of correct connection between the modules and HV amplifier.
Mirror type	All standard deformable mirrors produced by Flexible Optical are supported and can be selected form drop-down list. Parameters, specific for particular type, such as channel mapping and response type are adjusted automatically.
Settling delay	Membrane (MMDM) mirrors react to changing of control signal almost instantly, others slower devices might require some delay between sending data to DAC module and measuring response from image. This period of time (in seconds) can be adjusted with this parameter.
Control by	Either signals of individual deformable mirror actuators or Zernike modes could be used as control variables.
Load Zernike calibration data	There is built-in calibration data for generic samples of several popular mirror models, calibration data for other cases can be loaded with “this button. Supports `.dmc` (XML) format of MiZer software.
Mirror control values group:	
Set all	When checked, given value will be applied for all control variables (actuator channels or Zernike modes).
Values table	Values for individual control signals. Updated by the program during optimization or can be specified manually when optimization process is not running. Normalized $0 \ldots 1$ values in the case of control channels.
Save...	Saves current set of control variables into a text file.

Load...	Loads values of control variables from a text file
Random	Generates pseudo-random set of signals.
Degauss	Since piezo-actuator based mirrors (PDM) demonstrate appreciable amount of hysteresis, it is advisable to subject those devices to signal cycling around central point with decreasing amplitude. The process can be initiated by this button.
Apply	Sends control values given in the table (or “Set all” box) to the DAC.

Optimization group:

Algorithm	Selects the algorithm of optimization. Available algorithms are: Nelder-Mead Simplex, Irregular simplex, Parallel perturbation gradient descent, Random walker, Simulated annealing, Coordinate descent. See section 5 for more information.
Performance metric	Chooses the function to minimize. Available objective functions are: spot width, number of pixels above threshold, integral intensity, sharpness, edge sharpness, image pattern.
Inverse	Changes the sign of performance metric, essentially switching between minimum and maximum to search. C_{inv} in equation (1)
Image pattern	Name of .PGM file containing data for optimization with “image pattern” performance metric. Browse button (...) brings file selection dialog.
Intensity threshold	Cutoff level (in parts of maximum intensity) used for the calculation of the spot size with *spot width* and *number of pixels above threshold* objective metrics. Exact value is determined by optical configuration, but typically should be within 0.05...0.2 range.
Termination tolerance	Relative difference between the best and the current objective function values used as a condition for termination of current optimization pass. Often is set to zero, to control the termination manually.
Number of passes	Number of optimization passes. Sometimes it makes sense to perform the optimization several times, starting from already found best state.
Step size	Has slightly different meaning for different algorithms. For simplex it defines initial size of the simplex, for irregular simplex in defines maximum range of randomly choosed simplex vertexes coordinates. For random walker and simulated anneling it defines maximum length of step for each iteration (steps themselves are random). For two other algorithms it is just unit step, applied at each iteration.

Feedback gain	Gain coefficient used with parallel perturbation gradient descent algorithm only. For detail see algorithm description in [10].
Regularization factor	Additional term added to objective function to facilitate more stable behaviour and ability to locate global minimum. This parameter represents weight of the regularization part C_{reg} in equation(1). Typical suggested values are in $0 \ldots 10$ range, depending on configuration and selected metric.
Positioning factor	It is possible to direct center of the beam to some particlular position (see Table 2) through additional term in the objective function. This parameter represents C_{pos} in equation (1). Typical suggested values are in $0 \ldots 10$ range, depending on configuration and selected metric.
T initial, Tfinal, T factor, Const T steps	Parameters related to simulation annealing algorithm (initial temperature, final temperature, coefficient of temperature decrease, number of steps at constant temperature). Reasonable defaults are $T_{\text{initial}} = 200$, $T_{\text{final}} = 5$, $T_{\text{factor}} = 0.8$, const T steps = 100
best, current, pass	Values of objective function and current pass number.
Log to	Checking the box allows the user to record a sequence of images and objective functions into files with specified prefix (select with *Browse* button. Please note that is slightly degrades temporal performance of the software.
Visualize threshold	Checking the box switches on additional color overlay layer in the spot view window, which indicates parts of the image with pixel intensity higher than threshold.
Follow spot	When the box is checked, "Center cross-hair" in the spot view window automatically moved to the current spot position.
Display improvements only	If this box is checked during optimization, the image in spot view window is updated only for iterations which lead to quality metrics improvement.

