ADVANCED PHYSICS RESEARCH Open Access
www.advphysicsres.com

Differentiable Imaging: A New Tool for Computational Optical Imaging

Ni Chen, Liangcai Cao, Ting-Chung Poon, Byoungho Lee, and Edmund Y. Lam*

The field of computational imaging has made significant advancements in recent years, yet it still faces limitations due to the restrictions imposed by traditional computational techniques. Differentiable programming offers a solution by combining the strengths of classical optimization and deep learning, enabling the creation of interpretable model-based neural networks. Through the integration of physics into the modeling process, differentiable imaging, which employs differentiable programming in computational imaging, has the potential to overcome challenges posed by sparse, incomplete, and noisy data. As a result, it has the potential to play a key role in advancing the field of computational imaging and its various applications.

Computational imaging,[1] which merges the fields of imaging and computation, extracts valuable information from indirect measurements, enabling capabilities that cannot be achieved with conventional techniques. This approach has opened up new avenues in various aspects of imaging and often provides results beyond the physical limitations of optics. Computational imaging offers new forms of visual information,[2,3] reduced hardware complexity and cost,[4,5] and higher resolutions compared to traditional imaging techniques.[6,7] It has been successfully applied in various fields, including science and medicine, commerce and industry, safety, security, and defense. However, there are still challenges and opportunities in the field of computational imaging that require further exploration.

Differentiable programming, first introduced as "Differentiable Functional Programming" in 2015,[8] has gained popularity in recent years as a prominent research area and is widely considered the future of software engineering. This form of programming is a generalization of deep learning and has been unlocked by the popularity of machine learning frameworks such as TensorFlow,[9] PyTorch,[10] and JAX.[11] Differentiable programming has been applied to various research fields, including imaging, where it has been successfully used to overcome challenges in current computational imaging techniques. Examples of such applications include ptychography,[12] X-Ray tomography,[13] optical microscopy,[14,15] electron microscopy for denoising and phase imaging.[16,17]

In computational imaging, the full potential of differentiable programming has yet to be fully realized. This perspective paper aims to revisit the challenges in computational imaging and examine the promise of incorporating differentiable programming into this field, referred as differentiable imaging. This paper is organized as follows. First, an overview of computational imaging and differentiable programming is provided in Sections 1 and 2, respectively. In Section 3, the challenges of computational imaging are discussed along with the capabilities of differentiable programming, and how the latter could potentially overcome these challenges. Finally, the perspective is summarized in Section 4.

1. Computational Imaging

We distinguish computational imaging from its usage in the computer science community which typically refers to a one-to-one mapping from object space to image space. On the contrary, we are interested in understanding imaging physics and adopt the definition of computational imaging as the joint design of front-end optics and post-detection computing to

N. Chen
Wyant College of Optical Sciences
University of Arizona
Tucson, AZ 85721, USA
E-mail: nichen@arizona.edu
N. Chen
Physical Science and Engineering
King Abdullah University of Science and Technology
Thuwal 23955, Saudi Arabia
L. Cao
Department of Precision Instruments
Tsinghua University
Beijing 100084, China
T.-C. Poon
Bradley Department of Electrical and Computer Engineering
Virginia Tech
Blacksburg, VA 24061, USA
B. Lee
Department of Electrical and Computer Engineering
Seoul National University
Seoul 08826, South Korea
E. Y. Lam
Department of Electrical and Electronic Engineering
The University of Hong Kong
Hong Kong 999077, China

The ORCID identification number(s) for the author(s) of this article can be found under https://doi.org/10.1002/apxr.202200118

DOI: 10.1002/apxr.202200118

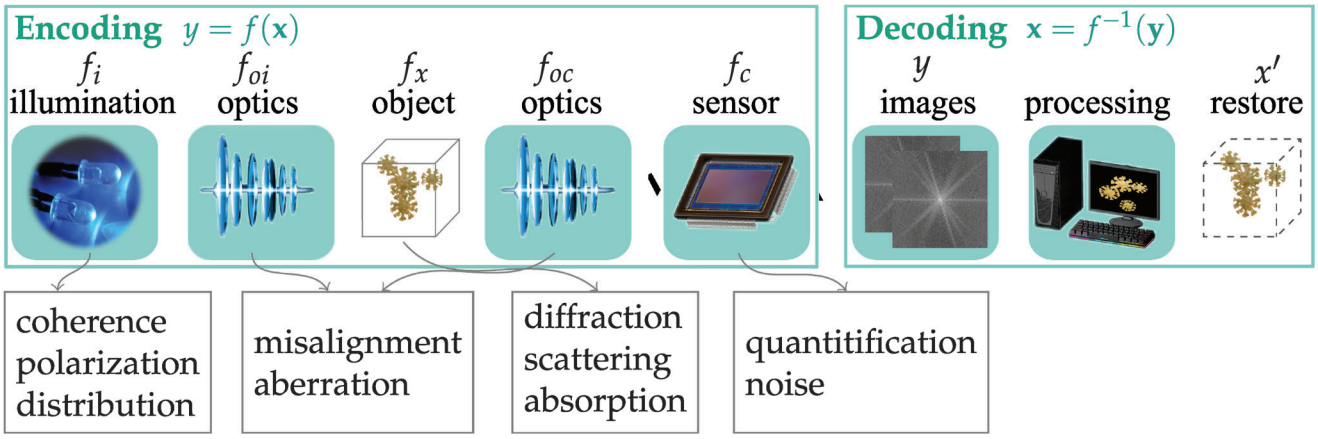


Figure 1. Schematic of a typical computational imaging system. The encoding process consists of light sources, delivery optics, light-object interaction, collection optics, and measurement sensors. The decoding process conducts numerical reconstruction of target physical quantities from measurements.

achieve scientific imaging from indirect measurements.[1] A typical computational imaging system mainly consists of two components: physical encoding and computational decoding, which is depicted schematically in **Figure 1**.

The encoding stage transfers the electromagnetic waves from an object into intensity measurement images. This involves how light travels through an entire optical system to reach electronic sensors and the mathematical formulation of the measurements in terms of physics, in which the light source, optical elements, light-wave interaction, and electronic signal transition all play a role. The decoding stage estimates the desired physical quantity using these measurements.[6] It differs from traditional digital image processing as it takes into account the underlying physics of the imaging system, requiring a precise model of the encoding process. We could simply model the encoding as $\mathbf{y} = f(\mathbf{x})$, where $\mathbf{x}$ is a vector representing the object, $f(\cdot)$ denotes the system transformation, and $\mathbf{y}$ is another vector representing the measurements, such as a collection of images.

