Coherent Plug-and-Play: Digital Holographic Imaging Through Atmospheric Turbulence Using Model-Based Iterative Reconstruction and Convolutional Neural Networks

Casey J. Pellizzari, Mark F. Spencer, and Charles A. Bouman, *Fellow, IEEE*

***Abstract*—In order to image a distant object through atmospheric turbulence, it is necessary to correct for the phase errors that would otherwise cause rapidly varying spatial blur in a conventionally focused image. One approach to solving this problem is to illuminate an object with coherent light and to use a digital holography (DH) receiver to form a coherent measurement. The associated amplitude and phase can then be used with model-based iterative reconstruction (MBIR) frameworks to estimate and correct for atmospheric phase errors from single-shot DH data (i.e., one sensor measurement). In this work, we present a new approach for the reconstruction of optically-coherent images from single-shot DH data in the presence of atmospheric turbulence, referred to as Coherent Plug-and-Play (C-PnP). Our algorithm integrates a convolutional neural network (CNN) image model with physics-based models for image reconstruction from DH data corrupted by atmospheric phase errors. C-PnP combines the modeling power of deep neural networks with the accuracy of existing physics models. Based on an extension of the plug-and-play framework, C-PnP uses multi-agent consensus equilibrium to balance the influence of these models. When compared with an existing approach using a simple image model, C-PnP improves image quality by a factor of $2.2\times$ and phase-error correction by a factor of $2.9\times$, on average. We obtain these results by considering a wide range of images, signal levels, and phase-error strengths.**

***Index Terms*—Atmospheric turbulence, coherent image reconstruction, digital holography, neural networks.**

I. Introduction

Digital holography (DH) uses coherent illumination and spatial-heterodyne interferometry to detect a real-valued interference pattern, known as a hologram [1]. Encoded in the real-valued hologram is information about the amplitude and phase of the illuminated object. We obtain this information by applying a demodulation and filtering process. In practice, DH data is corrupted by shot noise from the measurement process and from high-spatial-frequency variations, known as speckle. Speckle occurs when we coherently illuminate an object with a surface that is statistically rough relative to the laser wavelength. In many cases, aberrations within the optical path also corrupt the phase of the received light. Such is the case when imaging objects over long distances, since the atmosphere introduces aberrations that distort the resulting image. Thus, to form useful images from a DH sensor, we must overcome the limitations of measurement noise and speckle, and we must estimate and remove the phase errors caused by aberrations within the optical path.

A popular class of algorithms for estimating and digitally correcting phase errors from DH data involve maximizing an image sharpness metric [2]–[5]. These image-sharpening (IS) techniques reconstruct images from the magnitude of the complex-valued object reflectivity, which leads to images containing speckle. IS algorithms are sensitive to speckle variations and require incoherent averaging of multiple data realizations from multiple sensor measurements, known as multi-shot DH data, to produce accurate estimates of both the image and phase errors. In [6], a new IS technique was developed that uses improved parameterization of the phase-error function along with spatial binning for speckle-contrast reduction. This new IS algorithm works well for single-shot DH data; however, its performance suffers in high-noise conditions and it still produces a speckled image.

In addition to IS algorithms, regularized inversion techniques have been proposed for reconstructing images from DH data. In both [7] and [8], the authors present different methods for reconstructing complex-valued images directly from the real-valued hologram. Unfortunately, neither of these methods consider speckle variations or phase errors in the system, both of which degrade performance. Consequently, these frameworks would require significant modifications for practical use.

In [9]–[11], we developed a regularized inversion approach for DH image reconstruction. We used a Bayesian framework for estimating the unaberrated speckle-free image, r, and any phase errors, ϕ, from single-shot DH data (i.e., one sensor measurement). The approach, known as DH model-based iterative

Manuscript received April 8, 2020; revised August 30, 2020 and October 26, 2020; accepted November 26, 2020. Date of publication December 7, 2020; date of current version December 21, 2020. This work was supported by NSF under Grant CCF-1763896. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Ulugbek S. Kamilov. *(Corresponding author: Casey J. Pellizzari.)*

Casey J. Pellizzari is with the Department of Physics, United States Air Force Academy, CO 80840 USA (e-mail: casey.pellizzari@gmail.com).

Mark F. Spencer is with the Air Force Research Laboratory, Directed Energy Directorate, Kirtland Air Force Base, NM 87111 USA (e-mail: msphotonics@gmail.com).

Charles A. Bouman is with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: bouman@purdue.edu).

Digital Object Identifier 10.1109/TCI.2020.3042948

reconstruction (DH-MBIR), uses the expectation maximization (EM) algorithm to jointly estimate r and ϕ. While the term MBIR represents a wide class of algorithms, in this work, we use DH-MBIR for the specific algorithm found in [11].

DH-MBIR provides improved capabilities for reconstructing DH images in the presence of speckle, measurement noise, and atmospheric phase errors; however, it uses a simple analytical image model. In particular, the algorithm uses a Q-generalized Gaussian Markov random field (QGGMRF) to enforce correlation between neighboring pixels. This approach works well for simple images (e.g., binary resolution charts); however, it is not sufficient for more complex grayscale objects.

Recent advances in image processing have resulted in techniques that incorporate advanced image models into regularized inversion frameworks [12]–[17]. Many of these approaches split the estimation problem into multiple smaller problems and use a Gaussian denoising algorithm to enforce regularization of the image. These so-called "Plug-and-Play" (PnP) methods inherit the image model associated with the denoising algorithm, either explicitly or implicitly.

Of particular interest is the use of convolutional neural networks (CNNs) as the PnP denoiser. CNNs show an immense capacity for learning complex image models. While CNNs can also be used for reconstructing images directly, without iterative regularized-inversion frameworks, most networks do not explicitly incorporate fundamental knowledge of the underlying physics that describe the sensing process. For many sensing applications, including DH, we have a detailed understanding of the underlying physics. Naturally, we want to include this information to better constrain the estimation process.

In [18]–[20], the authors explicitly incorporate physics models into the CNN training process using so-called unrolling methods. However, unrolling methods cannot be directly applied to our problem due to the structure of the physical forward model. This limitation results from the fact that the forward model for estimating the real-valued image from DH data does not have a tractable form and must be minimized using successive approximation with the EM algorithm [9]–[11], [21].