Table 2: Spot view winodw: actions performed with the mouse

Left	drag	Select an area of interest. Visualized as a dashed red borderline. Switching between selected AOI and full frame is done with *AOI* checkbox.
Right	click	Select center position for intensity cross-section. Shown as a green hair-cross. During optimization the position is updated automatically after each iteration. Intensity plots are shown along margings of the window.
Right+Ctrl	click	Desired beam position. Indicated with small red cross. Distance between specified beam position (x_p, y_p) and actual calculated position multiplied to *positioning factor* and added to objective function (equation (2))

5 Performance metrics and optimization algorithms

Objective function to be minimized is defined as:

$$f(\mathbf{v}) = C_{\text{inv}} \cdot f_{\text{metric}}\Big(I(\mathbf{v})\Big) + C_{\text{reg}} \cdot f_{\text{reg}}(\mathbf{v}) + C_{\text{pos}} \cdot f_{\text{pos}}\Big(I(\mathbf{v})\Big), \tag{1}$$

where $\mathbf{v}$ is a vector of control values, $I(\mathbf{v})$ is resulting intensity distribution, first term describes performance metric of selected type with $C_{\text{inv}} = 1$ or -1 depending on *Inverse* parameter, second and third terms are responsible for regularization and positioning with corresponding factors C_{reg} (regularization) and C_{pos} (positioning factor):

$$f_{\text{reg}} = \sqrt{\sum_i v_i^2}, \qquad f_{\text{pos}} = \sqrt{(x_0 - x_p)^2 + (y_0 - y_p)^2} \tag{2}$$

Centroid spot position (x_0, y_0) and half-width r are calculated as:

$$x_0 = \frac{\sum\limits_{i,j} i \cdot I_{ij}}{\sum\limits_{i,j} I_{ij}}, \qquad y_0 = \frac{\sum\limits_{i,j} j \cdot I_{ij}}{\sum\limits_{i,j} I_{ij}}, \qquad r = \sqrt{\frac{\sum\limits_{i,j} I_{ij} \cdot \Big((i - x_0)^2 + (j - y_0)^2\Big)}{\sum\limits_{ij} I_{ij}}}, \tag{3}$$

where I_{ij} is intensity of the pixel (i, i), summation is done over all pixels with values higher than cutoff level I_{thr}. Spot half-width can be chosen as one possible metric (most robust and widely used), others are summarized in the table 3

Depending on the application, the most appropriate optimization algorithm can me chosen. Their main features are briefly summarized below. Optimization process is controlled by a number of parameters. As their values greatly depends on the optical setup and the nature of the problem and the optimization algorithm, the user is encouraged to experiment. Table 4 should give a reasonable starting point. Also refer to the table 1 for the explanation and the recommended values of parameters. Algorithms are well documented in textbooks and special papers and interested user is encouraged to check cited sources and references there in.

Simplex multidimentional optimization algorithm is simple, robust and demonstrates reasonably fast convergence. Present set of "control values" is taken as the first vertex for simplex constructing, which has $N+1$ vertexes for a mirror with N actuators (N-dimensional optimization). Other vertexes are separated from the first by the length specified as "Step size" parameter along of each dimension. Value of objective function is evaluated for each vertex and "worst" vertex is moved at each iteration according by application of some rules of transformation (reflexion, expansion, contraction are among those operations). The code is based on the algorithm proposed by Nelder and Mead as described in [9]. It often makes sense to restart optimization process several times with found best value as a starting point.

Irregular simplex method is a variant of the previous algorithm and has additional enhancement of purely arbitrary choice of vertexes (except the first), thus, after several passes it has a better chance to locate the global extremum.

The parallel perturbation gradient descent method is implemented as described in [10]. Basically, small perturbations δv_i of equal amplitude (determined by *Step size* parameter) but random sign are applied to every control signals in parallel at each iteration. Resulting perturbation δf of objective function is measured from system response and new set of control parameters at iteration $n+1$ is formed from state at previous iteration n as $v_i^{(n+1)} = v_i^{(n)} - \gamma \delta f / \delta v_i$. Weight coefficient γ has a meaning of *feedback gain*. The method is particularly effective for systems with high number of control parameters. It has ability to track changes in the system and thus suitable for correction of dynamic aberrations.

The random walk is a very simple empirical algorithm which at each step gives random increment to each control variable (maximum step size is limited to the value of corresponding parameter). The new state is accepted if objective function value is lower than previous. Despite its simplicity, method works surprisingly well for dynamic applications.

