Results in Optics 15 (2024) 100643

Contents lists available at ScienceDirect

Results in Optics

journal homepage: www.sciencedirect.com/journal/results-in-optics

Using deep learning for effective simulation of ghost reflections

L. Clermont [a,*], G. Adam [b]

[a] Centre Spatial de Liège, STAR Institute, Université de Liège, Belgium
[b] Télécom Physique Strasbourg, Strasbourg, France

ARTICLE INFO

Keywords:
Ghost reflection
Deep learning
Stray light
Optical instrument
Autoencoder
EAM

ABSTRACT

Stray light (SL) control is an important aspect in the development of optical instruments. Iterations are necessary between design and analysis phases, where ray tracing simulations are performed for performance prediction. This process involves trial and error, requiring to be able to perform rapid evaluation of SL properties. The limitation is that accurate SL simulations require sending many rays, which can be time consuming. In this paper, we use deep learning to improve the accuracy of SL maps even when obtained with very few rays. Two different deep learning methods are used, an autoencoder and an EAM. The training process is performed by generating a large database of artificial SL maps, with different noise levels reproduced with a Poisson distribution. Once the training completed, we show that the autoencoder performs the best and improves significantly the accuracy of SL maps. Even with extremely small number of rays, it recovers complex SL patterns which are not visible on raw ray traced maps. This method thus enables more efficient iterations between design and analysis. It is also useful for developing SL correction algorithms, as it requires tracing SL maps under large number of illumination conditions in a reasonable amount of time.

1. Introduction

Simulating stray light (SL) is an important step in the design process of an optical instrument. SL is any light that reaches the detector through a different path or process than the image forming beam, thus reducing image quality (Fest, 2013; Breault, 1995; Breault, 1977). In refractive systems, ghosts are SL paths caused by the partial reflection of light on lenses interfaces (Fig. 1). If the instrument has m interfaces, it can produce up to $m\cdot(m-1)/2$ ghosts on the detector at second order specular reflection (Fest, 2013). Hence even a simple system with a few lenses can produce a complex SL patterns composed of many ghosts (Clermont et al., 2018; Clermont et al., 2022). Higher orders ghosts, involving more than two reflections, are possible but often negligible compared to second order ghost as the energy is decreased by additional reflections. SL can also be produced by the scattering of light on the lenses interfaces (Stover, 2012; Harvey, 2019; Vernold and Harvey, 1997). SL simulations are necessary to verify the SL level and to validate the opto-mechanical design (Fest, 2013; Breault, 1995; Clermont and Aballea, 2021; Gauvin and Freniere, 2000). Often, iterations are needed between the design and analysis phases until a satisfactory performance is reached. Therefore, it is useful to be able to perform simulations in a time efficient manner. Moreover, it is sometimes necessary to simulate the SL under many conditions, for example over a large number of individual illumination angles for the development of a SL correction algorithm (Clermont et al., 2018; Clermont et al., 2022).

SL simulations are performed with a ray tracing software, where an optical model of the instrument is built. Rays are sent in the system and they interact with the different surfaces according to their optical properties (Fest, 2013; Peterson, 1999). Ultimately, the SL pattern is obtained by recording the rays that fall on the detector. Usually, we consider an illumination from a point-like source in the field of view, in which case the SL pattern is called the spatial point source transmittance (SPST) (Clermont et al., 2018; Clermont et al., 2022). As a finite number of rays are sent, the SL pattern is affected by a stochastic error inversely proportional to the square root of the number of rays (Clermont et al., 2020). The efficiency of a SL simulation is improved with various best practices (Pfisterer, 2004; Pfisterer, 2014). For example, ray aiming traces scattered rays only toward a specific direction of interest. A limit can also be set on the number of times rays are split through partial reflections or scattering events. Another method is to illuminate the instrument only through the stray light entrance pupil (SLEP) and to optimize the ray density spatial distribution (Clermont et al., 2020). SL analysis also benefits from improvements in computational power, in particular with the use of GPU. Despite the use of these methods,

* Corresponding author.
E-mail address: lionel.clermont@uliege.be (L. Clermont).

https://doi.org/10.1016/j.rio.2024.100643
Received 21 October 2023; Received in revised form 27 January 2024; Accepted 6 February 2024
Available online 10 February 2024

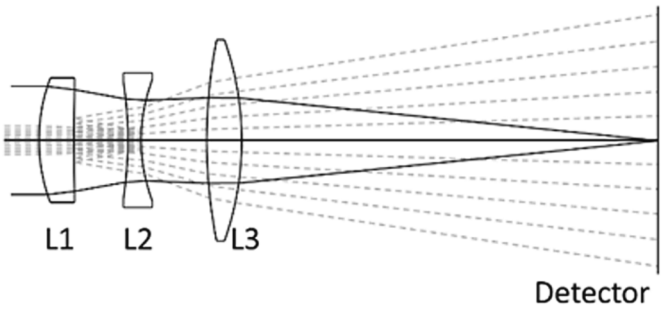

Fig. 1. An optical system with the image forming beam (solid line) and a ghost reflection between two lenses interfaces (dotted line).

performing a SL simulation is a time-consuming process. In the case of scattering on the lens interfaces, the SL pattern on the detector is smooth and therefore even with a relatively large noise it could be fitted with a model. For example, Harvey-Shack, ABg or even polynomial models (Harvey, 2019). Ghosts however present complex spatial distribution and therefore a simple fit isn't enough to get a SL map with reduced noise.

We use deep learning to improve the accuracy of ray tracing simulations for ghost reflections, especially when a very small number of rays are traced. We build a database of SL maps with different levels of ray tracing noise to train a deep learning algorithm. Afterward, we are able to estimate the noise free SL map from a noisy input map. Two particular deep learning algorithms are tested, both falling in the category of convolution neural networks. In this paper, we show that this method enables to significantly improve the accuracy of SL maps, in particular when the ray tracing is done with a very small number of rays. It thus represents a very useful tool for rapid evaluation of SL properties for optical instruments.

2. Methods

2.1. Deep learning approach

Deep learning, a branch of machine learning, is a series of computational methods which allows for a computer to perform tasks without being explicitly programmed to do so (Samuel, 1959; Koza et al., 1996; LeCun et al., 2015; Mitchell, 1997). Based on neural networks, it addresses problems by learning from data (Ma et al., 2021). The first phase consists in training the neural network, exposing it to a series of data which include the ground truth: the target of a task when entered with a given input. The training consists in calculating a loss function representing a distance between the ground truth and the output of the neural network. The parameters of the neural network are then optimized to minimize the loss function. Once the training is completed, the algorithm is able to realize the task of estimating the ground truth when entered with new input. Deep learning algorithms improve automatically through experience (Mitchell, 1997), hence it is necessary to have a large training dataset. While training is a time-consuming process, it is performed only once. Then, the application of the deep learning to make a prediction is nearly immediate. For example, for a 500x500 image, it takes 2 s using Python on a CPU with an Intel Xeon 2.20 GHz, and only 0.1 s on a GPU T4. This processing time is independent of the number of ghosts in the image or the spectral width used for ray tracing.

Applications of deep learning are numerous, from medical imaging (Anaya-Isaza et al., 2021) to speech recognition (Nassif et al., 2019), image rendering and denoising (Tewari, 2020; Tian et al., 2020; Chakravarty et al., 2017; Saeed et al.; Morteza et al.) or autonomous driving (Huang and Chen, 2020) among others. In the field of optics, deep learning is applied for computational imaging (Barbastathis, 2020; Li and Tian, 2022), photonic structure design (Ma et al., 2021; Tahersima et al., 2019) and lens design (Côté et al., 2019; Lalonde and Thibault, 2021; Côté, 2018). Many deep learning applications involve extracting information or improving quality of an image (Anaya-Isaza et al., 2021; Tewari, 2020). Therefore, deep learning appears as a good candidate for improving the accuracy of SL simulations. Here, the data to provide for the neural network training are noisy SL maps obtained with a small number of rays, and the ground truth which corresponds to the same maps obtained with a number of rays sufficiently large for ray tracing noise to be negligible.

Two different deep learning models are selected, both falling into the category of convolutional neural network (LeCun et al., 2015; Alzubaidi et al., 2021; O'Shea and Nash, 2015). The first is an autoencoder with approximately 600.000 parameters, initially developed by Chakravarty et al. for image rendering applications in the field of video games (Chakravarty et al., 2017). This technique is well suited for SL maps traced with fe the application of the deep learning to make a prediction is nearly immediate w rays, as it is developed to reconstruct sequences of images with extreme noise. Compared to the initial algorithm, only the autoencoder part is used while the recurrent part is removed, as the input is a single SL map rather than a sequence of images. An autoencoder is composed of two elements. The first one is the encoder that compresses data to a smaller dimension by removing unnecessary and useless information such as noise. The second is the decoder that reconstruct the image using the encoded data.