While unrolling methods are not feasible for our problem, we can use PnP methods to couple physics models with advanced image priors learned using a CNN. However, adapting these PnP methods to reconstruct real-valued images from DH data corrupted by atmospheric phase errors is not direct. Overcoming the intractable forward model and integrating the joint estimation and correction of the phase errors requires significant modifications to existing frameworks. In [22], we first explored the use of the alternating direction method of multipliers (ADMM) PnP framework [12] for coherent data. However, that approach was for synthetic aperture lidar images in the absence of any phase errors, and we did not use CNN image models.

Recently, a generalization of consensus optimization problems, like those solved by ADMM PnP, was developed [16]. The approach, known as Multi-Agent Consensus Equilibrium (MACE), provides a generalized framework to balance multiple agents that have different objectives. Furthermore, it relaxes the constraints on the denoising operator required by the PnP framework [13], [16], [23]. MACE has been successfully demonstrated for many applications that involve conventional linear forward models [24]–[27]. However, applying it to image reconstruction from DH data corrupted by atmospheric phase errors is more complicated.

In this work, we present a new approach for image reconstruction from single-shot DH data that is based on an extension of the MACE framework. We define three agents, two for updating the image estimate and one for updating the phase-error estimate. Our first image agent, F_1, favors images that are consistent with the stochastic physics-based measurement model of a DH sensor. To overcome the non-tractability of this coherent-physics model, we use the EM algorithm to find a suitable surrogate. For our second image agent, F_2, we train a CNN to output images that are consistent with a learned image model. The functional form of F_2 is simply a CNN that removes additive white Gaussian noise with a fixed variance. Thus, training F_2 is relatively simple and we can update it with new and better CNN denoisers as they become available. Finally, our phase-error agent, F_3, outputs estimates of the phase errors, which are consistent with both the DH physics model and with a relatively simple prior model. We use the MACE framework to balance the influence of these three agents during the estimation process.

The resulting algorithm, which we call Coherent Plug-and-Play (C-PnP), represents a comprehensive estimation framework that can overcome the combination of speckle, measurement noise, and phase errors to reconstruct complex grayscale images from a single-shot of DH data. When tested on synthetic data over a wide range of conditions, this new approach produces focused speckle-free images that are significantly more accurate than those produced by previous approaches for image reconstruction from single-shot DH data. Additionally, the increased image quality results in better estimates of the phase errors in nearly all cases.

Our main contributions include:

- The adaptation of the MACE framework to include the joint estimation of the atmospheric phase errors, a high-dimensional nuisance parameter;
- The introduction of a novel form of the MACE equations that allow for the required use of the EM algorithm in the formulation of the data-fidelity forward-model agent;
- The verification that the resulting algorithm produces significantly higher-quality images when compared to DH-MBIR over a wide range of signal levels and atmospheric turbulence strengths;
- The demonstration that improved image modeling leads to better estimates of the phase errors.

While our approach is based on the MACE framework, we note that another approach could be developed based on the fast iterative shrinkage/thresholding algorithm (FISTA) algorithms proposed in [14] and [17]. However, implementation of these algorithms is also not direct since they require the evaluation of a gradient which could be difficult for our problem due to the intractable nature of the forward model. Therefore, we choose to adapt the MACE framework given the success of our initial work with ADMM PnP [22].

In what follows, Section II better formulates the problem at hand, whereas Section III provides the details associated with our new approach to image reconstruction from single-shot DH data. Section IV then follows with our methodology for implementing this new approach, and Section V provides results with comparisons to previous approaches. In Section VI, we conclude this paper and recap the contributions present within.

II. Problem Formulation

In this section we describe the problem associated with reconstructing images from single-shot DH data in the presence of unknown atmospheric phase errors. We also motivate the need for improved image priors that go beyond the conventional analytical models previously used to solve this problem. Finally, we introduce the MACE framework and discuss its limitations for solving our problem.

A. *Digital Holographic Image Reconstruction*

A DH sensor allows us to measure the complex electromagnetic field reflected off of an object using spatial-heterodyne interferometry. Under the assumption of shot-noise limited detection, we model the complex-valued measurements, $y \in \mathbb{C}^M$, as

$$y = A_\phi g + w, \tag{1}$$

where $g \in \mathbb{C}^M$ is the unknown complex-valued reflection coefficient for the illuminated object, $w \in \mathbb{C}^M$ is the complex-valued measurement noise, and $A_\phi \in \mathbb{C}^{M\times M}$ is a linear transform that accounts for the propagation and measurement geometry and that is dependent on the phase errors, ϕ [11]. For a review of the DH sensor concept, we direct the reader to App. A.

In [11], we provided details on a generalized structure for A_ϕ. For this work, we restrict ourselves to cases where the distance between the object and sensor is sufficiently large enough that we may use a discrete Fourier transform (DFT) to model the propagation of light. We also restrict our analysis to cases in which the atmospheric turbulence is concentrated near the sensor, resulting in a shift-invariant point spread function (PSF) in the image domain. Our resulting model for A_ϕ is given by

$$A_\phi = D(a)D(e^{iP\phi})F, \tag{2}$$

where $D(\cdot)$ denotes an operator that produces a diagonal matrix from its vector argument, $a \in \mathbb{R}^M$ is binary aperture vector that represents the transparency of our circular aperture in the pupil plane, and $F \in \mathbb{C}^{M\times M}$ is a 2D DFT matrix. Furthermore, we introduce an interpolation matrix, $P \in \mathbb{R}^{M\times L}$, where $L \leq M$, that allows us to model the phase errors, $\phi \in \mathbb{R}^L$, on a low-resolution grid. This low-resolution representation allows us to reduce the number of unknowns during reconstruction.