The philosophy of computational imaging is to jointly design front-end encoding optics and post-detection decoding computation so as to balance optics and electronics processing capabilities.[1] This is typically employed to achieve the following goals:

- **Image reconstruction**, that is, restoring $\mathbf{x}$ from $\mathbf{y}$. This mainly focuses on the decoding process with (or without) prior knowledge of the encoding physics, to estimate the objects from the measurements, and fundamentally solve ill-posed inverse problems.[18,19] Phase retrieval and its variants are typical examples.[20–22] Two main techniques used to solve ill-posed problems are numerical optimization and neural networks. These methods incorporate prior information about the object, either analytically or based on empirical observations, to solve the problem. Numerical optimization usually starts with an analytical theoretical model f that perfectly fits the real imaging system and then optimizes to obtain $\mathbf{x}$ by minimizing a chosen error metric of difference between the numerical forward and real measurements.[23] Data-driven neural networks, on the other hand, typically aim to learn the implicit representation parameters by minimizing the loss functions from training datasets in order to obtain an inverse mapping from measurements to the signal, that is, $f^{-1}(\cdot) : \mathbf{y} \rightarrow \mathbf{x}$.[24–26]
- **Imaging system design**, that is, designing $f(\cdot)$ with constraints. This works on improving the existing imaging systems to achieve better performance or creating new imaging modalities to reduce measurement marginal costs, such as size, weight, or the total cost of a system. Examples include coded aperture for vision cameras[27]; structured illumination[28] and point-spread-function (PSF) engineering[29] in 3D imaging to tackle the dimension mismatch; coded apertures in X-ray/γ-ray imaging to improve the acquisition time[30]; lensless imaging[4,31] for achieving a compact form-factor of the system, etc.

Clearly, the two facets are fundamentally dependent on the encoding of the imaging systems, which must be meticulously modeled. However, the aforementioned encoding model typically oversimplifies the optics to a few objective parameters ignoring the wave nature of light, aberrations, and system sensitivities. This oversimplification diminishes the power of computational imaging, which may no longer be able to simulate the optical system realistically. To create a more realistic model, a deeper understanding of the individual components of the encoding optical system is necessary. The illumination system, which is made up of light sources and/or some optical elements, generates photons that travel either transmissively or reflectively through and interact with the target object. Afterward, the delivery and collection optics, as well as the electronic sensors, collect the object information encoded within these waves. Each of the sub-process has its own encoding function $f_*(\cdot, \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a collection of parameters. The light source, which could be a laser, laser diode, LED, X-ray, ambient light, or light source array, can be represented as a function $f_i(\cdot, \boldsymbol{\theta}_i)$ of wavelength, coherence, polarization, etc., and then determines the light-object interaction $f_x(\cdot)$ in conjunction with the illumination optics $f_{oi}(\cdot, \boldsymbol{\theta}_{oi})$ and the object physical properties. Possible light-object interaction models are thin transparency, Born or Rytov expansions, beam propagation, transmission matrix, etc. Both the illumination $f_{io}(\cdot, \boldsymbol{\theta}_{io})$ and collection optical systems $f_{co}(\cdot, \boldsymbol{\theta}_{co})$ could be functions of parameters related to apertures, lenses, polarizers, diffusers, spatial

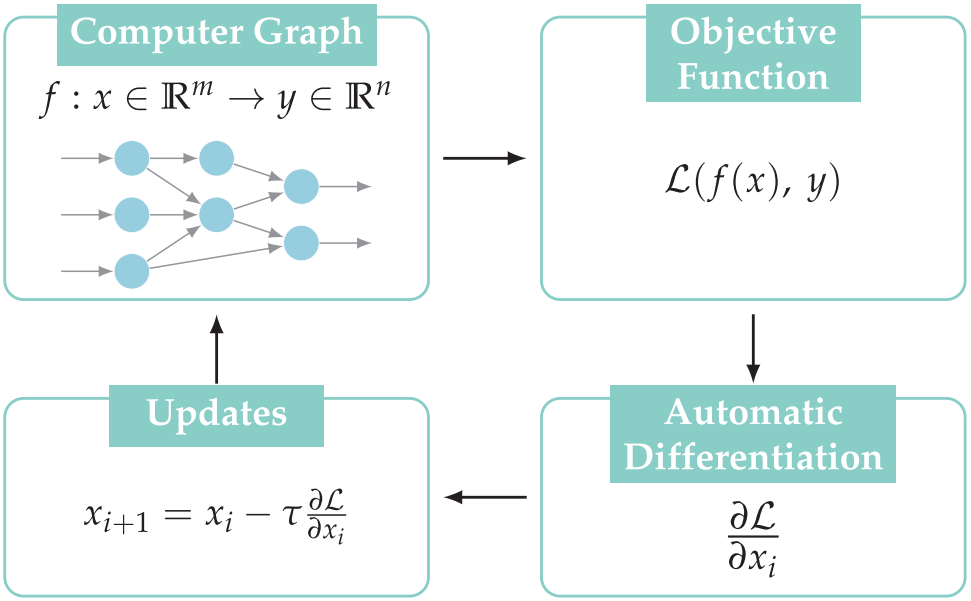


Figure 2. Schematic of generalized differentiable programming.

light modulators (SLM), and other elements. The sensor function $f_c(\cdot, \boldsymbol{\theta}_c)$, could be associated with sensor sampling, quantization, and photon-to-electron transduction accompanied by inevitable noise. The overall forward model of the system thus becomes a composition function:

$$\mathbf{y} = f(\mathbf{x}, \boldsymbol{\theta}) + n_{\mathcal{T}}, \qquad f = n_{\mathcal{P}} \circ f_c \circ f_{oc} \circ f_x \circ f_{oi} \circ f_i, \tag{1}$$

where $\circ$ is the function composition operator. $n_{\mathcal{P}}$ and $n_{\mathcal{T}}$ are signal-dependent photon noise and signal-independent thermal random noise, respectively.[23]

Given the above expression, the complexity of a realistic forward model in imaging systems stems from the contribution of each optical element as well as the uncertainty introduced by system imperfections and noise. It is important to acknowledge this complexity when developing image reconstruction algorithms or designing imaging systems.

2. Differentiable Programming

What is differentiable programming, and how does it exert such power? Based on various sources, we define it as[32,33]

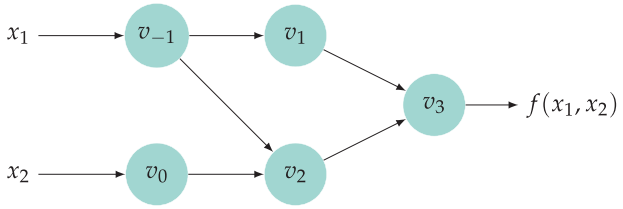
> *Differentiable programming* is a programming paradigm that creates software composed of *differentiable and parameterized building blocks* (a computer graph) that are executed via *automatic differentiation* and optimized in order to perform a specified task. These programs can also rewrite parts of themselves along a gradient.

As graphically depicted by **Figure 2**, a typical differentiable program consists of 1) a computer graph; 2) an objective function; 3) a calculation of the derivatives of the objective function with automatic differentiation; 4) a gradient-based optimization.