The second-deep learning technique uses enhancement attention modules (EAM) (Saeed et al.). It is composed of three parts: feature extraction, feature learning residual on the residual, reconstruction. This method is another good candidate for SL maps reconstruction as it is meant to predict images based on very noisy inputs. The high level principle is to extract features and to estimate the noise from it.

Both neural networks are trained on 30 epochs with a training dataset of 5000 SL maps, using the same custom loss function L consisting of three components: L_m, L_g and L_h. The first component, L_m, corresponds to the mean absolute error and is expressed by equation (1). In this equation, P_i is the value at pixel i on the target image, P'_i the value at pixel i on the predicted image and n the number of pixels. The second component, L_g, is a gradient domain loss used to penalize differences on edges in order to obtain ghosts with sharp edges. Expressed by equation (2), it uses a Laplacian of gaussian operation ∇ which computes the second derivative of the image (Sotak and Boyer, 1989). The third component, L_h, is a histogram loss similar to (Morteza et al.) but using a Kullback-Leibler divergence (Kullback and Leibler, 1951; Joyce and Lovric, 2011) as a metric to compute dissimilarities between the two probability distributions. This loss function is used to preserve the global structure of the target image on the predicted image. It also allows to work on distribution level rather than over individual pixels. This function is expressed by equation (3), where H and H' are the histogram of the target and predicted images, respectively. The histogram is evaluated on 2048 bins, and the sum in (3) goes on the different possible values of the histogram. The final loss function L is given by equation (4), with the coefficients set as $(\alpha_m, \alpha_g, \alpha_h) = (0.8, 0.1, 0.1)$. The goal of the deep learning training is to minimize the loss function L, and therefore to minimize the three components of the loss at the same time but with a different weight.

$$L_m = \frac{1}{n}\sum_i^n |P_i - P'_i| \tag{1}$$

$$L_g = \frac{1}{n}\sum_i^n |\nabla P_i - \nabla P'_i| \tag{2}$$

$$L_h = \sum_k H(k)\cdot log\frac{H(k)}{H'(k)} \tag{3}$$

$$L = \alpha_m L_m + \alpha_g L_g + \alpha_h L_h \tag{4}$$

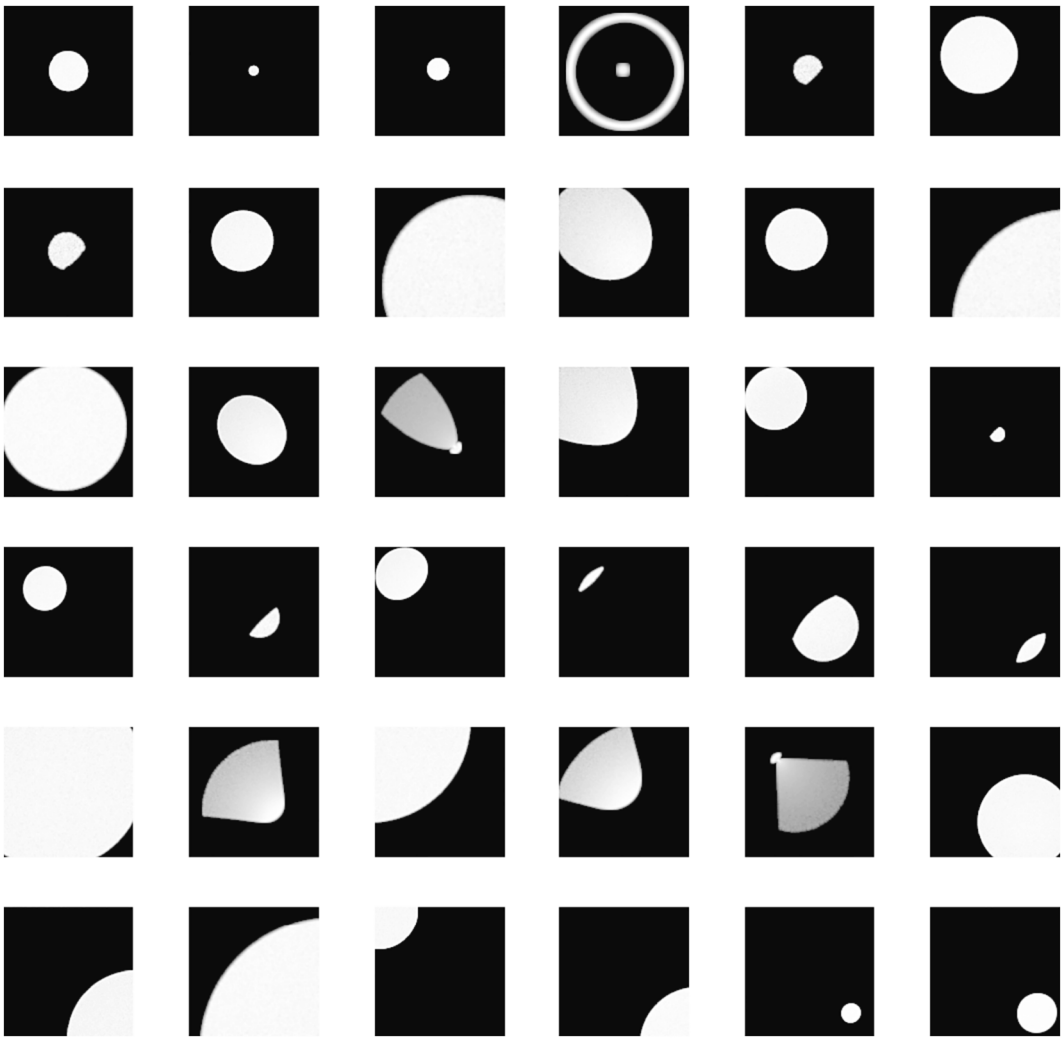

Fig. 2. Spatial distribution of individual ghosts maps, generated with a large number of rays to obtain a negligible ray tracing noise. The linear combination of individual ghost maps enables the construction of a database of artificial SL maps.

2.2. *Training database: Ground truth maps*

The training database must be filled with a large quantity of SL maps. One simple but inefficient way to generate the database would be to compute every single of them individually by ray tracing, based on the model of various optical systems under different illumination conditions to cover a wide variety of situations. Moreover, a very large number of rays should be traced for the ground truth maps, so that the ray tracing noise would be negligible. In practice, this is a cumbersome and time consuming task to be performed and therefore an alternate approach is preferred. We take advantage of SL being a linear and additive phenomenon (Laherrere et al., 1997): the SL pattern for an optical system is the sum of the patterns for the different ghosts created by the pairs of partially reflective interfaces. If a reasonably large library of individual ghosts maps was available, an infinite number of SL maps could be generated by combining them randomly. We will refer to these maps as "artificial SL maps", as they are representative of actual SL maps but do not relate to the model of a real optical system.

The library of individual ghosts is generated by ray tracing with the FRED software (FRED software). This is performed for a wide angle camera with a large number of lenses (Clermont et al., 2020; Manolis et al., 2014), providing several hundreds of individual ghost paths. A

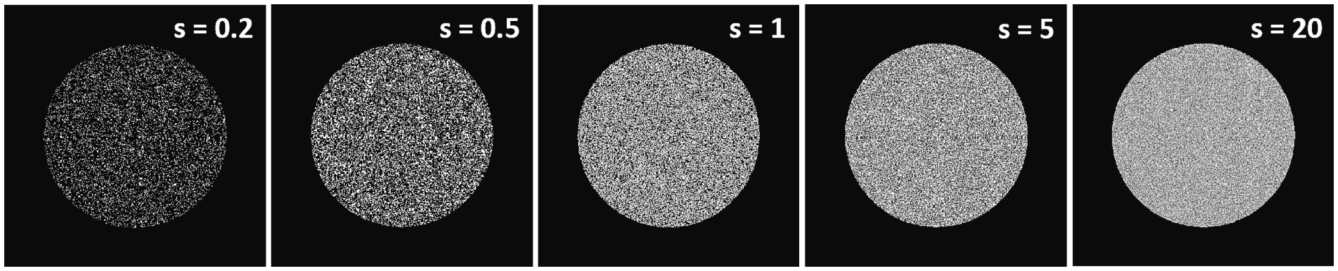

Fig. 3. Spatial distribution of a ghost obtained by ray tracing with an average of s rays per pixel. The map is very noisy for small values of s, while the map is more uniformly illuminated when s is increased.