The simulated annealing algorithm [9] uses analogy with thermodynamics, specifically with the way matter crystallizes. Like random walk algorithm it makes random steps, but under some circumstances it can step not only to the state with lower value of function (downhill), but to a state with higher value (uphill). Probability of such event determined by Boltzmann statistics and depends on differences in values as well as control parameter T (analog of temperature for thermodynamic process). Algoritms

Table 3: Performance metrics, available in the program

Spot width	Spot half-width as r as defined by equation 3
Number of pixels above threshold	Number of pixels those intensity $I_i > I_{\text{thr}}$. In effect very similar to spot width, but can be used in some cases when several bright spots are normally expected.
Integral intensity	Sum of pixel intensities $I = \sum_{i,j} I_{ij}$.
Sharpness	$S = \sum_{i,j} I_{ij}^2$. Different sharpness metrics are discussed in [8]
Edge sharpness	$S = \dfrac{\sum_{i,j} (I_{i+1,j} - I_{ij})^2 + (I_{i,j+1} - I_{ij})^2}{\sum_{i,j} I_{ij}}$. Proportional to local intensity differences between neighboring pixels, thus it has higher values for sharp ("in-focus") and lower for blured ("out-of-focus") images. To be used primary with imaging systems (after [8])
Image pattern	Intended for search a set of control values reproducing given intensity distribution. To be used together with image pattern ("mask") the metric is defined as difference between cumulative intensity "inside" (i.e. under bright part of the image) and cumulative intensity "outside" (under dark part): $$P = \frac{\sum_{i,j} M_{ij} I_{ij}}{\sum_{i,j} M_{ij}} - \frac{1}{2} \frac{\sum_{i,j} \left[\max(M) - M_{ij}\right] I_{ij}}{\sum_{i,j} \max(M) - M_{ij}},$$ where M_{ij} is an intensity distribution of the pattern. This feature is considered experimental and was not extensively tested.

starts with some high value of $T = T_{\text{initial}}$, performs a number of iterations (contsTsteps) at this value, then decreases the "temperature" by the factor of T_{factor}. Process stops upon achieving of "temperature" T_{final}. Algorithm is rather slow to converge (as it is allowed to travel "uphill"), but due to its stochastic nature, is supposed to be able to locate global extremum with slow "temperature" decrease and given enough time.

The coordinate descent is a classic optimization method (see e.g. [1]) which performs optimization along each axis sequentially and uses parabolic interpolation to locate minimum for each variable. It is fast and very well suited for control with Zenike modes.

Figure 5 gives sample evolution curves (performance metric plotted vs. iteration cycle number) for various optimization methods. Same settings (sometimes sub-optimal) were used for all methods. Performance metric for every iteration is plotted (not only those accepted as the best). This should give some ground for comparison and choice of algorithm for particular application.

Table 4: Recommended parameters values

Algorithm	Threshold	Step	Gain
Nelder-Mead simplex	0.2 . . . 0.05	0.5	N/A
Irregular simplex	0.05	0.5	N/A
Parallel perturbation gradient descent	0.05	0.02	0.01
Simulated annealing	0.2	0.02	N/A
Random walker	0.2	0.02	N/A
Coordinate descent	0.05	0.2	N/A

References

[1] B.D. Bunday. *Basic Optimisation Methods*. Edward Arnold, 1984.

[2] Flexible Optical B.V. DAC unit "dac-40-usb". http://www.okotech.com/images/pdfs/dac40usb_v2.0_readme.pdf, 2008.

[3] Flexible Optical B.V. EDAC40: Ethernet-connected 40-channel Digital-to-Analog converter module: User guide and service manual. http://www.okotech.com/images/pdfs/edac40_user_guide.pdf, December 2010.

[4] Flexible Optical B.V. FrontSurfer wavefront analysis and control system. http://www.okotech.com/images/pdfs/fs_manual_1.4.5.pdf, May 2011.

[5] Future Technologies Devices International Ltd. Drivers for USB converter. http://www.ftdichip.com/Drivers/D2XX.htm.

[6] IDS Imaging Development Systems GmbH. Drivers for IDS cameras. http://www.ueyesetup.com/.

[7] Microcomputer Applications Inc. KEYLOK piracy prevention hardware key drivers. http://www.keylok.com/support/downloads.aspx.

[8] Richard A. Muller and Andrew Buffington. Real-time correction of atmospherically degraded telescope images through image sharpening. *J. Opt. Soc. Am.*, 64(9):1200–1210, Sep 1974.

[9] William H. Press, Saul A. Teukolsky, William T. Vetterling, and Brian P. Flannery. *Numerical recipes in C (2nd ed.): the art of scientific computing*. Cambridge University Press, New York, NY, USA, 1992.