Several existing approaches for DH image reconstruction attempt to invert Eq. (1) to obtain an estimate of the complex-valued reflection coefficient, g [2]–[5]. However, images formed from the magnitude, or magnitude squared, of g contain speckle. Speckle occurs when an object's surface is rough relative to the illumination wavelength, resulting in g having a uniformly distributed phase [28]. In this case, g is well-modeled as a conditionally complex Gaussian random variable with distribution

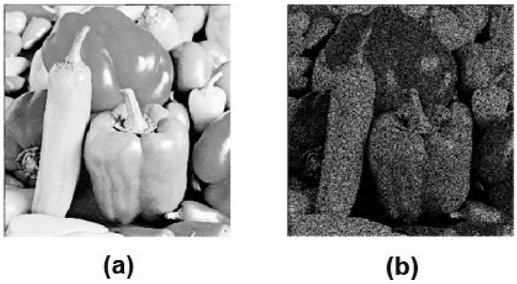

Fig. 1. Reflectance versus reflection coefficient. In (a), we show an example of the real-valued reflectance function, r, whereas in (b), we show the magnitude squared of a corresponding reflection coefficient realization, g.

$$p(g|r) \sim CN(0, D(r), 0). \tag{3}$$

where $p(\cdot|\cdot)$ represents a conditional probability distribution and $CN(\mu, C, \Gamma)$ indicates a multivariate complex normal distribution with mean, μ, covariance matrix, C, and pseudo-covariance matrix, Γ [28]. Here, r is the object's real-valued reflectance function, which is typically a smoother quantity with higher spatial correlation when compared to the reflection coefficient, g [9], [28].

Fig. 1 shows an example reflectance function, r, and the magnitude squared for a corresponding realization of the reflection coefficient, g. Note that r is the quantity we typically observe in incoherent images, while $|g|$ or $|g|^2$ is observed in coherent images, like those from single-shot DH data. Images formed from the amplitude, or amplitude squared, of g contain speckle, whereas images formed from r do not. For pixel $s \in S$, the two quantities are related according to $r_s = E[|g_s|^2|r_s]$, where $E[\cdot|\cdot]$ indicates the conditional expectation [28]. This relationship leads to the use of so-called speckle averaging where independent realizations of g, obtained from multi-shot DH data, are averaged to better estimate r and reduce speckle contrast in the image.

While it is possible to apply a smoothness constraint to g when reconstructing images from single-shot DH data, such an approach suffers from an inherent disagreement between the physical model and the regularizer. Figure 1 shows an example of $|g|$ and illustrates that it is not smooth for diffuse reflectors [28]. Conversely, r is a smoother quantity with higher spatial correlation between elements. Enforcing smoothness when reconstructing r is consistent with the physical model and leads to more accurate reconstructions [29].

Given the complex data, y, we wish to jointly estimate the speckle-free image, r, and the phase errors, ϕ, from single-shot DH data. Note that the data, y, is linearly related to g, but not to r. This means that direct inversion of single-shot DH data cannot be used to find r. Instead, we formulate this problem as a joint maximum a posterior (MAP) estimate given by

$$(\hat{r}, \hat{\phi}) = \underset{r,\phi}{\operatorname{argmin}} \left\{ -\log p\left(y|r,\phi\right) - \log p(r) - \log p\left(\phi\right) \right\}, \tag{4}$$

where we assume that r and ϕ are independent [11].

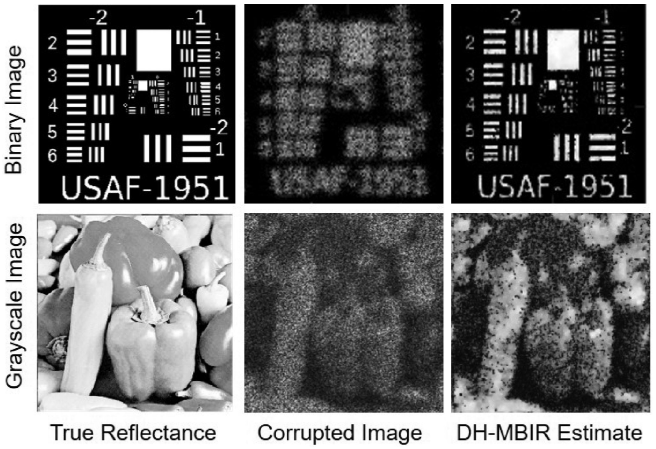

Fig. 2. Example image reconstructions using DH-MBIR with a simple prior model, $p(r)$, for a binary object (top) and a gray-scale object (bottom). The reconstruction for the grayscale object contains noise artifacts that are inconsistent with what we expect to see in natural images. Here, the prior model is not effectivly enforcing a realistic distribution for the reconstructed image.

In previous research [9]–[11], we proposed the DH-MBIR method to solved Eq. (4) using a simple prior model and an adaptation of the EM algorithm to address the intractable forward model. However, while the simple QGGMRF prior model is analytically-tractable, it does not capture the subtle characteristics of real image behavior. As a result, reconstructions of complex grayscale objects contain noise artifacts that are not consistent with what we expect to see in natural images [21].

Fig. 2 shows example image reconstructions of the DH-MBIR algorithm for a simple binary resolution chart and a grayscale object. In this work, the poor performance of the QGGMRF prior model at enforcing a realistic distribution for $\hat{r}$ motivates us to solve this problem using more-sophisticated image models for $p(r)$. Note that since the phase-error functions we wish to estimate are relatively simple when compared to reflectance images, we continue to use a simple and analytically-tractable Gaussian Markov random field (GMRF) model for $p(\phi)$.

B. MACE Framework

The MACE framework is a generalization of the ADMM PnP approach [12] that is often used to incorporate advanced image models into the reconstruction process. It provides a set of balance equations that allow us to moderate a solution to a problem given multiple agents, F_i for $i \in [1, 2, ...N]$, with differing objectives [16]. When the N agents are in consensus, the equilibrium equation is given by

$$\mathbf{F}(\mathbf{x}^*) = \mathbf{G}(\mathbf{x}^*), \tag{5}$$

where

$$\mathbf{F}(\mathbf{x}) = \begin{bmatrix} F_1(x_1) \\ \vdots \\ F_N(x_N) \end{bmatrix} \text{ and } \mathbf{G}(\mathbf{x}) = \begin{bmatrix} \bar{x} \\ \vdots \\ \bar{x} \end{bmatrix}, \tag{6}$$

and $\mathbf{x} = (x_1^T, x_2^T, ..., x_N^T)$ is a set of input state vectors. Here, $\mathbf{F}(\mathbf{x})$ is an operator formed by the application of the N agents, and $\mathbf{G}(\mathbf{x})$ is an averaging operator [16] where

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i. \tag{7}$$

The solution to the MACE equations is a fixed point of the mapping $\mathbf{T} = (2\mathbf{G} - I)(2\mathbf{F} - I)$, where I is the identity map [16]. We can find this fixed point using the Mann iteration with the following update

$$\mathbf{x} \leftarrow (1 - \rho)\mathbf{x} + \rho \mathbf{T}(\mathbf{x}), \tag{8}$$

where $\rho \in (0, 1]$ is a scalar that controls the convergence properties of the algorithm. Note that this update step allows for the parallel application of the N agents.