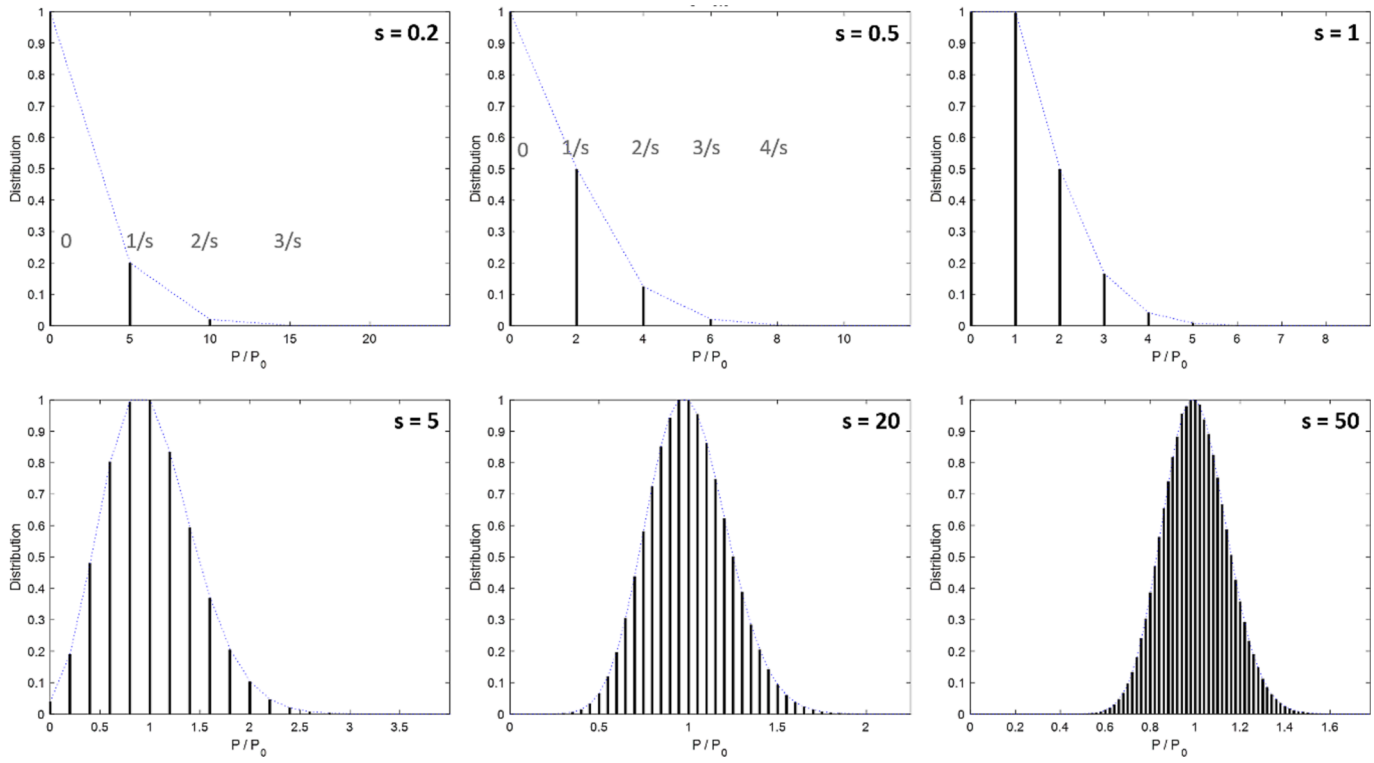

Fig. 4. Statistical distribution of the power P/P_0 on a pixel in the ghost, for different values of s, following a Poisson distribution.

large number of rays is traced so that the ray tracing noise on the individual maps is negligible. Furthermore the angle of illumination is varied to provide more diversity in the ghost patterns, including on-axis and off-axis ghosts. Ultimately, a library of 1500 individual ghosts is generated. Fig. 2 shows the maps for several of them, representing a broad variety of ghosts patterns. While some present a circular shape with a nearly uniform profile, others have exotic shapes and non-uniform energy distributions. Here, we considered angles of illumination along a 45° azimuth angle, therefore a mirror symmetry along the diagonal is observed for all the ghosts.

The generation of artificial SL maps is done by a linear combination of the individual ghost patterns. This is represented by Eq. (5), where the spatial distribution of the individual ghost j, g_j, is modulated by a random number γ_i to provide the distribution I of an artificial SL map. The sum can consider different number of ghost patterns to generate artificial maps associated to optical systems with different number of lenses. Finally, a database of 5000 artificial SL maps with arbitrarily complicated patterns was generated for the training of the neural network.

$$I = \sum \gamma_j \cdot g_j \tag{5}$$

2.3. Training database: Noisy maps

On top of the ground truth artificial SL map, we must generate the SL maps as they would appear in the presence of ray tracing noise. For that, we study the statistics of noise occurring from ray tracing, considering a simple circular and uniform ghost. Fig. 3 shows the spatial distribution of a ghost when traced with different number of rays, where s is the average number of rays received by a pixel. When s is very small, the map is extremely noisy and many pixels are not illuminated at all. When s is increased, the pixels are more uniformly illuminated and the noise is reduced.

When the number of rays is increased, the power per ray is decreased and thus the total power of the source is unchanged. Therefore, while the power P on any pixel is affected by noise, in average the pixels have a power P_0 independently of s. Fig. 4 shows the statistical distribution of the power P over the pixels for different values of s. It emphasizes a behaviour typical from a Poisson distribution (Haight, 1967), where the probability Γ for a pixel to have a power P is given by Eq. (6). For small values of s, there is a large probability that no rays fall on a pixel and thus the most probable value of P is 0. When s is typically above 20, the Poisson distribution is similar to a gaussian distribution with average value $\langle P \rangle = P_0$ and its standard deviation is inversely proportional to $\sqrt{s}$. For all values of s, the power P/P_0 can only take discrete values multiple of $1/s$. This gives the possibility for only a limited number of values when s is small but nearly a continuum when s is large.

$$\Gamma(P) = \frac{s^{\left(\frac{P}{P_0} \cdot s\right)}}{\left(\frac{P}{P_0} \cdot s\right)!} e^{-s} \tag{6}$$

To reproduce the ray tracing noise on the artificial SL maps, a noise with Poisson distribution is applied on each individual ghost map from the library. We assume that each ghost has the same total number of rays N. The average number of rays per pixel s_j inside the ghost j is thus given by Eq. (7), where A_j is the area of the ghost in pixels. In practice, the different ghosts do not necessarily have the same number of rays as they can be vignetted differently through the optical systems. It is nevertheless a good approximation as it means that a ghost more focused on the detector has a larger ray density s than a widespread ghost, hence the noise level is larger for large ghosts. Finally, we obtain the noisy ghost map g_j^* by multiplying the value of each pixel by a random number $Poisson(s, P_0)$, following a distribution as defined by Eq. (6) with $s = s_j$ and $P_0 = 1$. The noisy artificial SL map I^* is then obtained by combining the noisy maps, as shown on Eq. (9). This is performed for different values of N to generate a database of SL maps with different levels of ray tracing noise.

$$s_j = \frac{N}{A_j} \tag{7}$$

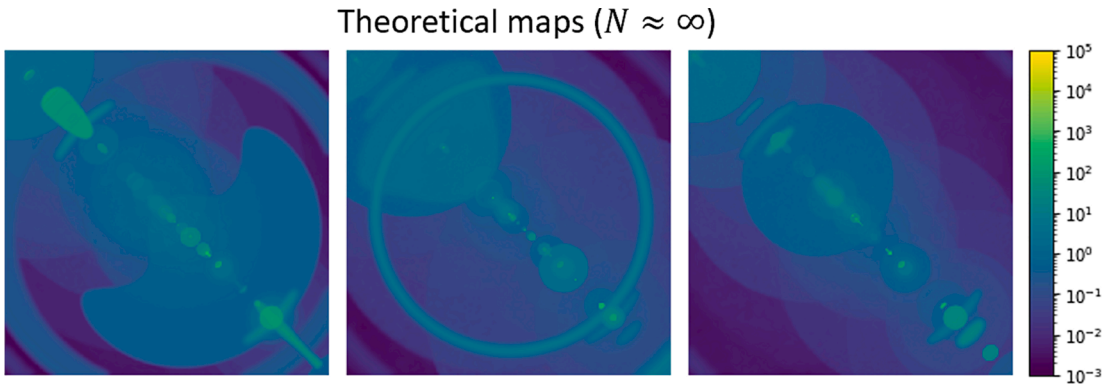

Fig. 5. SL maps reproduced artificially with the library of individual ghost maps.