To execute automatic differentiation, functions are first converted into an acyclic directed **computer graph** in terms of elementary operations (primitives) with known derivatives to which the chain rule may be applied. A computer graph is a composition of parameterized, differentiable functions, methods, and models that deal with inputs and outputs, which contain the program's control flow and data structures. This form of computer graph illustrates the interdependence of all the parameters. It should be noted that neural networks are a type of specific computer graph with many separate elements (nodes). To differentiate with respect to input parameters, each node holds information that contributes to the derivative at that point, which is combined and propagated using the chain rule and the differentiation rules for each operation.

Automatic differentiation, also known as "algorithmic differentiation"[34] is a set of techniques for automatically evaluating the gradient of functions that are implemented as computer programs.[35] Simply put, automatic differentiation is an algorithmic way to use the chain rule of differentiation at the elementary operation level, with intermediate derivative values stored in memory. This is based on the fact that all numerical computations are ultimately composed of a finite number of primitive operations with known derivatives. Combining these derivatives of the constituent operations through the chain rule then yields the derivative of the overall composition.

Automatic differentiation can be calculated in two ways: forward[36] and backward (reverse)[37,38] automatic differentiation. These two methods can be differentiated based on the direction in which derivatives are computed when the chain rule is applied. Forward automatic differentiation calculates numerical derivatives by performing elementary derivative operations in parallel with the evaluation of the function, going from input to output. On the other hand, backward automatic differentiation uses an extension of the forward computational graph to find the gradient by traversing the graph in reverse, from output to input. **Figure 3** demonstrates the evaluation of the derivatives of the function $y = f(x_1, x_2) = \sin(x_1) + x_1 x_2$ at $(x_1, x_2) =$



(a) Computer graph.

Forward primal trace

v_{-1}	$= x_1$	$= 2$
v_0	$= x_2$	$= 1$
v_1	$= \sin v_{-1}$	$= \sin 2$
v_2	$= v_{-1} \times v_0$	$= 2 \times 1$
v_3	$= v_1 + v_2$	$= 0.909 + 2$
y	$= v_3$	$= 2.909$

(b) Forward primal trace

Forward derivative trace

$\dot{v}_{-1}$	$= \dot{x}_1$	$= 1$
$\dot{v}_0$	$= \dot{x}_2$	$= 0$
$\dot{v}_1$	$= \dot{v}_{-1} \times \cos(v_{-1})$	$= -0.416$
$\dot{v}_2$	$= \dot{v}_{-1} \times v_0 + \dot{v}_0 \times v_{-1}$	$= 1$
$\dot{v}_3$	$= \dot{v}_1 + \dot{v}_2$	$= 0.584$
$\dot{y}$	$= \dot{v}_3$	$= \mathbf{0.584}$

(c) Forward automatic differentiation

Reverse derivative trace

$\bar{x}_1$	$= \bar{v}_{-1}$		$= \mathbf{0.584}$
$\bar{x}_2$	$= \bar{v}_0$		$= \mathbf{2}$
$\bar{v}_{-1}$	$= \bar{v}_1 \frac{\partial v_1}{\partial v_{-1}} + \bar{v}_2 \frac{\partial v_2}{\partial v_{-1}}$	$= \bar{v}_1 \cos(v_{-1}) + \bar{v}_0$	$= 0.584$
$\bar{v}_0$	$= \bar{v}_2 \frac{\partial v_2}{\partial v_0}$	$= \bar{v}_2 v_{-1}$	$= 2$
$\bar{v}_1$	$= \bar{v}_3 \frac{\partial v_3}{\partial v_1}$	$= \bar{v}_3 \times 1$	$= 1$
$\bar{v}_2$	$= \bar{v}_3 \frac{\partial v_3}{\partial v_2}$	$= \bar{v}_3 \times 1$	$= 1$
$\bar{v}_3$	$= \bar{y}$	$= 1$	

(d) Backward automatic differentiation

Figure 3. An example of automatic differentiation with $y = f(x_1, x_2) = \sin(x_1) + x_1 x_2$ evaluated at $(x_1, x_2) = (2, 1)$.

(2, 1) using forward and backward automatic differentiation, respectively. Assuming that the v_n are elementary functions with known individual derivatives, the computer graph in Figure 3a represents the function f as a composition of $v_3 \bigcirc v_2 \bigcirc v_1 \bigcirc v_0 \bigcirc v_{-1}$. At the given point $(x_1, x_2) = (2, 1)$, the value of f can be computed by following the sequence of evaluations shown in the primal trace of Figure 3b. In forward mode automatic differentiation, the primal evaluation from Figure 3b is augmented with derivative operations, represented as $\dot{v}_i = \frac{\partial v_i}{\partial x_i}$ in Figure 3c. Each line in the forward differentiation trace complements its corresponding line in the original primal trace. By starting with the initialization of $\dot{x}_1 = 1$ and setting the rest to zero, evaluation of $\frac{\partial y}{\partial x_1}$ can be conducted through the primal trace with the corresponding derivatives from top to bottom. In backward mode automatic differentiation, the adjoint operations $\bar{v}_i = \bar{y} = \frac{\partial y_i}{\partial v_i}$ are evaluated in reverse order, starting from the bottom of the forward primal trace evaluation and working towards the top, as demonstrated in Figure 3d. This process begins with the initialization of $\bar{v}_3 = \bar{y} = \frac{\partial y}{\partial y} = 1$, both $\frac{\partial y}{\partial x_1}$ and $\frac{\partial y}{\partial x_2}$ are computed in the same reverse pass.

The above example illustrates that forward automatic differentiation is an efficient and straightforward method for evaluating the derivatives of functions with a small number of inputs, f: $R^n \to R^m$ where n is small. However, its computational complexity increases with the number of inputs, n. On the other hand, for functions with a large number of inputs, $n \gg m$, backward automatic differentiation is a more efficient option, but requires more storage. This example also demonstrates the potential for expanding the calculation to include additional nodes, variables, and operations. When classical control flow, such as an if statement, is present, the computational graph branches and gradients are only computed in the applicable branch. As a result, it is possible to compute derivatives not only for mathematical functions, but for general-purpose computer code that includes control flow, loops, recursions, and so on.

3. Differentiable Imaging

Despite recent advancements in image reconstruction algorithms[6,18] and imaging system design,[4,39] the potential of computational imaging remains untapped by the existing challenges in the field. To fully realize its capabilities, it is necessary to tackle these challenges through the adaptation of differentiable programming. In the following, we revisit the challenges in computational imaging, examine how differentiable programming can overcome these obstacles, and highlight two promising directions for resolution.

3.1. Challenges in Computational Imaging

Computational imaging faces two major interrelated challenges: (1) the uncertainty in the physical model and (2) the difficulty in implementing a joint-design philosophy that balances computational techniques with practicality.[1] The uncertainty in the physical model raises the complexity of system modeling and necessitates the use of more advanced computational techniques. On the other hand, the requirement for precise and stable systems in joint-design philosophy makes it challenging to implement in practice.