MACE is a generalization of the commonly used MAP reconstruction given by

$$\hat{x} = \underset{x}{\operatorname{argmin}} \left\{ f_1(x) + \beta f_2(x) \right\}, \tag{9}$$

where $f_1(x) = -\log(y|x)$ and $\beta f_2(x) = -\log p(x)$ [16]. Here, β is a scalar that allows us to control amount of regularization.

When using MACE to solve Eq. (9), we may define our agents as proximal maps for f_1 and βf_2 [16], [24]–[27], given by

$$F_1(x_{in}) = \underset{x}{\operatorname{argmin}} \left\{ f_1(x) + \frac{1}{2\sigma^2} ||x - x_{in}||^2 \right\} \tag{10a}$$

$$F_2(x_{in}) = \underset{x}{\operatorname{argmin}} \left\{ \beta f_2(x) + \frac{1}{2\sigma^2} ||x - x_{in}||^2 \right\}. \tag{10b}$$

These proximal maps take an arbitrary input image, x_{in}, and map it to an output that is closer in proximity to the minimum of f_1 and βf_2, respectively. The second term in each agent restricts how strongly we map x_{in} towards the minimum of f_1 or βf_2. We control the strength of this mapping by adjusting the parameter σ^2.

Intuitively, both F_1 and F_2 are competing agents for our image update. The agent, F_1, pushes the output towards a solution that is consistent with the physical measurements, and F_2 pushes the output towards a solution that is consistent with our prior knowlege. We use MACE to balance these two image agents and produce an output that is consistent with both.

While we initially define our agents using proximal maps that incorporate optimization of explicit analytical functions, a critical advantage of using MACE is that we can move beyond such restrictive frameworks. MACE allows us to generalize the estimation process by replacing these proximal maps with agents that use machine learning techniques and require no optimization.

To see why this works, consider that Eq. (10b) is mathematically equivalent to a Gaussian denoiser [12], [23]. The operation outputs the MAP estimate of x given a noisy input, x_{in}, and a Gaussian forward model with variance, $\sigma_n^2 = \beta\sigma^2$. Therefore, we borrow from the Plug and Play approaches and replace F_2 with a denoiser designed to remove Gaussian noise with standard deviation σ_n. In particular, we can use a CNN denosier that has been trained to implicitly learn $p(r)$.

Using a CNN Gaussian denoiser for F_2 provides three primary benefits. First, we are not required to conduct an optimization step when applying this agent. Instead, we simply apply the denoiser. Second, we inherit the implicit $p(r)$ learned by the CNN to better capture subtle image characteristics that explicit analytical models lack. The third benefit of using a denosier for our agent is that we can replace it with better denoisers as they become available.

A direct implementation of the MACE framework for our DH imaging problem is not possible for two reasons. First, we are reconstructing the real-valued reflectance, r, directly from data, y, and therefore the forward model is intractable. Using Eqs. (1)–(3) and (34), we find the form of our likelihood function is given by

$$p(y|r,\phi) \sim CN(0, C_{y|r,\phi}, 0), \tag{11}$$

where $C_{y|r,\phi}$ is a dense covariance matrix given by

$$C_{y|r,\phi} = A_\phi D(r) A_\phi^H + \sigma_w^2 I, \tag{12}$$

and the superscript H indicates the Hermitian transpose [11]. The resulting log likelihood function becomes

$$f_1(r,\phi) = \log |C_{y|r,\phi}| + y^H C_{y|r,\phi}^{-1} y. \tag{13}$$

From Eq. (13) we see that evaluating $f_1(r,\phi)$ requires finding the determinant and inverse of a large dense matrix, making it intractable to use in practice. Therefore, the proximal map in Eq. (10a) cannot be used directly.

The second reason we cannot implement MACE directly for our problem is that we must jointly estimate the phase errors. As detailed in the next section, we require three different agents operating on two separate unknowns. Our approach generalizes the basic framework developed in [16].

III. Coherent Plug-and-Play

In this section we present our extension of the MACE framework for use in reconstructing images from single-shot DH data corrupted by phase errors. We start by defining an ideal set of equilibrium equations and agents. Next we use the EM algorithm to define new agents that are analytically tractable. Finally, we present the full set of update steps for our algorithm.

A. Ideal Agents

We start by rewriting our original MAP estimation problem, given by Eq. (4), as

$$(\hat{r}, \hat{\phi}) = \underset{r,\phi}{\operatorname{argmin}} \left\{ f_{\sum}(r,\phi) \right\}, \tag{14}$$

where

$$f_{\sum}(r,\phi) = f_1(r,\phi) + \beta f_2(r) + f_3(\phi), \tag{15}$$

and

$$\begin{aligned} f_1(r,\phi) &= -\log p(y|r,\phi) \\ \beta f_2(r) &= -\log p(r) \\ f_3(\phi) &= -\log p(\phi). \end{aligned} \tag{16}$$

Similar to the baseline MACE framework, we introduce a β in front of f_2 as a means to control the amount of regularization for r when we use a denoising algorithm as our prior agent. We do not need a similar term for f_3 since we use an analytical function that already contains a parameter to control the regularization of ϕ.

With Eq. (15) in mind, we define roles for three agents:

- *Agent 1:* Enforces consistency between the image estimate and our physical measurements
- *Agent 2:* Enforces consistency between the image estimate and our image model learned using a CNN denoising algorithm
- *Agent 3:* Estimates and corrects phase errors

Given these desired roles, we propose the following three agents

$$F_1(r_{in}, \phi_{in}) = \underset{r}{\operatorname{argmin}} \left\{ f_1(r, \phi_{in}) + \frac{1}{2\sigma^2} ||r - r_{in}||^2 \right\}, \tag{17a}$$

$$F_2(r_{in}) = \underset{r}{\operatorname{argmin}} \left\{ \beta f_2(r) + \frac{1}{2\sigma^2} ||r - r_{in}||^2 \right\}, \tag{17b}$$

$$F_3(r_{in}) = \underset{\phi}{\operatorname{argmin}} \left\{ f_1(r_{in}, \phi) + f_3(\phi) \right\}. \tag{17c}$$

Here, F_1 is a proximal map for f_1 with respect to r only and F_2 is a proximal map for f_2. The third agent computes the MAP estimate of the phase errors given the input image, r_{in}. Note that F_3 is a function of r because f_1 is a function of r.