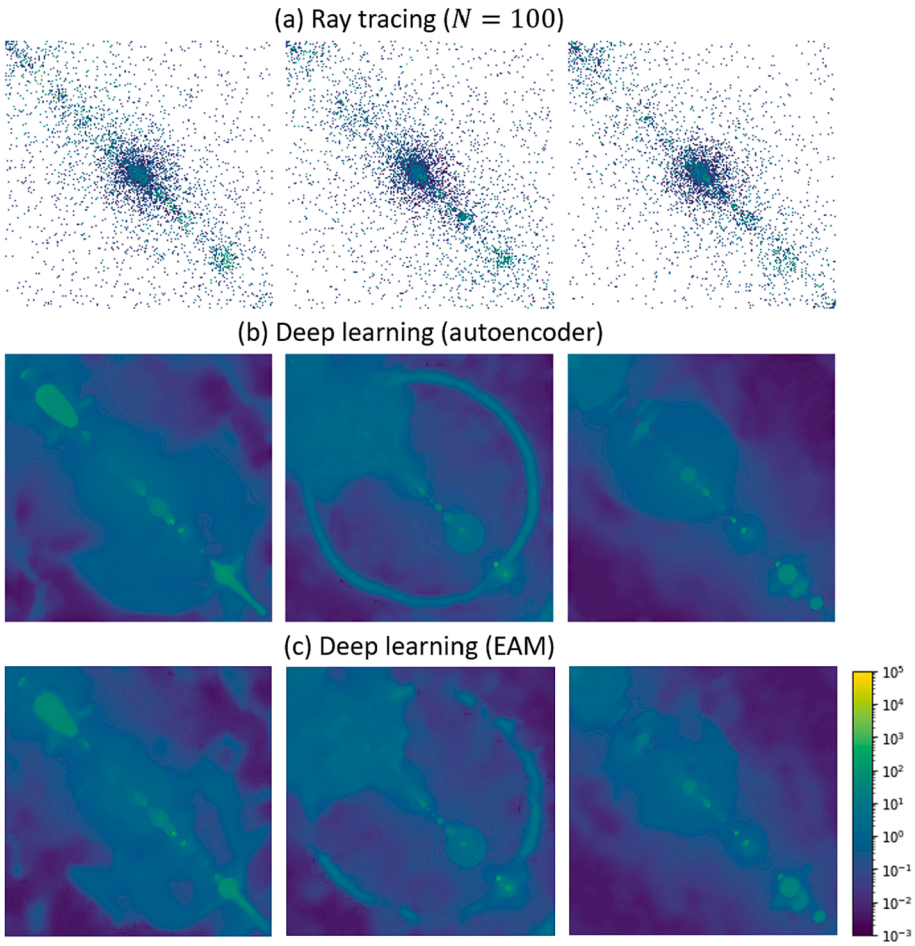

Fig. 6. (a) Artificial SL maps with a ray tracing noise model for N = 100. (b) Reconstructed SL map with deep learning autoencoder. (b) Reconstructed SL map with deep learning EAM model.

$$g_j^*(x,y) = g_j(x,y) \cdot Poisson\left(s = s_j; P_0 = 1\right) \tag{8}$$

$$I^*(x,y) = \sum \gamma_j \cdot g_j^*(x,y) \tag{9}$$

3. Results

3.1. Artificial SL maps

Fig. 5 shows examples of SL maps generated artificially and without considering ray tracing noise. This is the kind of maps generated to constitute the ground truth of the training dataset (theoretical maps), however these ones in particular were not used for the training but rather are generated to test the neural network once it is trained. The maps are composed of a large quantity of ghosts with arbitrarily complex features. The three maps are very different from each other, despite being generated from the same library of individual ghosts. They represent typical SL patterns that can be obtained in real optical instruments.

Fig. 6(a) shows the same maps as Fig. 5 when ray tracing noise is considered, applying the Poisson model with $N = 100$. With such a small number of rays, the SL maps are extremely noisy and many pixels

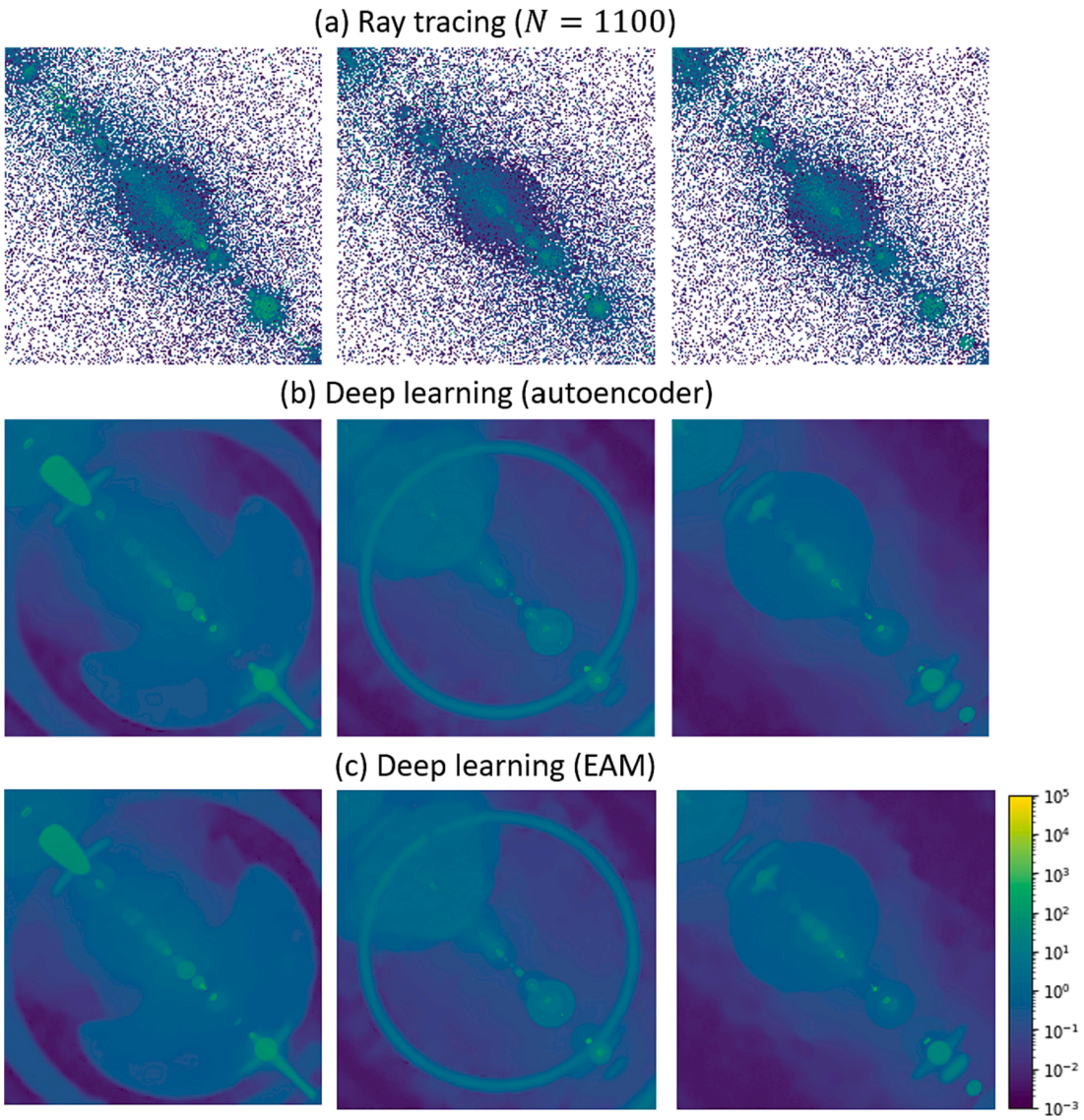

Fig. 7. (a) Artificial SL maps with a ray tracing noise model for N = 1100. (b) Reconstructed SL map with deep learning autoencoder. (c) Reconstructed SL map with deep learning EAM model.

