# Automatic Lens Design based on Differentiable Ray-tracing

**Xinge Yang, Qiang Fu, Wolfgang Heidrich**

*Visual Computing Center, King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia*

*xinge.yang, qiang.fu, wolfgang.heidrich@kaust.ed u.sa*

**Abstract:** We propose a fully differentiable optical design method enabled by curriculum learning. Preliminary results show that our framework is suitable to solve highly non-convex problems like cellphone lens design. 

## 1. Methods

Traditional optical design methods require starting structures from known and successful forms. It's typically realized by experienced engineers with constant interventions during the optimization. Recent work in automatic staring point generation [1] and differentiable ray-tracing [2–5] show a promising potential to solve optical design problems as an optimization task with neural networks. But they either can only generate spheric lenses or need a well-designed starting point. In this paper, we optimize a multi-aspheric-lens system without any prior design by introducing curriculum learning.

The imaging process can be simulated by a backward ray-tracing rendering. Consider an optical system with only geometric principles. By sampling rays from sensor pixels and sequentially computing intersection and refraction with each optical surface, we can acquire the output rays, and further do rendering in the scene space. The intersection and refraction processes can be formulated as:

$$\mathbf{o}' = \mathscr{I}(\mathbf{o},\mathbf{d}) \iff \begin{cases} \mathbf{o}' = \mathbf{o} + \mathbf{d} * t \\ \mathscr{S}(\mathbf{o}') = 0 \end{cases}, \quad \mathbf{d}' = \mathscr{R}(\mathbf{o}',\mathbf{d}) \iff \begin{cases} \mathbf{n} = \nabla \mathscr{S} \\ \mathbf{n} \times \mathbf{d}' = \mu(\lambda) * \mathbf{n} \times \mathbf{d} \end{cases} \tag{1}$$

where $\mathbf{o}, \mathbf{d}$ denote ray position and direction, $\mathscr{I}, \mathscr{R}$ represent intersection and refraction, $\mathscr{S}$ is the surface function, $\mathbf{n}$ is the surface normal vector, and $\mu$ is the refraction index which is related to the wavelength. After going through the optical module, rays keep propagating until hitting the scenes. We treat the scenes as all-in-focus RGB images ("scene image") which has been proved suitable in the previous works [2, 6] if our target is to train a fixed focus lens. The illumination of each ray is computed by a bi-linear interpolation among neighbor four pixels,

$$I(w_i, w_j) = \begin{bmatrix} 1 - w_i & w_i \end{bmatrix} \begin{bmatrix} I(0,0) & I(0,1) \\ I(1,0) & I(1,1) \end{bmatrix} \begin{bmatrix} 1 - w_j \\ w_j \end{bmatrix} \tag{2}$$

where $I(0,0)$ denotes the bottom left pixel value, $w_i, w_j$ denote the horizontal and vertical distances between ray position and the bottom left pixel. The pixel value of the rendered image is determined by averaging the illumination of all rays sampled. The whole pipeline of our model is illustrated in Fig. 1 (a), and each stage of the pipeline can be either computed explicitly or implicitly with Newton iterations. To optimize a photographic lens, we want to minimize the difference between ground-truth images and rendered images. It is also equivalent to minimizing mixed optical aberrations, e.g., distortion and coma. In back-propagation, the gradient can be computed by the chain rule, and we use the auto-diff frameworks like PyTorch to compute the derivatives of each step.

*Memory Constraint* Differentiable ray tracing requires huge memory for both forward and backward propagation. The memory constraint has limited the sample-per-pixel (spp) value and rendered image resolution in the previous works [2, 7]. To settle this, we propose a memory-efficient strategy by separating the forward and backward passes. In the forward pass, differentiable mode is turned off and no intermediate parameters are stored, thus we can use a higher spp for rendering. In the backward pass, we split images into smaller patches and iterate them to compute and accumulate gradients.

*Curriculum Learning* Traditional optical design methods usually start on existing design forms and need experienced engineers to keep interfering with the optimization. This is because lens design is a highly non-convex optimization task, and the search space contains lots of local minimums, saddle points, and flat regions. We propose a curriculum learning method that dynamically adjusts the target region during the training to settle this challenge. We use a radiative Hamming window to select the region with maximum root-mean-square (RMS) error, and then apply a higher weight for them in the loss function.