Uncertainty of the Physical Model: The main challenge in inverse image reconstruction in computational imaging is uncertainty in the forward encoding process, which arises from system imperfections and noise.[23] Imperfect optical devices and

elements, misalignment due to practitioner skills and imaging conditions, and device- and content-dependent noise are some of the factors that contribute to this uncertainty. Accounting for these uncertainties in the inverse problem solver and system design is crucial but remains an under-explored area.[26,40,41] However, accurately characterizing the encoding system is often difficult or impractical, and researchers must strike a balance between more realistic models and simpler, computationally feasible models. Additionally, the sensitivity to noise and the level of artifacts introduced by post-processing are practical considerations. To address these challenges, there is a need to concurrently model uncertainties and design the subsequent decoding algorithms.

Practice of Joint-Design Philosophy: The philosophy behind computational imaging aims to strike a balance between the processing capabilities of optics and electronics through concurrent and joint design. This has numerous benefits for computational imaging as demonstrated in various studies.[42–44] Joint design can be accomplished through various means, including computational joint optimization and encoding-decoding joint design (or in-loop hardware-software design). Although in-loop hardware-software design is comparatively sluggish compared to computational practices, the primary challenge in computational imaging lies in computational techniques. Particularly, there are primarily two approaches for solving inverse problems[45]: classical optimization and neural network-based algorithms, each with its own advantages and disadvantages.

- Classical optimization algorithms. The ill-posed nature of most computational imaging problems leads to the use of regularization methods in classical optimization techniques.[46] These methods typically consist of a data term and a regularization term, where the data term represents the underlying physics model and the regularization term incorporates problem constraints. A wide range of methods has been proposed to choose these two components and implement appropriate optimization algorithms. Some regularizers, such as the ℓ_1 norm,[47] involve non-differentiable metrics, which necessitate the use of proximal gradients for iterative optimization.[48] Common algorithms include least absolute shrinkage and selection operator (Lasso),[49] iterative shrinkage and threshold algorithm (ISTA)[50] and its variants fast ISTA (FISTA)[51] and TwIST,[52] alternating direction method of multipliers (ADMM),[53,54] and Adam.[55] However, many of these algorithms require a simplified gradient expression of the metric function, making it difficult to tackle complex inverse imaging problems that arise from measurement noise, nonuniqueness, and high dimensions of the solution. In some cases, it may even be impossible to apply these algorithms to these challenging inverse imaging problems.[44,56–58]
- Neural network algorithms. The use of neural networks has been successful in many research fields,[59–61] but it is not without its challenges. In the field of computational imaging, one such challenge is the difficulty in preparing ground-truth data.[24,62–64] Using data-based neural networks without knowledge of the forward physical model can result in predictions that are unexplainable, lack verification and reproducibility, and are not adaptable to imaging system uncertainties. To overcome these limitations, researchers have explored physics-informed methods that incorporate physical knowledge and priors into neural networks,[65] such as recurrent dictionary training, cascaded unroll networks,[58,66] and single-pass untrained networks.[67,68] However, this incorporation reduces the efficiency of deep learning.

Neither classical optimization nor neural networks alone are capable of effectively handling complex models with uncertainties, which is why current computational imaging techniques tend to separate optics and computation rather than allow them to interact. Although some techniques utilize joint optimization approaches,[43,44] in-loop hardware-software co-design is rarely practiced.

However, computational imaging strives for a greater understanding of the world through the improvement of both image reconstruction and system design. This requires advanced modeling of physical systems and the development of computational techniques that can handle complex models and dynamic optical systems. To achieve this, computational techniques must be expressive and capable of handling uncertainties. With these challenges in mind, differentiable programming emerges as the most suitable technique to tackle a considerable part of the difficulties in computational imaging. This is because differentiable programming embodies the characteristics necessary to address the aforementioned challenges. By outlining these characteristics, we can demonstrate how differentiable programming can be applied in computational imaging.

3.2. Characteristics of Differentiable Programming

Recall that uncertainty modeling and a lack of computational techniques to handle complex imaging models are the two main challenges in computational imaging. Differentiable programming could overcome these because of the following characteristics.

- **Work on generic computer programs**. Firstly, differentiable programming automates the differentiation process in machine precision and can be applied to any regular computer program that employs control flow mechanisms (branches, conditional loops, and recursion), allowing for greater flexibility in selecting the data term, regularization term, and optimizer in classical optimization, without requiring manual calculation of derivatives. This can enable the use of different data terms, such as least-square error or Poisson maximum likelihood, depending on the noise distribution of the measurements.[69] Additionally, uncertainties can be incorporated into the forward imaging model with less simplification, and physics-based and data-based priors can be used as regularizers.[51–53] Second, automatic differentiation also provides more options for problem modeling strategies.[21,70] For example, in 3D surface imaging, the height can be converted from the phase of the wavefront that is measured using interferometry, and this inverse problem can be simplified by treating the height as the optimization target and defining the function's domain to be real. Overall, the flexibility offered by automatic differentiation allows for a wider range of optimization strategies and imaging capabilities.

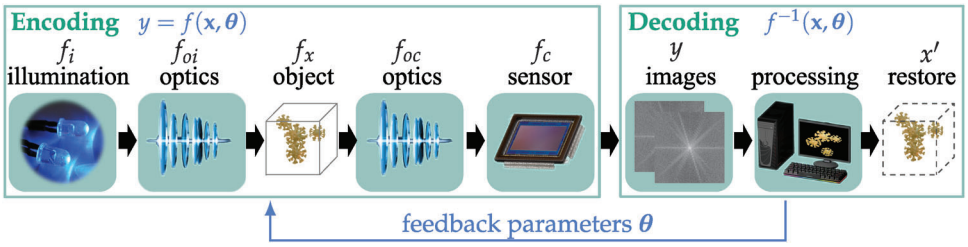


Figure 4. Schematic of a differentiable imaging system. Both the encoding and decoding processes involve additional parameter θ of the imaging system, which can also be used for feedback to improve the imaging systems.

- **Mix-and-match of differentiable building blocks**. Differentiable programming enables the integration of prior knowledge at multiple levels of abstraction by composing differentiable building blocks. This mix-and-match capability allows for a deeper level of integration, unlike modular system design that can limit the potential for joint optimization and lead to mass manufacturing.[1] The ability to mix-and-match different mathematical formalisms facilitates the joint design of both inverse imaging problems and imaging systems, making it possible to address challenges at both small and large scales.
- **Incorporate off-the-shelf machine learning frameworks**. The use of automatic differentiation in machine learning has enabled the development of various methods that rely on gradient information for model training.[9–11,71] This has made it possible for researchers to quickly and easily combine off-the-shelf network building blocks with numerical optimization using just a few lines of code. By combining existing computational techniques, it becomes possible to leverage the efficiency of numerical optimization and the expressiveness of neural networks in the field of computational imaging.