are not illuminated. These maps are so inaccurate that they would in practice not be exploited for the sake of a stray light analysis. Nevertheless, we show that the application of the deep learning model on these maps enable to reconstruct predicted maps very similar to the ground truth. Fig. 6(b) and (c) show the results using the deep learning autoencoder and EAM models respectively. Remarkably, the neural network models are able to reconstruct complex features that are impossible to distinguish on the noisy input maps (a). While not

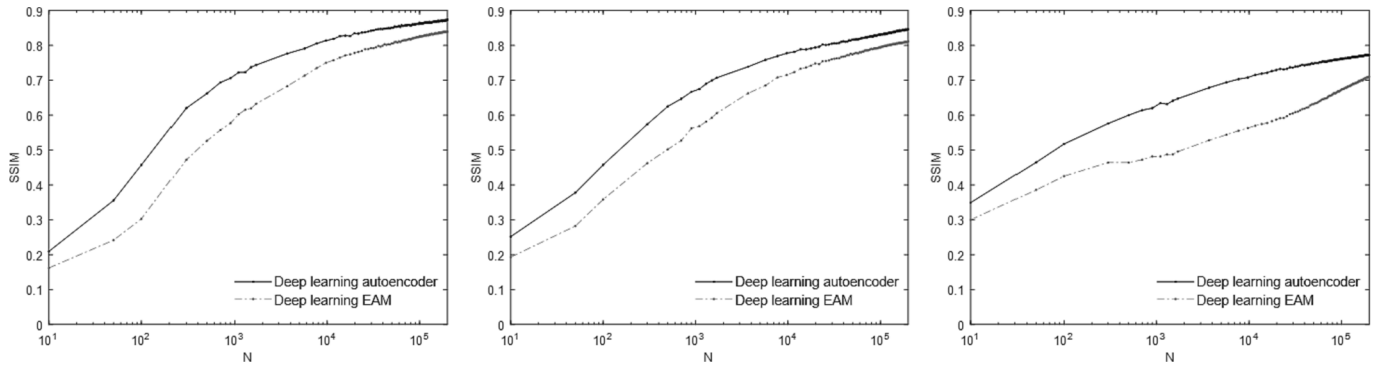

Fig. 8. SSIM of the maps from Fig. 5 when predicted with the deep learning autoencoder and EAM models, as a function of the number of rays N in the input map. The autoencoder systematically gives a better prediction as its value is closer to 1 than the EAM.

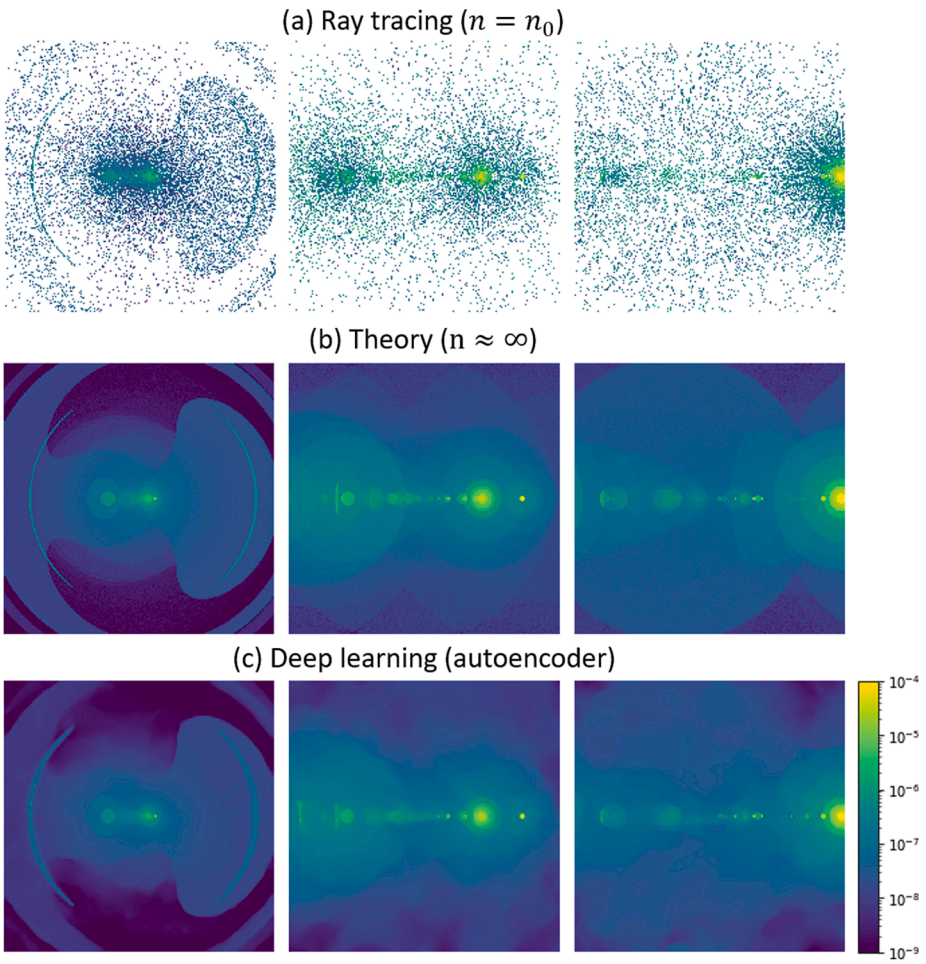

Fig. 9. SL maps for a wide-angle camera, at fields 5°, 27°, 50°. Noisy maps obtained by ray tracing with a small number of rays $n = n_0$ (a) or a very large number of rays (b). (c) Maps reconstructed with the deep learning autoencoder, considering as an input the maps ray-traced with..$n = n_0$

perfectly reconstructed, the deep learning provides maps which are exploitable for interpretation even though the number of rays traced is extremely low. In particular, the autoencoder gives results which a visually the best. Fig. 7 shows similar results, where the number of rays per ghost N is increased to 1100. While the noisy maps are still unexploitable, both deep learning models are able to reconstruct SL maps which are extremely similar to the ground truth.

A useful tool for quantitative assessment of deep learning prediction is the structural similarity index measure (SSIM). Instead of comparing the images pixel by pixel, it measures the similarity between two images by evaluating changes in structural information (Wang et al., 2005). The SSIM compares the target map with the predicted map and gives a value smaller or equal to 1. The closer to 1 and the better the prediction, with SSIM = 1 meaning a perfect prediction. Here, as SL maps are often characterized by a large dynamic range, the SSIM is applied on a log scale. Fig. 8 shows the SSIM for the different SL maps reconstructed by the deep learning autoencoder and EAM models, as a function of the number of rays per ghost N in the input map. As it shows, in all cases the SSIM increases towards 1 when N increases. Moreover, the deep learning autoencoder gives a SSIM systematically closer to 1 than EAM model. Therefore, the autoencoder is preferred over the EAM, which allows the best to reconstruct SL maps based on ray tracing with a very small number of rays.

3.2. Real SL maps

In this section, we apply the deep learning approach to recover the SL maps ray traced for a real optical system. The first system which is considered is same wide-angle camera used to generate the library of individual ghosts for the neural network training. Three illumination angles are considered, 5°, 27° and 50°, all with an azimuth angle of 0°. These angles are different than those used for generating the ghost library. Fig. 9(a) shows the ray tracing results with a very small number of rays, showing as expected very noisy maps. Here, the total number of rays sent at the input of the system ($n = n_0$) is in the tens of thousands. However, only a fraction of these rays are able to reach the detector as we are in the case of a wide-angle camera, where the pupil stop is much smaller than the size of the input aperture. For the sake of comparison, these maps were also computed with a much larger number of rays so that ray tracing noise is negligible (Fig. 9(b)). The deep learning autoencoder is applied and the predicted maps are shown on Fig. 9(c). As it shows, complex features are reconstructed and the map is very similar to the target.

