Lens design optimization by back-propagation

Congli Wang, Ni Chen, and Wolfgang Heidrich
Visual Computing Center, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia
{*congli.wang, ni.chen, wolfgang.heidrich*}*@kaust.edu.sa*

Abstract: We implement a ray tracing engine using automatic differentiation, which enables accurate derivatives to be evaluated via back-propagation. Exploiting the gradients, multi-variable designs (e.g. freeforms) can be optimized by gradient descent.

1. Lens design optimization by back-propagation

In this work, we consider non diffraction-limited lens system design with sequential ray tracing. In this situation, geometric optics is the basic principle where sampled rays pass through a known sequence of parameterized optical surfaces before reaching to the image sensor. For such a parameterized design to evolve in a feasible parameter search space in the hope of reducing a given merit function, custom techniques (e.g., damped least squares [1]) are required to perform this optimization. Such process usually involves multiple gradient evaluations, that are obtained via finite difference approximation in most ray tracing engines.

Here, we propose to use back-propagation [2, 3] together with gradient descent as an alternative approach for general design optimization. Back-propagation, or known as the reverse-mode in automatic differentiation [4], is a widely used algorithm for computing gradients in training neural networks. Conceptually speaking, this technique constructs a computational graph where variables are connected according to computations. In back-propagation, the previously constructed computational graph will be looped reversely starting from a final node, and the target gradients are computed in an efficient yet numerically accurate way, without using approximations or repetitive primal evaluations (i.e., the forward ray tracing in our case) as the case in finite difference, and hence enabling a more efficient and accurate gradient evaluation for later usage.

Inspired by this computation spirit, we investigate using back-propagation for design optimization, especially for freeform designs where a large amount of parameters are being optimized, leveraging the advantages of back-propagation. To this purpose, we implement a ray tracing engine on top of automatic differentiation. A lens system is parameterized by a vector variable $\boldsymbol{\theta}$, which is a collection of total parameters for multiple optical surfaces, examples are surface curvatures, air spacing, freeform coefficients, and etc. By sampling rays originated from different angles, spot diagram $\mathbf{p}(\boldsymbol{\theta})$, i.e., intersections on the image sensor, are obtained as a function of $\boldsymbol{\theta}$.

To optimize or tolerancing a design, additional analysis needs to be performed, for example in spot diagram analysis a root-mean-square (RMS) spot error is obtained by taking $\mathbf{p}_i$ as inputs at field of view i, and outputs an error metric $\varepsilon(\boldsymbol{\theta}) \in \mathbb{R}$, given a deterministic error function $f(\cdot)$. This is known as the merit function:

$$\varepsilon(\boldsymbol{\theta}) = \sum_i f(\mathbf{p}_i(\boldsymbol{\theta})), \qquad \text{and its derivatives by the chain rule:} \quad \frac{\partial \varepsilon}{\partial \boldsymbol{\theta}} = \sum_i \frac{\partial \mathbf{p}_i}{\partial \boldsymbol{\theta}} \frac{\partial f}{\partial \mathbf{p}_i}. \tag{1}$$

Using automatic differentiation, for a given parameter value $\boldsymbol{\theta}$, both $\varepsilon(\boldsymbol{\theta})$ and $\partial\varepsilon/\partial\boldsymbol{\theta}$ in Eq. (1) can be obtained numerically, offering an intrinsic way for us to employ this information for purpose-specific applications, such as design optimization or local sensitivity analysis via the Jacobian matrices.

In design optimization, the goal is to find a set of optimal parameters $\boldsymbol{\theta}^*$ that minimizes $\varepsilon(\boldsymbol{\theta})$ in Eq. (1). Once $\partial\varepsilon/\partial\boldsymbol{\theta}$ is available, gradient descent can be employed for this purpose, given a learning rate α, and hence doing back-propagation to update the design parameters:

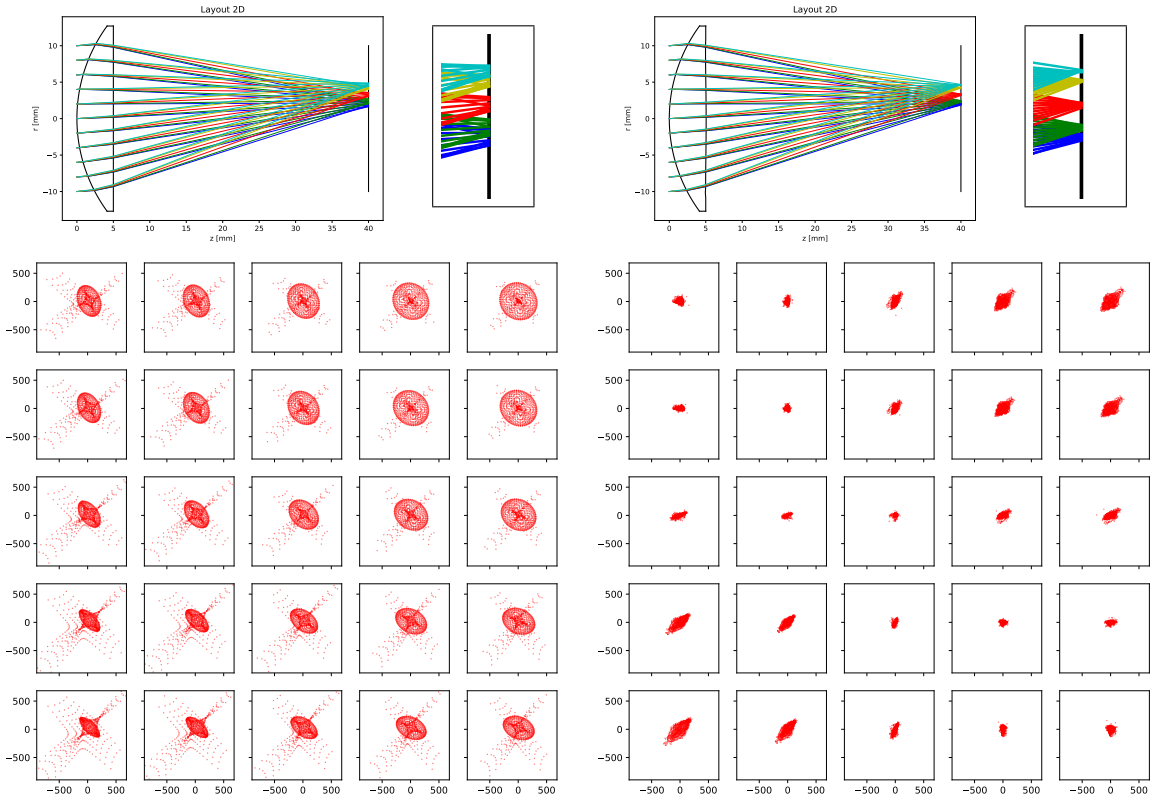
$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \varepsilon(\boldsymbol{\theta}), \qquad \text{iteratively solved by} \quad \boldsymbol{\theta}^{(k+1)} \leftarrow \boldsymbol{\theta}^{(k)} - \alpha \left.\frac{\partial \varepsilon}{\partial \boldsymbol{\theta}}\right|_{\boldsymbol{\theta}=\boldsymbol{\theta}^{(k)}}. \tag{2}$$

Alternatives of Eq. (2) may be utilized instead of simple gradient descent. We employed the Adam optimizer [5].

2. Example design

We consider the problem of optimizing a plano-convex singlet (entrance pupil diameter 20 mm, front surface curvature 0.0464 mm^{-1}) using the proposed ray tracing engine, by making its back surface as a cubic B-spline

freeform surface (51×51 coefficients), with the knot positions being a fixed grid distribution. Our task is to optimize the freeform surface to reduce the total aberration at a skew angle of $5° \pm 2°$. Here, the B-spline coefficients are the variables $\boldsymbol{\theta}$, and $\varepsilon(\boldsymbol{\theta})$ is a summation of all the RMS spot errors at 5×5 viewing angles ranging from 3° to 7°, at a wavelength of $\lambda = 587.6\,\text{nm}$. Figure 1 illustrates the initial and optimized setups with ray tracing, as well as the spot diagrams at different field of views. As seen, the optimized lens with the freeform surface suffer from smaller aberrations. Figure 2 visualizes the optimized freeform surface, showing a bi-conic shape wavefront, demonstrating the success of our optimization to minimize skew angle aberrations.



Initial: Mean RMS error 257.29 µm | Optimized: Mean RMS error 80.28 µm

Fig. 1: Design optimization by gradient descending $\boldsymbol{\theta}$. Spot diagram units are in µm.

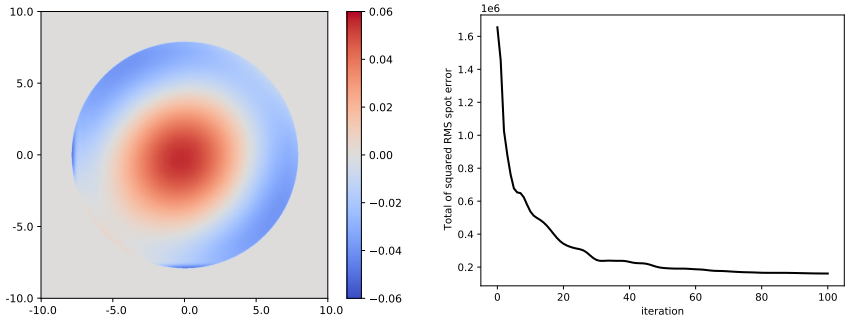


Fig. 2: Optimized freeform surface (units are in mm), and $\varepsilon(\boldsymbol{\theta})$ with respect to iteration k.

References

1. J. Meiron, "Damped least-squares method for automatic lens design," J. Opt. Soc. Am. **55**, 1105–1109 (1965).
2. Y. LeCun, D. Touresky, G. Hinton, and T. Sejnowski, "A theoretical framework for back-propagation," in *Proceedings of the 1988 connectionist models summer school,* , vol. 1 (1988), pp. 21–28.
3. R. Hecht-Nielsen, "Theory of the backpropagation neural network," in *Neural networks for perception,* (Elsevier, 1992), pp. 65–93.
4. A. G. Baydin, B. A. Pearlmutter, A. A. Radul, and J. M. Siskind, "Automatic differentiation in machine learning: A survey," J. Mach. Learn. Res. **18** (2018).
5. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations,* (2015).