In summary, differentiable programming offers a solution to the challenges in computational imaging by enabling the modeling of uncertainty, providing a flexible approach for large-scale inverse imaging problem solving through the combination of modular techniques, and integrating pre-existing machine learning frameworks. Moreover, its ability to dynamically modify computation graphs during optimization allows for the design of advanced closed-loop hardware-software imaging systems.

Thanks to the availability of industrial-grade software frameworks,[9–11,71] differentiable programming in computational imaging has seen rapid development. It solves inverse imaging problems through either differentiable optimization or differentiable neural networks. Despite not being widely recognized, differentiable imaging can be defined as:

> *Differentiable imaging* is a computational imaging paradigm where certain aspects of the problem are parameterized as differentiable, irrespective of whether the inverse problems are tackled by numerical optimization, neural networks, or a combination of the two.

As depicted in **Figure 4**, differentiable imaging models encompass extra system parameters in comparison to the conventional computational imaging presented in Figure 1 with either numerical optimization or neural networks. These parameters in the imaging model enhance the image reconstruction, or can be leveraged for further improvement or design of imaging systems.

3.3. Current Status of Differentiable Imaging

The emergence of differentiable imaging in both modeling and design in computational imaging, as illustrated in **Figure 5**, has the potential to transform the approach to solving complex imaging problems. This innovative concept offers greater efficiency, flexibility, and expressiveness in addressing various limitations and challenges faced by various computational imaging systems. In the following section, we provide a succinct overview of how differentiable imaging can help resolve these issues.

Dealing with Uncertainty and Imperfections in Imaging Systems: As described in Section 1, uncertainty or misalignment exists everywhere in a real optical system. In a majority of imaging systems, the forward model is distance-dependent, and the target location must be determined prior to inverse image reconstruction; otherwise, the reconstruction algorithm's reliability may be compromised. Auto-focusing, for instance, must precede inverse imaging.[24] In the case of holographic imaging for the reconstruction of complex fields, auto-focusing techniques are not always effective. This problem could be solved by parameterizing the object's position along with the target complex field and optimizing them together using differentiable programming.[72] Similarly, in electron microscopy, phase retrieval, and defocus estimation can be performed concurrently to produce stable and high-quality imaging.[17] In all-optical computing and diffractive neural networks, mechanical misalignment and fabrication inaccuracies and imperfections are always present in the physical implementation and affect wave propagation. By parameterizing and training neural networks with differentiable parameters that represent the translation error or imperfections in the fabrication, researchers can improve the robustness and extend the layers of the diffractive networks.[73–76]

Modeling Complex Imaging Systems: Many optical imaging systems involve complex physical phenomena, necessitating the simplification of the forward model in inverse imaging. Differentiable imaging enables more precise encoding models, allowing for imaging capabilities beyond those that are currently inaccessible. Despite the accurate modeling in the inline holography to achieve complex field imaging capability,[72] differentiable imaging could also be used to address more complex physical phenomena, such as multiple scattering, which impedes the use of conventional numerical imaging reconstruction methods. For example, in X-ray nanotomography, differentiable imaging allows imaging beyond the depth of focus limit.[13] In other imaging modalities, the complexity of encoding results from the need to retrieve additional information. In ptychography, for example, probe reconstruction is also required, which makes inverse

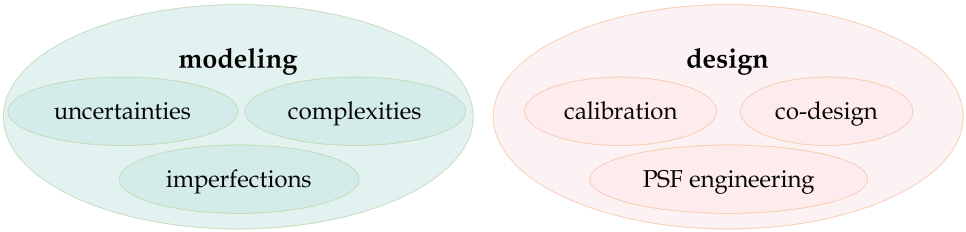


Figure 5. Current development of differentiable imaging.

ptychographic imaging challenging. By introducing differentiable programming, not only can the probe and the object be estimated simultaneously,[12] but also a more accurate and alterable forward model with more freedom to choose the error metric and priori information is possible, enabling higher-quality imaging in terms of speed, accuracy, and robustness.[77,78]

Co-Design or Re-Configurable Systems: In PSF engineering for enhancing imaging quality,[79–81] a co-design for calibrating real optics system setup and engineering the point spread function in a microscope was proposed by representing the issue as an inverse problem and constructing a differentiable wave optics simulator as a composition of trainable modules.[82] In terms of lens design, a broad range of imaging applications can be realized by providing derivative insights into the lens design pipeline to chain variable parameters and their gradients to an error metric via differential ray tracing.[83–85] In hologram encoding, the camera-in-loop paradigm has been used to achieve system calibration in a holographic display by introducing system imperfection parameters into neural networks.[86] Differentiable microscopy, which employs implements trainable optical elements at key locations on the optical path via differentiable programming, has the potential to create new interpretable microscope architectures or new optical systems, leading to unconventional and better optical designs.[15]

3.4. Opportunities and Challenges of Differentiable Imaging

Despite the progress made so far, differentiable imaging has yet to realize its full potential. With the availability of large datasets, increased computational power, and advanced instrumentation, differentiable imaging holds great promise in providing new and powerful tools for both improving image reconstruction and designing innovative imaging systems. Two such directions are outlined below.

Black-Box to Gray-Box Through Hybrid Neural Networks: The rapid growth in large imaging datasets over the past decade has not been matched by a corresponding advance in our understanding of imaging systems. This gap is partly due to the lack of analytical frameworks that can effectively integrate various data types, incorporate complex prior knowledge of physics, and extract meaningful insights from the resulting information. Many entities in optical imaging systems, such as dynamical reactions, are structurally richer than the data types used in most current machine learning research, necessitating more algorithmic development of differentiable frameworks. The recent surge in large imaging datasets has made it increasingly difficult to effectively analyze and understand these systems using traditional approaches. Structurally rich entities in optical imaging, such as dynamical reactions, demand the development of algorithmic frameworks to handle the complexity of the data. Machine learning methods currently face two major challenges when dealing with this type of data: creating interpretable models and incorporating prior knowledge of physics. One approach to addressing these challenges is through the integration of physics and data prior to the training pipeline. The utilization of backpropagation automatic differentiation, a foundation of popular machine learning frameworks like TensorFlow and PyTorch, can be used to separate neural networks into trainable and fixed parts. This allows for the explicit modeling of optimization components and learning the rest.[87,88] There is ongoing research in this area, such as developing optimization-based primitives, that can connect machine learning and numerical optimization.[89,90] By bridging the gap between machine learning and numerical optimization, differentiable imaging has the potential to move from black-box to gray-box models. This shift would improve data efficiency, increase interpretability, and facilitate gray-box verification, as it utilizes specialized sub-modules rather than purely fully-connected architectures.