We refer to the agents in Eq. (17) as our ideal agents. They are designed to solve the original MAP problem defined in Eq. (14) through the direct optimization of our intractable log likelihood function, f_1. Therefore, explicit application of agents F_1 and F_3 is not possible. In the next section, we adapt ideas from [9] and use the EM algorithm to develop alternate forms of F_1 and F_3 that we can use in practice.

B. EM Extension

To overcome the intractable forms of F_1 and F_3, we design a surrogate for the log likelihood function, written as $Q(r,\phi; r', \phi')$, where r' and ϕ' are the current estimates of the reflectance and phase errors, respectively [9]. We then use this surrogate to define two alternate agents, $\tilde{F}_1$ and $\tilde{F}_3$, that allow us to establish a set of MACE equations that we can solve in practice.

Surrogate functions are useful because they allow us to approximate a complex function, f, about some point, x', with a simpler function, Q, similar to a Taylor series approximation [30]. For any function $f(x)$, we define a surrogate function, $Q(x; x')$, to be an upper-bounding function such that $f(x) \leq Q(x; x') + c \; \forall x$, where $c = f(x') - Q(x'; x')$ is a constant representing the difference between the two functions at the point of approximation, x' [30].[1]

[1] The constant, c, depends on the point of approximation x'. However, for more concise notation, we drop this dependency.

There are several properties of a surrogate function that make them useful for optimization problems. First, the monotonicity property implies that if we construct a surrogate at point x' and minimize it with respect to x, we have also minimized f with respect to x. That is, if $Q(x; x') <= Q(x'; x')$, then

$$f(x) \leq Q(x; x') + c \leq Q(x'; x') + c = f(x'). \tag{18}$$

Here, the first inequality comes directly from our definition of Q, the second inequality comes from the fact that we have reduced Q, or at least kept it constant, and the third inequality comes from our definition of Q evaluated at x'. The second property of surrogate functions states that if both Q and f are continuously differentiable functions of x, then the gradients of both functions are equal at point x'. Lastly, the third property of surrogate functions states that any fixed point of the surrogate function, x^*, must be a point where $\nabla f(x^*) = 0$. Additional properties and explanations of surrogate functions can be found in [30].

In this work, we use the EM algorithm as a formal framework for constructing a surrogate function for $f_1(r, \phi)$. Specifically, we introduce g as the unobserved data which simplifies the functional form of f_1. Next, we marginalize over g by taking the expected value. The resulting surrogate function is given by

$$\begin{aligned} Q(r, \phi; r', \phi') &= E_g\left[-\log p(y|g, r, \phi) \mid r', \phi'\right], \\ &= E_g\left[-\log p(y|g, \phi)p(g|r) \mid r', \phi'\right], \\ &= Q_r(r; r', \phi') + Q_\phi(\phi; r', \phi') + c, \end{aligned} \tag{19}$$

where $E_g[\cdot|\cdot]$ indicates a conditional expectation with respect to g, c is constant with respect to r and ϕ, and

$$Q_r(r; r', \phi') = E_g\left[g^H D(r)^{-1} g \mid r', \phi'\right] - \log|D(r)|, \tag{20a}$$

$$Q_\phi(\phi; r', \phi') = E_g\left[\frac{1}{\sigma_w^2}\,||y - A_\phi g||^2 \mid r', \phi'\right]. \tag{20b}$$

By introducing g in Eq. (19) we simplify the forward model, when compared to Eq. (13), since there is a linear relationship between y and g, and a relatively simple relationship between g and r. Note that we still must take a determinant and inverse in Eq. (20a); however, because $D(r)$ is diagonal, this becomes trivial. Also note that Eq. (19) is separable in r and ϕ which further simplifies optimization with respect to each variable.

Using our surrogate function for f_1, we define our new EM-based agents as

$$\tilde{F}_1(r_{in}; r', \phi') = \underset{r}{\operatorname{argmin}}\left\{Q_r(r; r', \phi') + \frac{1}{2\sigma^2}\,||r - r_{in}||^2\right\}, \tag{21a}$$

$$F_2(r_{in}) = \underset{r}{\operatorname{argmin}}\left\{\beta f_2(r) + \frac{1}{2\sigma^2}\,||r - r_{in}||^2\right\}, \tag{21b}$$

$$\tilde{F}_3(r', \phi') = \underset{\phi}{\operatorname{argmin}}\left\{Q_\phi(\phi; r', \phi') + f_3(\phi)\right\}. \tag{21c}$$

Applying these agents represents a single iteration of the EM algorithm. Taking the expectation in Eq. (19) constitutes the E-step, and minimizing with respect to r and ϕ during the optimizations associated with Eqs. (21a) and (21c) constitutes the M-step.

Given our EM agents, we define our equilibrium equations as

$$\begin{bmatrix} \tilde{F}_1(r_1^*; \bar{r}^*, \phi^*) \\ F_2(r_2^*) \\ \tilde{F}_3(\bar{r}^*, \phi^*) \end{bmatrix} = \begin{bmatrix} \bar{r}^* \\ \bar{r}^* \\ \phi^* \end{bmatrix}, \tag{22}$$

where

$$\bar{r}^* = (r_1^* + r_2^*)/2. \tag{23}$$

Equation (22) represent the basis of our C-PnP framework. Solving this equation allow us to jointly estimate both the real-valued reflectance and the phase errors from single-shot DH data in a way that separates the data fidelity and prior terms for r. Furthermore, theorem III.1 shows that, for the agents defined in Eq. (21), any solution, $[\mathbf{r}^*, \phi^*]$, to the EM-based MACE equations is also a solution to the original MAP estimation problem.

Theorem III.1: For the agents defined in Eq. (21), any solution, $[\mathbf{r}^*, \phi^*]$ to the EM-based MACE equations given by Eq. (22) also has the property that $\nabla_{r,\phi} f_{\sum}(\bar{r}^*, \phi^*) = 0$, where $\mathbf{r}^* = [r_1^{*T}, r_2^{*T}]^T$ and $\bar{r}^* = (r_1^* + r_2^*)/2$.