[10] M. A. Vorontsov, G. W. Carhart, and J. C. Ricklin. Adaptive phase-distortion correction based on parallel gradient-descent optimization. *Opt. Lett.*, 22(12):907–909, Jun 1997.

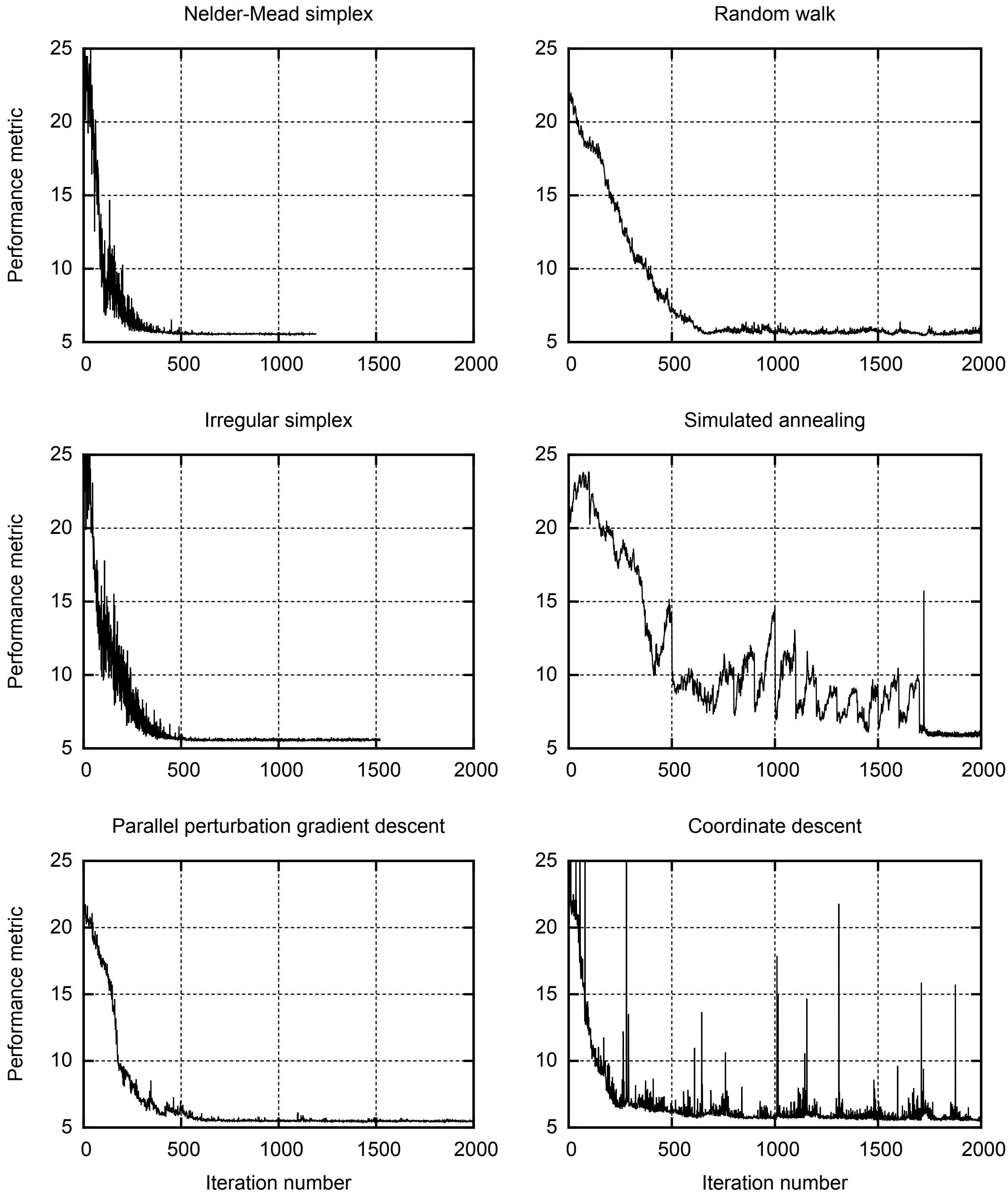

Figure 5: Sample evolution curves for different optimization algorithms. Same settings (sometimes sub-optimal) were used for all methods. Performance metric for every iteration is plotted (not only those accepted as the best). Plots are limited in number of iterations, some algorithms (simulated annealing, for instance) continue to converge.