Differentiable Imaging as a Framework for Designing All-In-Loop Imaging Systems: Keep in mind that "balancing" is fundamentally an optimization process, and computational imaging aims to achieve this by jointly designing the front-end encoding optics and post-detection decoding computation. Unlike image reconstruction, the field of imaging system design has remained largely unexplored. This is due to the challenges of modeling an optical system with uncertainties and constructing a dynamic optimization framework for all-in-loop hardware-software co-design. Optical systems, which are comprised of optical elements, can be modeled with modular components, much like computer graphs in differentiable programming, to solve complex problems. Differentiable programming's modeling and optimization capabilities hold great potential for the development of novel imaging systems through a modular, plug-and-play approach. To achieve this, it's necessary to properly formulate imaging systems and problems and to mathematically encode the correct priors in optimization, while taking into account the feedback for system reconfiguration.[91–93]

While differentiable imaging holds great potential, its implementation must also consider the limitations inherent to differentiable programming.

- **Modeling requires physical and mathematical knowledge**. Implementing differentiable imaging can be challenging as it demands a combination of physical and mathematical expertise to properly formulate the problem, encode relevant priors

mathematically, and determine which parts of the code need differentiation.[94] Additionally, for imaging applications that involve interactions between light and matter, such as biology or material science, domain-specific knowledge may also be necessary.[16,17]

- **Effective modeling requires a deep understanding of the differentiable programming language**. Because the machine must calculate the derivative for each specified part of the code, the computational cost can be substantial. As there are usually several ways to differentiate code, it can be unclear how effective differentiable programming will be in practice. Hence, choosing the right type of automatic differentiation (forward, backward, or a combination) to maximize computer efficiency is crucial.[34]
- **Development of differentiable primitives is required**. It can be challenging to implement differentiable imaging due to the inherent nature of some components in the model. The requirement for all model components to be differentiable conflicts with the discreteness of digital images, mathematical expressions,[95] computational data structures,[96] and processes.[97–99] Additionally, the use of complex numbers in optical imaging that deals with electromagnetic waves results in the need for complex differentiation, making the development of differentiable primitives a common challenge. These limitations can make it difficult to effectively implement differentiable programming in practice.

4. Conclusion

In conclusion, the progress of computational imaging has faced numerous computational limitations that have hindered its advancement. Both classical optimization and deep learning algorithms have limitations in processing complex experimental data effectively. However, differentiable programming provides a promising solution by bridging the gap between model-based optimization and neural network algorithms and creating interpretable model-based neural networks. As a relatively new field, differentiable imaging has the potential to transform the field of computational imaging. By integrating physics into the model and effectively modeling phenomena of varying complexity, it can address the limitations posed by sparse, incomplete, and noisy data. With its ability to solve long-standing challenges in the field, differentiable imaging holds the promise of being a significant catalyst for change in computational imaging.

Acknowledgements

The authors express gratitude to Dr. Congli Wang (UC Berkeley) for his valuable input on an early draft of this paper, Dr. Xin Li (Beijing Institute of Technology) for his insightful feedback and discussions, and Prof. Sigurdur Thoroddsen (King Abdullah University of Science and Technology) for his English editing assistance.

Conflict of Interest

The authors declare no conflict of interest.

Keywords

computational imaging, differentiable imaging, differentiable programming, neural networks

Received: December 20, 2022
Revised: February 1, 2023
Published online: March 23, 2023

[1] J. N. Mait, G. W. Euliss, R. A. Athale, *Adv. Opt. Photonics* **2018**, *10*, 409.
[2] L. Waller, L. Tian, *Nature* **2015**, *523*, 416.
[3] K. C. Zhou, R. P. McNabb, R. Qian, S. Degan, A.-H. Dhalla, S. Farsiu, J. A. Izatt, *Optica* **2022**, *9*, 593.
[4] V. Boominathan, J. T. Robinson, L. Waller, A. Veeraraghavan, *Optica* **2021**, *9*, 1.
[5] A. Ozcan, E. McLeod, *Annu. Rev. Biomed. Eng.* **2016**, *18*, 77.
[6] G. Barbastathis, A. Ozcan, G. Situ, *Optica* **2019**, *6*, 921.
[7] J. Wu, H. Zhang, W. Zhang, G. Jin, L. Cao, G. Barbastathis, *Light: Sci. Appl.* **2020**, *9*, 1.
[8] C. Olah, Neural Networks, Types, and Functional Programming, **2015**, http://colah.github.io/posts/2015-09-NN-Types-FP/.
[9] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, et al., *TensorFlow: Large-scale machine learning on heterogeneous systems*, https://www.tensorflow.org, **2015**, Software available from tensorflow.org.
[10] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, S. Chintala, *Pytorch: An imperative style, high-performance deep learning library, in Advances in Neural Information Processing Systems*, (Eds.: H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, R. Garnett), Vol. *32*, Curran Associates, New York **2019**, pp. 8024–8035.
[11] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, Q. Zhang, JAX: composable transformations of Python+NumPy programs, http://github.com/google/jax, **2018**.
[12] Y. S. Nashed, T. Peterka, J. Deng, C. Jacobsen, *Procedia Comput. Sci.* **2017**, *108*, 404.
[13] M. Du, Y. S. G. Nashed, S. Kandel, D. Gürsoy, C. Jacobsen, *Sci. Adv.* **2020**, *6*, eaay3700.
[14] I. Vishniakou, J. D. Seelig, *Opt. Express* **2021**, *29*, 21418.
[15] K. Herath, U. Haputhanthri, R. Hettiarachchi, H. Kariyawasam, A. Ahmad, B. Ahluwalia, C. Edussooriya, D. Wadduwage, *Differentiable microscopy designs an all optical quantitative phase microscope* **2022**.
[16] N. Nguyen, F. Liang, D. Engel, C. Bohak, P. Wonka, T. Ropinski, I. Viola, *Differentiable electron microscopy simulation: Methods and applications for visualization* **2022**.
[17] T. Zhou, M. Cherukara, C. Phatak, *npj Comput. Mater.* **2021**, *7*, 1.
[18] K. Dillon, Y. Fainman, Y.-P. Wang, *J. Electron. Imaging* **2016**, *25*, 053016.
[19] A. Dave, A. K. Vadathya, R. Subramanyam, R. Baburajan, K. Mitra, *IEEE Trans. Comput. Imaging* **2019**, *5*, 37.
[20] F. Momey, L. Denis, T. Olivier, C. Fournier, *J. Opt. Soc. Am. A* **2019**, *36*, D62.
[21] M. Chen, D. Ren, H.-Y. Liu, S. Chowdhury, L. Waller, *Optica* **2020**, *7*, 394.