Proof: See Appendix B. ■

To solve Eq. (22), we use a combination of the Mann iteration for r and the application of $\tilde{F}_3$ to update ϕ. Our update equations are given by

$$\begin{aligned} \mathbf{w} &\leftarrow \begin{bmatrix} \tilde{F}_1(r_1; \bar{r}', \phi') \\ F_2(r_2) \end{bmatrix} \\ \phi &\leftarrow \tilde{F}_3(\bar{r}', \phi') \\ \mathbf{r} &\leftarrow \mathbf{r} + 2\rho\left[\mathbf{G}(2\mathbf{w} - \mathbf{r}) - \mathbf{w}\right], \end{aligned} \tag{24}$$

where $\mathbf{r} = [r_1^T, r_2^T]^T$, $\bar{r}' = (r_1' + r_2')/2$, and $\mathbf{G}$ is the averaging operator defined in Eq. (6). Note that the three agents can be applied in parallel.

The separable nature of our problem ensures that sequential updates of r and ϕ are equivalent to jointly updating the two variables. Our approach applies the Mann iteration to update r in a manner that converges to a fixed point of the operator $\mathbf{T} = (2\mathbf{G} - I)(2\mathbf{F} - I)$. We also update ϕ until it converges to a fixed point of $\tilde{F}_3$. Theorem III.2 shows that when these updates converge to a fixed point, that fixed point is also a solution to the MACE equations defined in Eq. (22). Additionally, per Theorem III.1, any solution to the EM MACE equations is a point at which the gradient of our original MAP cost function is zero.

Theorem III.2: When the update steps in Eq. (24) converge to a fixed point, $[\mathbf{w}^*, \phi^*, \mathbf{r}^*]$, where $\mathbf{w}^* = [w_1^{*T}, w_2^{*T}]^T$ and $\mathbf{r}^* = [r_1^{*T}, r_2^{*T}]^T$, the fixed point is also a solution to the MACE equations defined in Eq. (22).

Proof: See Appendix B. ■

C. Agent Implementation

In this section, we provide an overview of how we implement our EM-based agents in practice. Appendix C-A contains explicit details for how we generate the surrogate function and

App. C-B and C-C provide the update equations used for both $\tilde{F}_1$ and $\tilde{F}_3$, respectively.

1) Image Agent, $\tilde{F}_1$: Application of the agent, $\tilde{F}_1$, requires that we first specify our surrogate function about our current point, (r', ϕ'). Once we have our surrogate, we minimize the argument of Eq. (21a). Fortunately, the update for each pixel during this minimization is decoupled and has a closed-form solution. The solution is equivalent to rooting the 3rd order polynomial given by Eq. (58). This process is well suited for parallelization. Note that agent $\tilde{F}_1$ is parameterized by σ_w^2, which is shared with agent $\tilde{F}_3$ and by σ^2 which is shared with agent F_2.

2) Image Agent, F_2: One of the key benefits of C-PnP is that it allows us to replace the proximal map, given by Eq. (21b), with a CNN denoiser, $\mathcal{G}$, designed to remove Gaussian noise with variance

$$\sigma_n^2 = \beta\sigma^2. \tag{25}$$

We therefore implement F_2 according to

$$F_2(r_{in}) = \mathcal{G}(r_{in}). \tag{26}$$

While we can replace F_2 with other Gaussian denoising algorithms, for this work, we use a modified version of the denoising convolutional neural network (DnCNN) architecture [31]. We provide details for our specific network in Section IV-B. Note that agent F_2 is parameterized by β and by σ^2, which is shared with agent F_1.

3) Phase-Error Agent, $\tilde{F}_3$: As was the case for $\tilde{F}_1$, we must first specify our surrogate function about the current point, (r', ϕ'), in order to implement agent $\tilde{F}_3$. Next, we minimize according to Eq. (21c). Given the relatively simple structure of atmospheric phase errors, when compared to natural images, we use an analytical function to model ϕ. In particular, we use a GMRF given by

$$p(\phi) = \frac{1}{z}\exp\left\{-\sum_{\{i,j\}\in\mathcal{P}} b_{i,j}\frac{|\phi_i - \phi_j|^2}{2\sigma_\phi^2}\right\}. \tag{27}$$

Here, z is the partition function, $b_{i,j}$ is the weight between pixel pairs, $\mathcal{P}$ is the set of all pair-wise cliques falling within the same neighborhood, and σ_ϕ controls the variation in ϕ [30].

Equation (27) is a model of the low-resolution phase errors, ϕ. When applying our forward-model operator, A_ϕ, we must interpolate to a full-resolution representation. In this work, we use a nearest-neighbors interpolation scheme. This simplifies the form of P to a $M \times L$ matrix with elements in the set $\{0, 1\}$. Each row of the tall matrix, P, has a single non-zero element corresponding to the nearest low-resolution sample.

To evaluate agent $\tilde{F}_3$, we use a single iteration of iterative coordinate descent (ICD). In App. C-C, we provide the cost-function used for updating each pixel with a 1D line search. Note that agent $\tilde{F}_3$ is parameterized by the size of the interpolation matrix, L, the neighborhood weights, $b_{i,j}$, and σ_ϕ^2, which controls the variation in ϕ. It is also parameterized by σ_w^2, which is shared with agent $\tilde{F}_1$.

```
Initialize: $\phi' \leftarrow \mathbf{0}$
Repeat $N_1$ times: {outer loop}
  {Initialize: $r' \leftarrow r_1 \leftarrow r_2 \leftarrow |A_\phi^H y|^{\circ 2}$
    Repeat $N_2$ times: (inner loop)
     ( E-step:
        $Q_r(r; r', \phi') = E_g\left[g^H D(r)^{-1} g \;\middle|\; r', \phi'\right] - \log|D(r)|$
        $Q_\phi(\phi; r', \phi') = E_g\left[\frac{1}{\sigma_w^2}\left|\left|y - A_\phi g\right|\right|^2 \;\middle|\; r', \phi'\right]$
       M-step:
        $\mathbf{w} \leftarrow \begin{bmatrix} \tilde{F}_1(r_1; \bar{r}', \phi') \\ \mathcal{G}(r_2) \end{bmatrix}$
        $\phi \leftarrow \tilde{F}_3(\bar{r}', \phi')$
        $\mathbf{r} \leftarrow \mathbf{r} + 2\rho\left[\mathbf{G}(2\mathbf{w} - \mathbf{r}) - \mathbf{w}\right]$
        $\bar{r}' \leftarrow \frac{1}{2}(r_1 + r_2)$, $\phi' \leftarrow \phi$
     )
  }
```

Fig. 3. The C-PnP algorithm consists of an outer loop used to initialize ϕ and an inner loop to iteratively conduct the joint estimation. Here, $|\cdot|^{\circ 2}$ indicates the element-wise magnitude square of a vector. As noted in Theorem III.2, when the update steps converge to a fixed point, it is also a solution to the MACE equations defined in Eq. (22). We include the explicit form of the update equations in App. C.