Fig. 10 shows radial profiles of the SL map along the horizontal axis, for an illumination angle of 5°. The left figure shows the profile of the ray tracing maps, obtained with different number of rays: $n = n_0$, $n = 10{\cdot}n_0$ and $n = 100{\cdot}n_0$. The graphs also contain the profile of the target for comparison. The right plots show the profile of the maps reconstructed with the deep learning algorithm. As it shows, for $n = n_0$ many pixels are not illuminated and the ray tracing profile is extremely inaccurate. When applying the deep learning model however, the reconstructed map has a profile close to the target. For $n = 10{\cdot}n_0$ and $100{\cdot}n_0$ the profiles of the ray tracing maps are better but still extremely noisy, while the deep learning reconstruction are very close to the target,

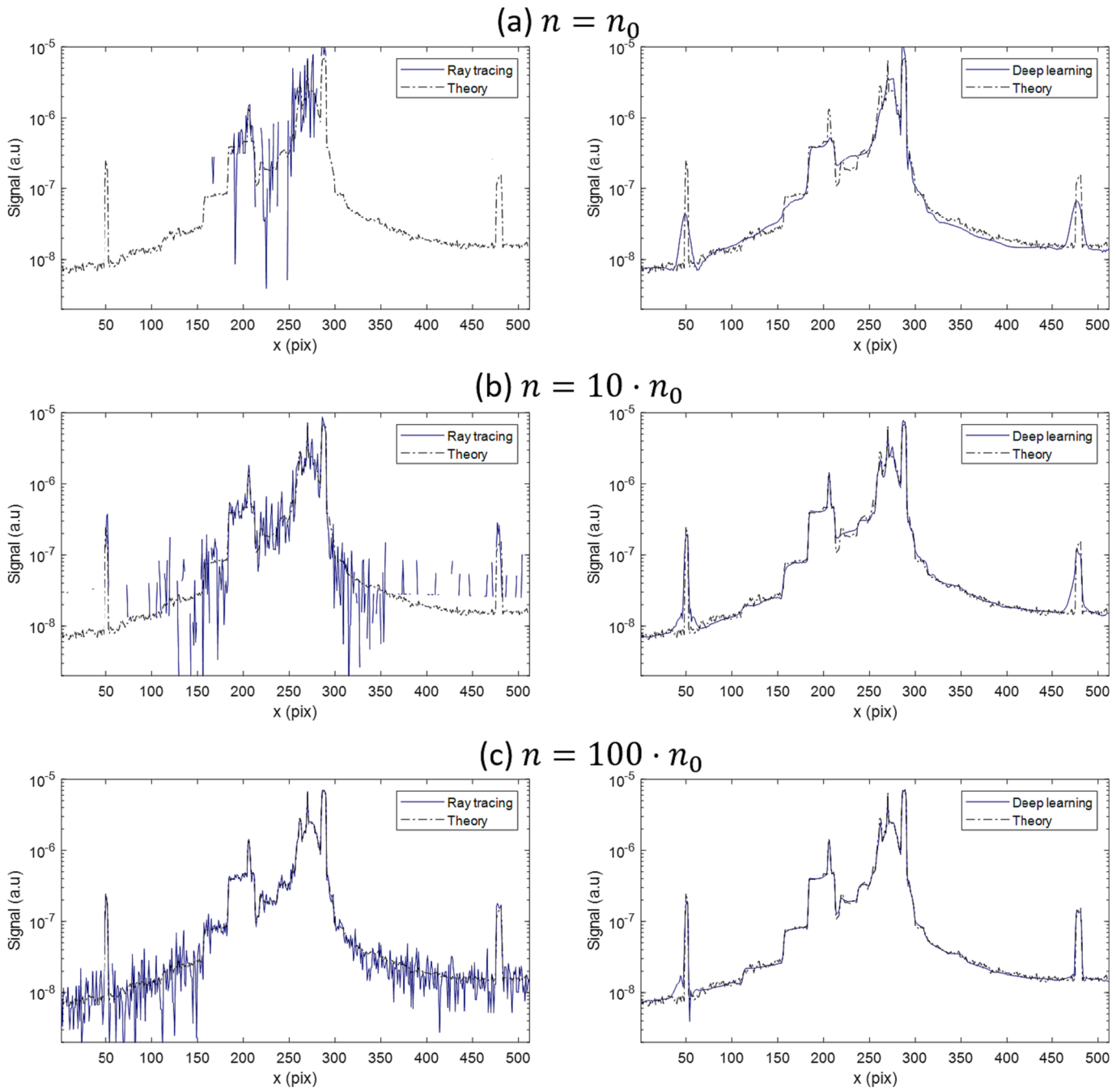

Fig. 10. Radial profile of the SL maps, for a wide-angle camera at field 5°. The profiles compare the theoretical map, obtained with a very large number of rays, with the ray traced maps obtained with a limited number of rays ($n = n_0$, $10{\cdot}n_0$ and $100{\cdot}n_0$), and the map obtained with the deep learning based on that ray tracing.

even capable to retreive small details.

Next, the method is applied on a second optical system, completely different from the first, and which was not used for the generation of the ghost library. The goal is to demonstrate that the training based on the library computed for the first insturment presents enough variety so that we can apply the deep learning model on different instruments as well. The selected instrument is a movie lens, selected from an optical database from the software CodeV (CodeV software) (layout from U.S. Patent 3464763). Its zoom configuration is set to a focal length of 19.5 mm, yielding a field of view of $\pm$ 10°. Fig. 11 shows a sketch of the system.

For the illumination angles 0°, 7° and 10°, Fig. 12(a) and (b) show the SL map obtained respectively for a small number of rays, $n = n_0$, and for a very large number of rays for the sake of comparison. Here the value n_0 is also in the several thousand but only a fraction reaches the detector. The maps obtained with the small number of pixels are very noisy and unexploitable in practice. When applying the deep learning model, we are able to reconstruct SL maps shown on Fig. 12(c), which emphasizes the complex features of the target. Similarly as for the first instrument, Fig. 13 shows the profiles of the SL maps when considering different number rays ($n = n_0$, $10{\cdot}n_0$ and $100{\cdot}n_0$). The conclusions are similar as for the first instrument. When n is small the profile of the ray traced maps is extremely inaccurate with many pixels not illuminated, while the profile of the map reconstructed with the deep learning is very close to the profile of the target map, and the accuracy is even more excellent when the number of rays of the input map is increased. This shows that the deep learning approach works well for reconstructing SL maps based on ray tracing even with a very small number of rays, for different kind of instruments. This is therefore a particularly interesting

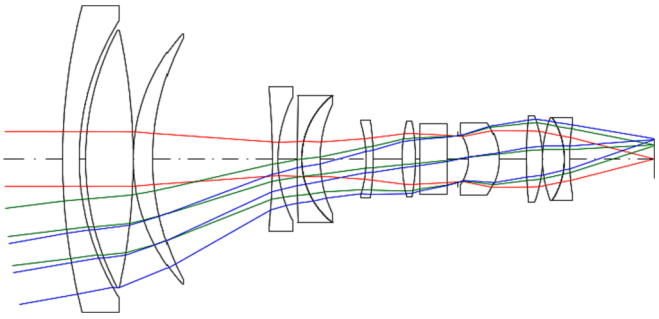

Fig. 11. Sketch of the U.S. Patent No. 3464763 movie lens, set to a focal length of 19.5mm, yielding a field of view of 10°.

tool for stray light analysis where rapid iterations are required between the design and analysis phases.

4. Conclusion

In this paper, we use deep learning to improve the accuracy of ray traced SL maps. Two deep learning methods are tested: autoencoder and EAM. The neural networks are trained by exposing them to a database of SL maps with different levels of ray tracing noise. To populate the database, we create artificial SL maps by combining randomly individual ghost patterns obtained by ray tracing with negligible noise. Various levels of ray tracing noise are obtained by applying on the ghost a random error following a Poisson distribution. The neural networks are then trained by optimizing their parameters to minimize a composed loss function. After the training process is completed, the deep learning methods are tested on artificial SL maps with different levels of noise. Even with very noisy SL maps, it is able to recover complex SL features present in the target maps. Both visually and using the SSIM method, we find that the autoencoder method performs better than the EAM and therefore the former is preferred. When the deep learning method is applied on real SL maps obtained by simulations on actual optical systems, it is also able to recover the complex patterns present in the target maps. This method allows to obtain accurate simulation of the SL properties of an instrument even when performing a ray tracing simulation with a very small number of rays. This is particularly useful for rapid evaluation of SL properties in optical instrument, where rapid iterations between design and analysis is necessary. Also it will be useful