[22] Z. Huang, P. Memmolo, P. Ferraro, L. Cao, *PhotoniX* **2022**, *3*, 1.
[23] M. Burger, H. Dirks, J. Müller, in Large Scale Inverse Problems, DE GRUYTER, **2013**, pp. 135–180.
[24] Z. Ren, Z. Xu, E. Y. Lam, *Optica* **2018**, *5*, 337.
[25] Z. Ren, Z. Xu, E. Y. Lam, *Adv. Photonics* **2019**, *1*, 1.
[26] T. Zeng, E. Y. Lam, *IEEE Trans. Comput. Imaging* **2021**, *7*, 1080.
[27] A. Levin, R. Fergus, F. Durand, W. T. Freeman, *ACM Trans. Graphics* **2007**, 70.
[28] G. Zheng, R. Horstmeyer, C. Yang, *Nat. Photonics* **2013**, *7*, 739.
[29] A. Greengard, Y. Y. Schechner, R. Piestun, *Opt. Lett.* **2006**, *31*, 181.
[30] M. J. Cieślak, K. A. Gamage, R. Glover, *Radiat. Meas.* **2016**, *92*, 59.
[31] J. Wu, L. Cao, G. Barbastathis, *Opt. Lett.* **2020**, *46*, 130.
[32] A. Hernández, G. Millerioux, J. M. Amigó, *Differentiable programming: Generalization, characterization and limitations of deep learning* **2022**.
[33] T.-M. Li, M. Gharbi, A. Adams, F. Durand, J. Ragan-Kelley, *ACM Trans. Graphics* **2018**, *37*, 1.
[34] A. G. Baydin, B. A. Pearlmutter, A. A. Radul, J. M. Siskind, *JMLR* **2017**, *18*, 5595.
[35] J. F. Nolan, Master's Thesis, Massachusetts Institute of Technology, **1953**.
[36] R. E. Wengert, *Commun. ACM* **1964**, *7*, 463.
[37] S. Linnainmaa, Master's Thesis, University of Helsinki.
[38] S. Linnainmaa, *BIT* **1976**, *16*, 146.
[39] W. Li, J. Qi, A. Sihvola, *Adv. Opt. Mater.* **2020**, *8*, 2001000.
[40] D. Gilton, G. Ongie, R. Willett, *IEEE Trans. Comput. Imaging* **2021**, *7*, 661.
[41] A. Zhou, W. Wang, N. Chen, E. Y. Lam, B. Lee, G. Situ, *Opt. Express* **2018**, *26*, 23661.
[42] H. W. Babcock, *Publ. Astron. Soc. Pac.* **1953**, *65*, 229.
[43] G. Zang, R. Idoughi, R. Tao, G. Lubineau, P. Wonka, W. Heidrich, *ACM Trans. Graphics* **2018**, *37*, 1.
[44] N. Chen, C. Wang, W. Heidrich, *Laser Photonics Rev.* **2021**, *15*, 2100008.
[45] A. Mohammad-Djafari, *Regularization, bayesian inference and machine learning methods for inverse problems*†, **2021**.
[46] C. R. Vogel, *Computational Methods for Inverse Problems*, Society for Industrial and Applied Mathematics, **2002**.
[47] J. Bect, L. Blanc-Féraud, G. Aubert, A. Chambolle, in *Lecture Notes in Computer Science*, Springer Berlin Heidelberg **2004**.
[48] J. Eckstein, D. P. Bertsekas, *Math. Program.* **1992**, *55*, 293.
[49] R. Tibshirani, *J. R. Stat. Soc. Sec. B* **1996**, *58*, 267.
[50] I. Daubechies, M. Defrise, C. D. Mol, *Commun. Pure Appl. Math.* **2004**, *57*, 1413.
[51] A. Beck, M. Teboulle, *SIAM J. Imag. Sci.* **2009**, *2*, 183.
[52] J. Bioucas-Dias, M. Figueiredo, *IEEE Trans. Image Process.* **2007**, *16*, 2992.
[53] S. Boyd, *Found. Trends Mach. Learn.* **2010**, *3*, 1.
[54] L. Song, Z. Ge, E. Y. Lam, *IEEE Trans. Image Process.* **2022**, *31*, 3295.
[55] D. P. Kingma, J. Ba, in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, **2014**.
[56] D. J. Brady, K. Choi, D. L. Marks, R. Horisaki, S. Lim, *Opt. Express* **2009**, *17*, 13040.
[57] W. Zhang, L. Cao, D. J. Brady, H. Zhang, J. Cang, H. Zhang, G. Jin, *Phys. Rev. Lett.* **2018**, *121*, 093902.
[58] N. Chen, C. Wang, W. Heidrich, *IEEE Trans. Comput. Imaging* **2021**, *7*, 288.
[59] K. Gregor, Y. LeCun, in *Proc. 27th Int. Conf. Mach. Learn.*, ICML'10. Omnipress, Madison, WI, USA, ISBN 9781605589077, **2010**, pp. 399-406.
[60] A. Mousavi, A. B. Patel, R. G. Baraniuk, in *2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, IEEE **2015**.
[61] A. Mousavi, R. G. Baraniuk, in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE **2017**.
[62] A. Sinha, J. Lee, S. Li, G. Barbastathis, *Optica* **2017**, *4*, 1117.
[63] M. Lyu, W. Wang, H. Wang, H. Wang, G. Li, N. Chen, G. Situ, *Sci. Rep.* **2017**, *7*, 1.
[64] Y. Rivenson, Y. Zhang, H. Günaydın, D. Teng, A. Ozcan, *Light Sci. Appl.* **2018**, *7*, 17141.
[65] T. Zeng, Y. Zhu, E. Y. Lam, *Opt. Express* **2021**, *29*, 40572.
[66] V. Monga, Y. Li, Y. C. Eldar, *IEEE Signal Process Mag.* **2021**, *38*, 18.
[67] V. Lempitsky, A. Vedaldi, D. Ulyanov, in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, IEEE **2018**.
[68] A. Qayyum, I. Ilahi, F. Shamshad, F. Boussaid, M. Bennamoun, J. Qadir, *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, 1.
[69] M. Du, D. Gürsoy, C. Jacobsen, *J. Appl. Crystallogr.* **2020**, *53*, 748.
[70] H. Wang, W. Tahir, J. Zhu, L. Tian, *Opt. Express* **2021**, *29*, 17159.
[71] J. Bezanson, A. Edelman, S. Karpinski, V. B. Shah, *SIAM Review* **2017**, *59*, 65.
[72] N. Chen, C. Wang, W. Heidrich, Differentiable holography, https://repository.kaust.edu.sa/handle/10754/679652, **2022**.
[73] D. Mengu, Y. Zhao, N. T. Yardimci, Y. Rivenson, M. Jarrahi, A. Ozcan, *Nanophotonics* **2020**, *9*, 4207.
[74] T. Zhou, X. Lin, J. Wu, Y. Chen, H. Xie, Y. Li, J. Fan, H. Wu, L. Fang, Q. Dai, *Nat. Photonics* **2021**, *15*, 367.
[75] T. Zhou, L. Fang, T. Yan, J. Wu, Y. Li, J. Fan, H. Wu, X. Lin, Q. Dai, *Photonics Res.* **2020**, *8*, 940.
[76] J. Shi, M. Chen, D. Wei, C. Hu, J. Luo, H. Wang, X. Zhang, C. Xie, *Opt. Express* **2020**, *28*, 37686.
[77] S. Ghosh, Y. S. G. Nashed, O. Cossairt, A. Katsaggelos, in *2018 IEEE International Conference on Computational Photography (ICCP)*, IEEE **2018**.
[78] S. Kandel, S. Maddali, M. Allain, S. O. Hruszkewycz, C. Jacobsen, Y. S. G. Nashed, *Opt. Express* **2019**, *27*, 18653.
[79] A. Klauss, F. Conrad, C. Hille, *Sci. Rep.* **2017**, *7*, 1.
[80] B. Shuang, W. Wang, H. Shen, L. J. Tauzin, C. Flatebo, J. Chen, N. A. Moringo, L. D. C. Bishop, K. F. Kelly, C. F. Landes, *Sci. Rep.* **2016**, *6*, 1.
[81] T. Zhao, T. Mauger, G. Li, *Biomedical Opt. Express* **2013**, *4*, 1464.
[82] J. Page, P. Favaros, in 27th IEEE International Conference on Image Processing (ICIP), IEEE, **2020**.
[83] C. Wang, N. Chen, W. Heidrich, *Opt. Express* **2021**, *29*, 30284.
[84] C. Wang, N. Chen, W. Heidrich, *IEEE Trans. Comput. Imaging* **2022**, *8*, 905.
[85] M. Dufraisse, P. Trouvé-Peloux, J.-B. Volatier, F. Champagnat, in *Unconventional Optical Imaging III*, (Eds.: M. P. Georges, G. Popescu, N. Verrier), Vol. *12136*, SPIE, **2022**, pp. 164–173.
[86] Y. Peng, S. Choi, N. Padmanaban, G. Wetzstein, *ACM Trans. Graphics* **2020**, *39*, 1.
[87] M. Thayaparan, M. Valentino, D. Ferreira, J. Rozanova, A. Freitas, *TACL* **2022**, *10*, 1103.
[88] B. Amos, J. Z. Kolter, in Proceedings of the 34th International Conference on Machine Learning, volume 70 of *Proceedings of Machine Learning Research*, PMLR, **2017**, pp. 136–145.
[89] A. Agrawal, S. Barratt, S. Boyd, *IEEE/CAA J. Autom. Sinica* **2021**, *8*, 1355.
[90] M. AlQuraishi, P. K. Sorger, *Nat. Methods* **2021**, *18*, 1169.
[91] H.-Y. Li, H.-T. Zhao, M.-L. Wei, H.-X. Ruan, Y. Shuang, T. J. Cui, P. del Hougne, L. Li, *Patterns* **2020**, *1*, 100006.
[92] Z. Wang, H. Zhang, H. Zhao, T. J. Cui, L. Li, *Nanophotonics* **2022**, *11*, 2011.
[93] L. Li, H. Zhao, C. Liu, L. Li, T. J. Cui, *eLight* **2022**, *2*, 1.
[94] C. Elliott, *Proc. ACM Trans. Program. Lang.* **2018**, *2*, ICFP, 1.
[95] G. Lample, F. Charton, in *International Conference on Learning Representations* **2019**.