D. C-PnP Algorithm

In this section, we provide the full set of steps used in the C-PnP framework. These steps consists of a set of nested iterations, as shown in Fig. 3. Specifically, there is an outer loop used for initializing the phase errors and an inner loop that implements the update equations defined in Eq. (24). We first describe the inner loop, which is the heart of the C-PnP framework, then explain the utility of the outer loop.

1) C-PnP Inner Loop: In general, the steps of the inner loop iteratively apply the E and M-steps of the EM algorithm. For the E-step, we compute the surrogate function terms, Q_r and Q_ϕ, according to Eq. (56) of App. C-A. During the M-step, we use these functions in our agents to update r and ϕ according to Eq. (24). We repeat the steps of the inner loop until we reach a convergence criteria. This convergence criteria may be in terms of how close the output of the two agents are to each other, the change in ϕ, or in terms of reaching some predetermined number of iterations, N_2, as shown in Fig. 3.

2) C-PnP Outer Loop: While the inner loop is the heart of the C-PnP estimation framework, we designed the outer loop to produce improved initialization of ϕ [9], [29]. Since both the log likelihood function and its surrogate are nonconvex, the algorithm is likely to converge to a local minimum that is not global [29]. With respect to the MACE framework, this means that multiple MACE solutions exist, with some being better than others.

To ensure we find a good solution, in the sense that it produces a focused image, we iteratively change our initial conditions. At the start of the algorithm, we initialize the phase-error function with zeros. Then, for a fixed number of outer-loop iterations, N_1, we run the inner loop N_2 times. Each time we restart the inner loop, r is re-initialized, but we keep the previous value of ϕ. Thus, with each iteration of the outer loop, the inner loop begins with a slightly better estimate of ϕ which results in better initialization of r. On the final outer-loop iteration, we run the inner loop until we achieve our convergence criteria.

IV. Methods

In this section, we explain the methods used to assess the performance of the C-PnP over a range of grayscale object types, phase-error strengths, and measurement-noise levels. Given the success of the DH-MBIR algorithm when compared with IS algorithms [9]–[11], we use the DH-MBIR algorithm as a baseline for comparison.

A. Data Generation

To generate synthetic test data, we used the DH simulation tool developed in [11]. For our input reflectance functions, r, we used the first ten images in Set12, obtained from [32]. Note that these images were not used to train our denoiser, $\mathcal{G}$. To simulate the return field for a given reflectance function, r, we first generated a reflection coefficient according to Eq. (3). Next, we used the numerical techniques from [33] to propagate g through a single phase screen located at the pupil plane. Following [33], we generated the atmospheric phase errors using an FFT-based technique and a Kolmogorov model for the refractive-index power-spectral density (PSD). For this simulation, we set the physical size of the 256×256 source grid to be 5×5 m, and the 256×256 pupil-plane was 0.27×0.27 m. The propagation distance was set to 5 km and our aperture pupil diameter was $D = 0.27$. We characterized the strength of the atmospheric phase errors according to the metric D/r_0, where r_0 is the coherence length of the phase screen. Large values of D/r_0 indicate phase errors with significant variation over the receiving aperture area.

To simulate detection by a DH system using an off-axis IPRG, we padded the 256×256 propagated field to obtain an array size of 512×512. Next, we applied a 512×512 binary circular pupil function, a, which had a circle of ones, 256 pixels wide, centered in an array of zeros. After the pupil, we first applied a thin-lens phase function which collimated the propagated light, then applied a Fast Fourier Transform to form an image in the focal plane. Next, we mixed the image with an off-axis reference beam and detected the resultant power. The reference-beam power was set at approximately 80% of the well depth per detector element [i.e., 80% of 5×10^4 $photoelectrons$ (pe)]. We also modeled Gaussian read noise with a standard deviation of 40 pe and digitized the output to 12 $bits$. After detection, we took an FFT of the single-shot holographic data and isolated the signal of interest, which was a 256×256 complex image of the signal in the pupil plane. We set the signal-to-noise ratio (SNR) according to

$$\text{SNR} = \frac{s^2(y) - s^2(w)}{s^2(w)}, \tag{28}$$

where $s^2(\cdot)$ indicates the variance of the vector argument. For a complex-valued vector, x, the complex-valued mean and real-valued variance are given by

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i, \tag{29}$$

and

$$s^2(x) = \frac{1}{N}\sum_{i=1}^{N}(x_i - \mu_x)(x_i - \mu_x)^*, \tag{30}$$

B. CNN Image Agent

Our CNN image agent is based on the DnCNN architecture and uses residual learning to estimate and remove Gaussian noise from images. We started with the Keras/TensorFlow DnCNN code and the 400 training images found at [32] and retrained the network using an L1 loss function. In accordance with [34], we found the L1 loss function to produce sharper edges when compared to the standard L2 loss function. After training for 50 epochs, we imported the network into MATLAB for use with the C-PnP code. Note that we only trained a single network with $\sigma_n = 0.1$. Fixing σ_n reduces the number of hyperparameters and simplifies training and execution since we only have to train and deploy a single CNN.

C. Hyperparameter Selection

Prior to reconstruction, we normalized the input data, y, so that it had zero mean and unit variance. To do so, we first find the complex-valued mean using Eq. (29) and subtract that from y. Next, we divide by the standard deviation of the result using Eq. (30). We found this normalization process simplified the selection of hyperparameters.