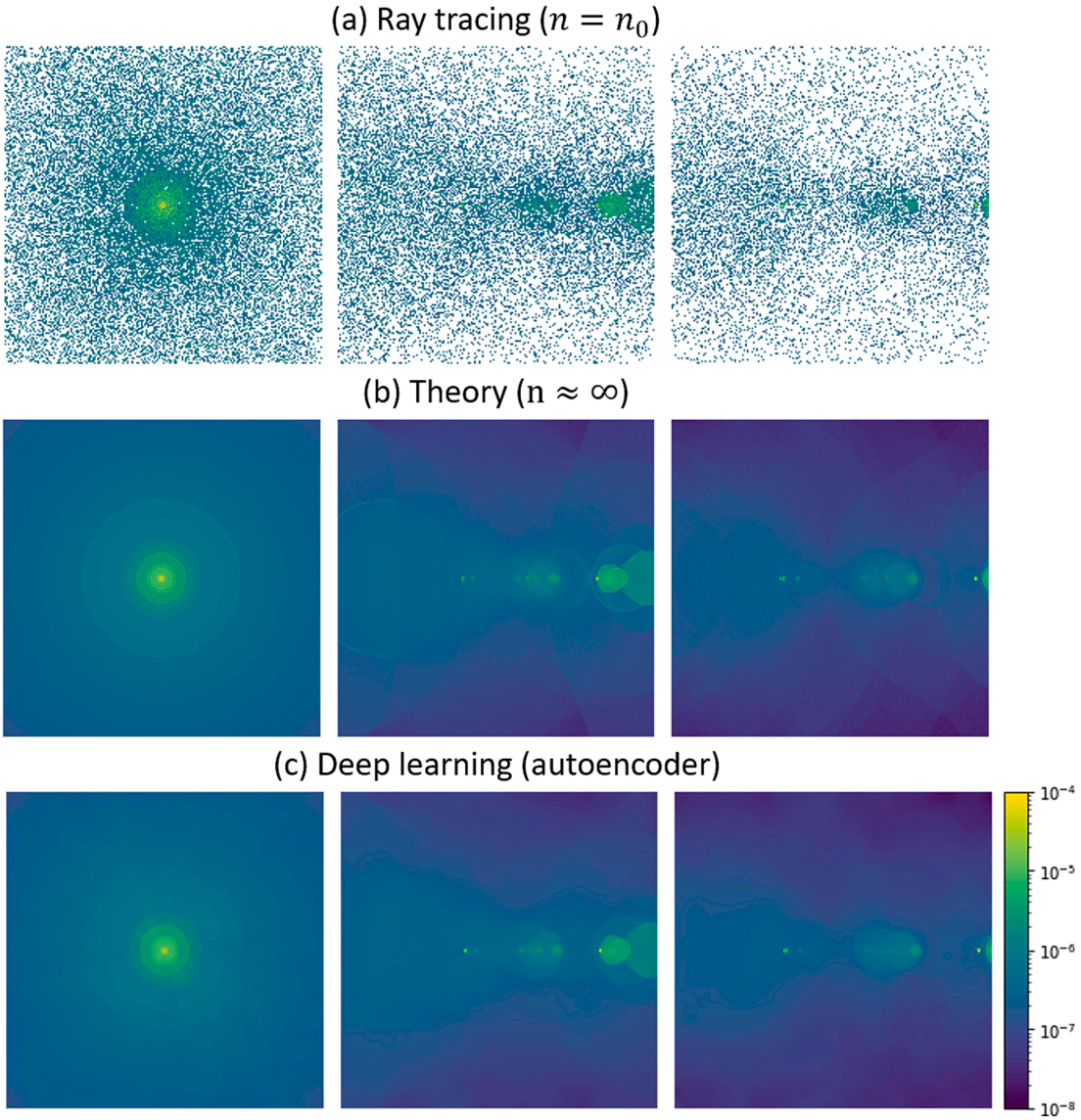

Fig. 12. SL maps for a movie lens, at fields 0°, 7°, 10°. Noisy maps obtained by ray tracing with a small number of rays $n = n_0$ (a) or a very large number of rays (b). (c) Maps reconstructed with the deep learning autoencoder, considering as an input the maps ray traced with..$n = n_0$

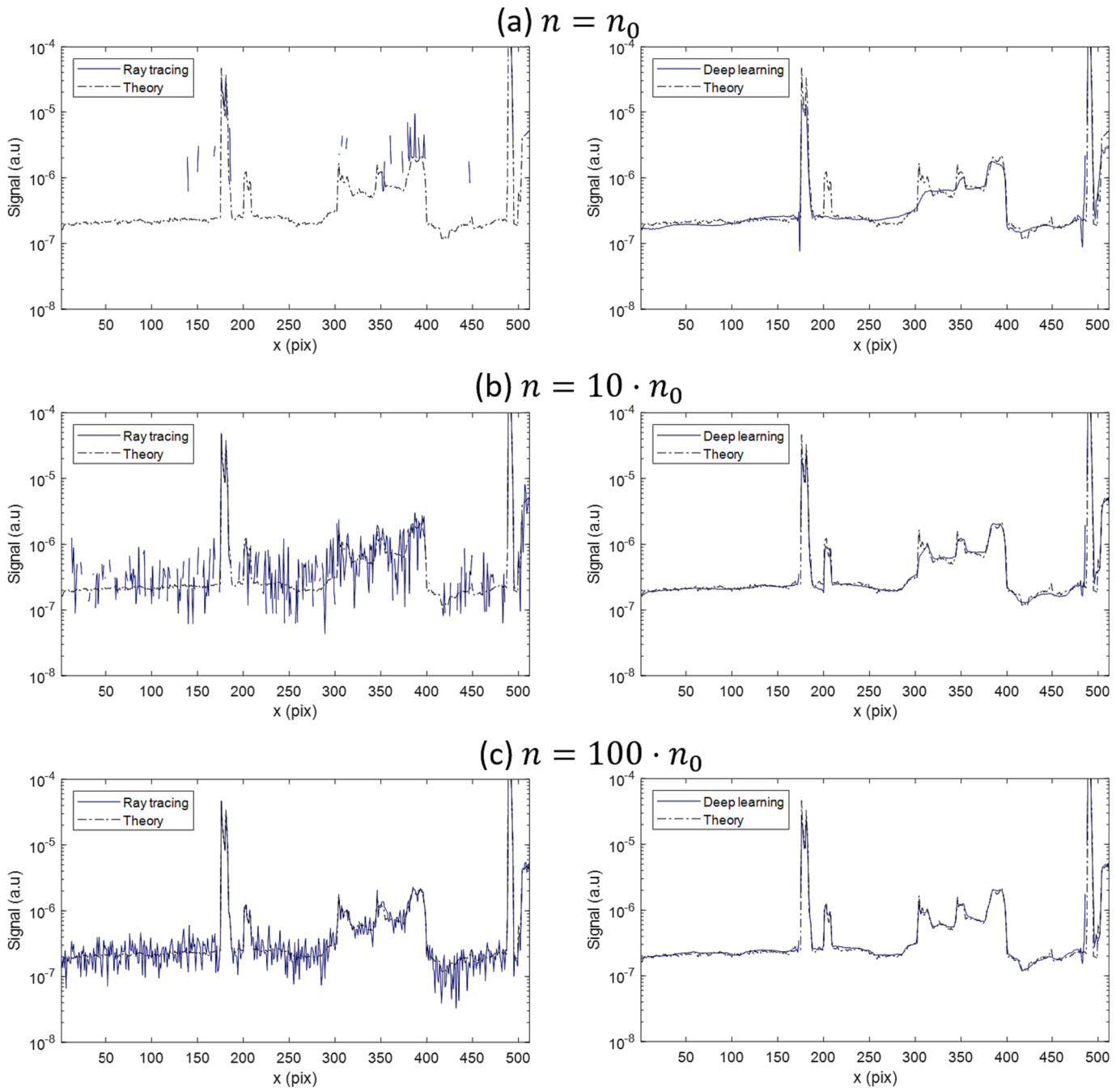

Fig. 13. Radial profile of the SL maps, for a movie lens at field 10°. The profiles compare the theoretical map, obtained with a very large number of rays, with the ray traced maps obtained with a limited number of rays (($n = n_0$, $10{\cdot}n_0$ and $100{\cdot}n_0$)), and the map obtained with the deep learning based on that ray tracing.

to estimate rapidly the SL maps for a large quantity of illumination conditions, for example when developing SL correction algorithms.

The deep learning approach could be improved by creating a database with artificial maps even more representative of real optical systems. For example, we made the assumption of a noise modeled with a simple Poisson distribution when considering a uniform ray density on each ghost. In practice we could consider as well a variable ray density over the ghost pattern, which can happen due to the bending of the rays by the lenses.

Funding

This research did not receive any specific grant from agencies in the public, commercial, or not-for-profit sectors. Instead, it was performed on the free time of the authors, driven by passion only.