[96] E. Grefenstette, K. M. Hermann, M. Suleyman, P. Blunsom, in *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2, NIPS'15*. MIT Press, Cambridge, MA, USA **2015**, pp. 1828–1836.
[97] A. Grover, E. Wang, A. Zweig, S. Ermon, in *International Conference on Learning Representations*, **2019**.
[98] A. Graves, *Adaptive computation time for recurrent neural networks* **2016**.
[99] A. Trask, F. Hill, S. E. Reed, J. Rae, C. Dyer, P. Blunsom, in *Advances in Neural Information Processing Systems*, (Eds.: S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, R. Garnett), vol. *31*, Curran Associates, Inc., **2018**.

Ni Chen obtained her B.S. degree in software engineering from Harbin Institute of Technology, China, in 2008 and her Ph.D. degree in electrical engineering from Seoul National University, South Korea, in 2014. She is presently an Associate Research Professor at the Wyant College of Optical Sciences in the University of Arizona, where she specializes in differentiable computational imaging for advanced optical imaging. Chen's primary focus of research is computational imaging, which encompasses 3D and 4D imaging, as well as 3D display technologies.

Liangcai Cao received his B.S. degree from the Department of Physics at Harbin Institute of Technology, China, in 1999 and his Ph.D. degree from the Department of Precision Instruments at Tsinghua University, China, in 2005. Currently, he serves as a Professor at the Department of Precision Instruments at Tsinghua University. Cao has published over 100 articles in peer-reviewed journals and conference proceedings. He has been recognized as a fellow of both OSA and SPIE. Cao's research interests are focused on imaging, storage, and display technology based on volume holography.

Ting-Chung Poon received his Ph.D. degree in electrical engineering from the University of Iowa in 1982. He is currently a Professor of Electrical and Computer Engineering at Virginia Tech, USA. He is a Fellow of IOP, OSA, and SPIE and was awarded the 2016 Dennis Gabor Award. He serves as a Guest Editor/Associate Editor/Editor for the IEEE Transactions on Industrial Informatics and Applied Sciences. He is the founding chair of the OSA topical meeting Digital Holography and 3-D Imaging. In addition, he has served as chair of the OSA Emmett N. Leith Medal Committee and as a member of the OSA Joseph Fraunhofer Award/Robert Burley Prize Committee.

Byoungho Lee obtained his Ph.D. degree in EECS from the University of California at Berkeley in 1993. He is presently a Professor in the School of Electrical and Computer Engineering at Seoul National University, South Korea. Lee's research interests involve augmented reality displays, 3D displays, and metamaterial applications. He has been recognized as a fellow of the SPIE, OSA, and SID and served as the president of the Optical Society of Korea. Lee has also served on the board of directors of the OSA and holds the distinction of being a fellow of the Korean Academy of Science and Technology, as well as a senior member of the National Academy of Engineering of Korea. Lee has garnered several accolades throughout his career, including the National Jin-Bo-Jang Badge of Science of Korea.

Edmund Y. Lam received his B.S., M.S., and Ph.D. degrees in electrical engineering from Stanford University. He previously held the position of Visiting Associate Professor in the Department of Electrical Engineering and Computer Science at the Massachusetts Institute of Technology. Currently, he serves as a Professor of electrical and electronic engineering at the University of Hong Kong. In addition, he holds positions as the Computer Engineering Program Director and a Research Program Coordinator at the AI Chip Center for Emerging Smart Systems. Lam's research interests include computational imaging algorithms, systems, and applications. He has been recognized as a fellow of Optica, SPIE, IS&T, and HKIE, and as a founding member of the Hong Kong Young Academy of Sciences.