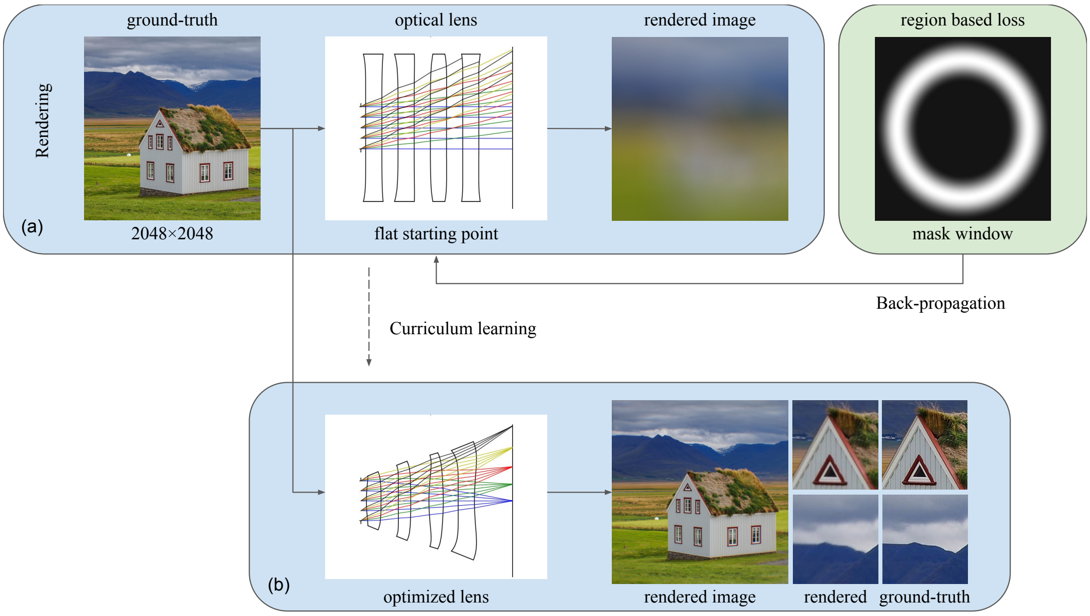


Fig. 1. Pipeline of our proposed differentiable ray-tracing model. Starting from a randomly generated design, our model can optimize lens parameters and positions for the best imaging quality. Attention window is dynamically adjusted for a faster training speed and getting out from local minimums.

## 2. Experiment Results

To apply our model to cellphone lens design, we first generate a ”flat” starting point with randomly initialized surface parameters and positions. We use a 1/2.3” sensor with an image height of 7.66 mm and optimize four pieces of even-order aspheric lenses. For the 1st, 2nd and 4th elements, we use N-PK51 ($n_d = 1.529$, $V_d = 76.98$), and for the 3rd element, we use N-LAK34 ($n_d = 1.755$, $V_d = 52.3$). During training, curvature $c$, conic term $k$, polynomial term parameters $a_i$, and position $d$ of all surfaces are optimized for the best imaging quality. The polynomial term is set up to the 8th order. The target field-of-view is set to 1.2 rad, which is commonly used in the commercial cellphone main lens. The F-number is 2.8. We also discretize the full spectrum to three wavelengths (486 nm, 587 nm, and 656 nm), and render different RGB channels for a more realistic rendering. The final design and rendered image can be seen in Fig. 1 (b).

## 3. Conclusion

In this paper, we propose a curriculum learning method and apply it to cellphone lens design. With our method, we can start without any prior design, constantly update lens parameters, and automatically jump out from local minima. We envision the proposed method can improve conventional optical design with less human interventions.

## References


1. G. Côté, J.-F. Lalonde, and S. Thibault, “Deep learning-enabled framework for automatic lens design starting point generation,” Opt. Express **29**, 3841 (2021).
2. Q. Sun, C. Wang, Q. Fu, X. Dun, and H. Wolfgang, “End-to-end complex lens design with differentiable ray tracing,” ACM Trans. Graph. **40**, 1–13 (2021).
3. C. Wang, C. Ni, and W. Heidrich, “dO: A differentiable engine for Deep Lens design of computational imaging systems,” (2021).
4. C. Wang, N. Chen, and W. Heidrich, “Towards self-calibrated lens metrology by differentiable refractive deflectometry,” Opt. Express **29**, 30284–30295 (2021).
5. C. Ni, C. Wang, and W. Heidrich, “Compact Computational Holographic Display (Invited Article),” Front. Photonics **3** (2022).
6. E. Tseng, A. Mosleh, F. Mannan, K. St-Arnaud, A. Sharma, Y. Peng, A. Braun, D. Nowrouzezahrai, J.-F. Lalonde, and F. Heide, “Differentiable compound optics and processing pipeline optimization for end-to-end camera design,” ACM Trans. Graph. **40**, 1–19 (2021).
7. M. Nimier-David, D. Vicini, T. Zeltner, and W. Jakob, “Mitsuba 2: A retargetable forward and inverse renderer,” ACM Trans. Graph. **38**, 1–17 (2019).