CRediT authorship contribution statement

L. Clermont: Conceptualization, Formal analysis, Investigation, Methodology, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **G. Adam:** Conceptualization, Formal analysis, Software, Validation, Visualization, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no conflict of interest.

Data availability

Data will be made available on request.

References

Alzubaidi, L., Zhang, J., Humaidi, A.J., et al., 2021. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. J. Big Data 8, 53. https://doi.org/10.1186/s40537-021-00444-8.

Anaya-Isaza, A., Mera-Jiménez, L., Zequera-Diaz, M., 2021. An overview of deep learning in medical imaging. Informat. Med. Unlocked 26, 100723. https://doi.org/10.1016/j.imu.2021.100723.

Barbastathis, G. On the use of deep learning for computational imaging. in (Dholakia, K. & Spalding, G. C. eds.) Optical Trapping and Optical Micromanipulation XVII, Vol. 54, 10.1117/12.2571322 (SPIE, Online Only, 2020).

R.P. Breault, "control of Stray Light", Chapter 38 in Handbook of Optics, 1995. Vol. 1, 38.1– 38.35.

Chakravarty et al., "Interactive Reconstruction of Monte Carlo Image Sequences using a Recurrent Denoising Autoencoder", ACM Transactions on Graphics, Vol. 36, No. 4, Article 98. July 2017. https://doi.org/10.1145/3072959.3073601.

Breault, R.P., 1977. Problems and techniques in stray radiation suppression. Proc. SPIE 0107.

Clermont, L., Michel, C., Blain, P., Loicq, J., Stockman, Y., 2020. Stray light entrance pupil: an efficient tool for stray light characterization. Opt. Eng. 59 (2), 025102.

Clermont, L., Michel, C., Mazy, E., Pachot, C., Daddi, N., Mastrandread, C., Stockman, Y., 2018. Stray-light calibration and correction for the MetOp-SG 3MI mission. Proc. SPIE 10704, 1070406.

Clermont, L., Aballea, L., 28 May 2021. Stray light control and analysis for an off-axis three-mirror anastigmat telescope. Opt. Eng. 60 (5), 055106 https://doi.org/10.1117/1.OE.60.5.055106.

Clermont, L., Michel, C., Stockman, Y., 2022. Stray Light Correction Algorithm for High Performance Optical Instruments: The Case of Metop-3MI. Remote Sens. 14, 1354. https://doi.org/10.3390/rs14061354.

CodeV software, Version 18.61, edited by Synopsis (www.synopsis.com).

Côté, G., Lalonde, J.F., Thibault, S., 2019 Sep 30. Extrapolating from lens design databases using deep learning. Opt. Express 27 (20), 28279–28292. https://doi.org/10.1364/OE.27.028279. PMID: 31684583.

Geoffroi Côté, Jean-François Lalonde et Simon Thibault : Toward Training a Deep Neural Network to Optimize Lens Designs. In Frontiers in Optics / Laser Science, page JW4A.28. Optical Society of America, septembre 2018.

Fest, E., 2013. Stray Light Analysis and Control. SPIE Press, Bellingham, Washington.

FRED software, Version 18.61, edited by Photon Engineering (www.photonengr.com).

Gauvin, M.A., Freniere, E.R., 2000. Reducing stray light in Opto-Mechanical Systems. In: 2nd International Conference on Optical Design and Fabrication, pp. 277–280.

Haight, F.A., 1967. Handbook of the Poisson Distribution. John Wiley & Sons, New York, NY, USA.

Harvey, J.E., 2019. Understanding Surface Scatter: A Linear Systems Formulation. SPIE Press, Bellingham, Washington.

Y. Huang and Y. Chen, "Survey of State-of-Art Autonomous Driving Technologies with Deep Learning," 2020 IEEE 20th International Conference on Software Quality, Reliability and Security Companion (QRS-C), Macau, China, 2020, pp. 221-228, doi: 10.1109/QRS-C51114.2020.00045.

Joyce, J.M., 2011. Kullback-Leibler Divergence. In: Lovric, M. (Ed.), International Encyclopedia of Statistical Science. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-04898-2_327.

Koza, J.R., Bennett, F.H., Andre, D., Keane, M.A. (1996). Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. In: Gero, J.S., Sudweeks, F. (eds) Artificial Intelligence in Design '96. Springer, Dordrecht. 10.1007/978-94-009-0279-4_9.

S. Kullback, R. A. Leibler , On Information and Sufficiency, Ann. Math. Statist. 22(1): 79-86 (March, 1951). DOI: 10.1214/aoms/1177729694.

Jean-Marc Laherrere, Laurent Poutier, Thierry Bret-Dibat, Olivier Hagolle, C. Baque, P. Moyer, and E. Verges "POLDER on-ground stray light analysis, calibration, and correction", Proc. SPIE 3221, Sensors, Systems, and Next-Generation Satellites, (31 December 1997); 10.1117/12.298073.

François Lalonde, and Simon Thibault. 2021. Deep learning-enabled framework for automatic lens design starting point generation. Opt. Express 29, 3 (Feb 2021), 3841–3854.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. https://doi.org/10.1038/nature14539.

Li, Y., Tian, L., 2022. Computer-free computational imaging: optical computing for seeing through random media. Light Sci. Appl. 11, 37. https://doi.org/10.1038/s41377-022-00725-8.

Ma, W., Liu, Z., Kudyshev, Z.A., et al., 2021. Deep learning for the design of photonic structures. Nat. Photonics 15, 77–90. https://doi.org/10.1038/s41566-020-0685-y.

I. Manolis, J.-L. Bézy, R. Meynart, M. Porciani, M. Loiselet, G. Mason, D. Labate, U. Bruno, and R. De Vidi "The 3MI instrument on the Metop second generation", Proc. SPIE 10563, International Conference on Space Optics — ICSO 2014, 1056324 (17 November 2017); 10.1117/12.2304182.

Tom M. Mitchell, Machine learning, International Edition. McGraw-Hill Series in Computer Science, McGraw-Hill 1997, ISBN 978-0-07-042807-2, pp. I-XVII, 1-414.

Morteza et al., Adversarial Distortion Learning for Medical Image Denoising. 10.48550/arXiv.2204.14100.

Nassif, A.B., Shahin, I., Attili, I., Azzeh, M., Shaalan, K., 2019. Speech Recognition Using Deep Neural Networks: A Systematic Review. IEEE Access 7, 19143–19165. https://doi.org/10.1109/ACCESS.2019.2896880.

O'Shea, K., Nash, R.: An introduction to convolutional neural networks. https://arxiv.org/abs/1511.08458 (2015).

Peterson, G., 1999. Stray light calculation methods with optical ray trace software. Proc. SPIE 3780.

Pfisterer, R., 2004. Clever tricks in optical engineering. Proc. SPIE 5524.

R. Pfisterer, "FRED Optical Engineering Software Tutorial", Unpublished Software Tutorial Manual from Photon Engineering (2014).

Saeed et al., Real Image Denoising with Feature Attention. 10.48550/arXiv.1904.07396.

Samuel, A.L., 1959. Some studies in machine learning using the game of checkers. IBM J. Res. Dev. 3 (3), 210–229.

Sotak, G.E., Boyer, K.L., 1989. The laplacian-of-gaussian kernel: A formal analysis and design procedure for fast, accurate convolution and full-frame output. Comput. Vis. Graphics Image Process. 48 (2), 147–189. https://doi.org/10.1016/S0734-189X(89)80036-2.

Stover, J.C., 2012. Optical Scattering: Measurement and Analysis. SPIE Press, Bellingham, Washington.

Tahersima, M.H., Kojima, K., Koike-Akino, T., et al., 2019. Deep Neural Network Inverse Design of Integrated Photonic Power Splitters. Sci. Rep. 9, 1368. https://doi.org/10.1038/s41598-018-37952-2.

Tewari, A., et al., 2020. State of the art on neural rendering. Comput. Graph. Forum 39 (2), 701–727.

Tian, C., Fei, L., Zheng, W., Yong, X.u., Zuo, W., Lin, C.-W., 2020. Deep learning on image denoising: An overview. Neural Netw. 131, 251–275. https://doi.org/10.1016/j.neunet.2020.07.025.

Vernold, C.L., Harvey, J.E., 1997. Comparison of Harvey-Shack scatter theory with experimental measurements. Proc. SPIE 3141.

Wang, Z., Bovik, A., Sheikh, H., 2005. Structural similarity based image quality assessment. Digital Video Image Quality and Perceptual Coding, Marcel Dekker Series in Signal Processing and Communications.