The C-PnP framework has a total of nine hyperparameters. These parameters can be grouped according to the agent parameters, consisting of

- L: Size of the phase-error vector
- σ_w^2: Variance of measurement noise
- σ^2: Strength of the proximal mapping
- σ_n^2: Strength of CNN Gaussian denoiser
- σ_ϕ^2: Variance in phase-error model
- $b_{i,j}$: Neighborhood weights for phase-error model

and the algorithm parameters, consisting of

- N_1: Number of outer loops
- N_2: Number of inner loops
- ρ: Convergence rate for Mann iteration

We found that many of these hyperparameters can be fixed and still provide good reconstruction quality over a wide range of conditions. For this work, we fix eight of these parameters and automatically set the ninth, σ_w^2, based on the measured signal strength of the input data.

Table 1 shows the values of the fixed parameters for both C-PnP and DH-MBIR. These two algorithms share a similar nested loop structure parameterized by N_1 and N_2. To ensure a fair comparison, we set $N_1 = 20$ and $N_2 = 250$ for both algorithms. In addition to sharing a similar nested-loop structure, both algorithms estimate ϕ using a low-resolution representation. For both algorithms, we set $L = 128^2$ which amounts to a down

TABLE I
PARAMETERS USED FOR C-PNP AND DH-MBIR ALGORITHMS

	C-PnP	DH-MBIR
Common Parameters		
Number of outer loops, N_1	20	20
Number of inner loops N_2	250	250
Size of phase-error vector, L	128^2	128^2
Variance in phase-error model, σ_ϕ	0.1 rad	0.1 rad
Neighborhood weights for $p(\phi)$, b	3×3 $G(0.1)$	3×3 $G(0.1)$
C-PnP Specific Parameters		
Strength of proximal mapping, σ^2	0.075	-
Strength of CNN denoiser, σ_n^2	0.01	-
Convergence rate for Mann iteration, ρ	0.8	-
DH-MBIR Specific Parameters		
Image-model parameter, γ	-	3
Image-model parameter, T	-	0.1
Image-model parameter, q	-	2
Image-model parameter, p	-	1.1

sampling of the raw data size, $M = 256^2$, by a factor of $1/4$. We also set $\sigma_\phi = 0.1\ rad$, and used a 3×3 neighborhood for both algorithms, where we set the neighborhood weights, b, using a Gaussian kernel, $G(\cdot)$, with a standard deviation of 0.1 pixels. For the MBIR-specific parameters, we used $\gamma = 3$, $T = 0.1$, $q = 2$, and $p = 1.1$ for the QGGMRF image model [11]. For the C-PnP-specific parameters, we fixed $\rho = 0.8$ and $\sigma^2 = 0.075$. Through trial and error, we found these two C-PnP parameters to work well over a wide range of conditions.

For the noise parameter, σ_w^2, we found that a higher value produced better estimates of the phase errors, but it also reduced the image quality by increasing the contrast. Therefore, we used a higher value, $\sigma_w^2 = 0.3$, during the iterative phase-initialization process, and an equal or lesser value during the final outer-loop iteration. This final lower value was set based on the SNR of the data according to $\sigma_w^2 = 1/\text{SNR}$. Note that we measured the SNR directly from the noisy hologram spectrum using the method described in [10].

D. Quality Metrics

To measure the distortion between the reconstructed images, $\hat{r}$, and the simulation input, r, we used the SSIM function in MATLAB with parameters $[\alpha_{luminance}, \beta_{contrast}, \gamma_{structure}] = [0.25\ \ 0.50\ \ 1.00]$. These values place more emphasis on the structural similarity and less on the luminance similarity. We found these values useful in penalizing over-regularization and blurring.

To measure the quality of the reconstructed phase error, we compare the interpolated phase-error estimate, $P\hat{\phi}$, to the actual full-resolution phase error. We used a metric referred to here as the peak Strehl ratio, S_p. We define S_p as

$$S_p = \frac{\{\text{PSF}_{\text{corrected}}\}_{\max}}{\{\text{PSF}_{\text{diffraction limited}}\}_{\max}}, \tag{31}$$

where $\{\cdot\}_{\max}$ indicates that we take the maximum value of the argument, $\text{PSF}_{\text{corrected}}$ is the point-spread function (PSF) of the imaging system after correction of phase errors, and $\text{PSF}_{\text{diffraction limited}}$ is the diffraction limited PSF. Thus, the peak Strehl ratio is a normalized measure of how close the imaging system performance is to the diffraction limit, neglecting the effects of tilt.

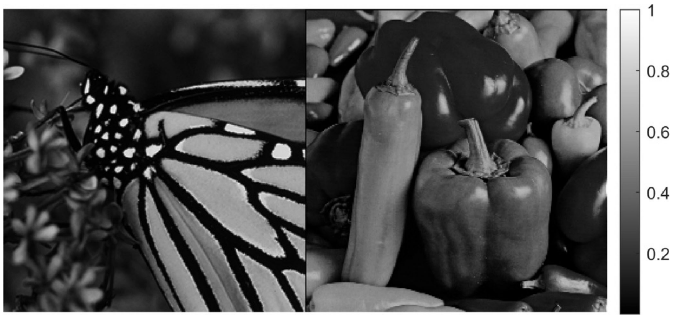

Fig. 4. True reflectance function, r, for Figs. 5 and 6

Finally, to measure how well our C-PnP reconstructions solve the MACE equations, as defined in Eq. (22), we measured the normalized MACE Eq. error, defined as

$$\epsilon = \frac{||\bar{r}-\tilde{F}_1(r_1,\bar{r},\phi)||_1}{||\bar{r}||_1} + \frac{||\bar{r}-F_2(r_2)||_1}{||\bar{r}||_1} + \frac{||\phi-\tilde{F}_3(\bar{r},\phi)||_1}{||\phi||_1}. \tag{32}$$

Here, the metric ϵ tells us how well the left and right sides of the MACE equations agree.

V. RESULTS

Using the DH simulation environment described above, we generated test data with SNR $= [20,\ 10,\ 5,\ 2]$ and $D/r_0 = [0,\ 5,\ 10,\ 15]$ for each of the ten images ($4 \times 4 \times 10 = 160$ total realizations). In Figs. 5 and 6, we show example reconstructions for two of the ten test images, as a function of D/r_0. On the top row, we show the corrupted images formed by a simple back projection with no phase correction, given by $|A_0^H y|^{\circ 2}$. We show the DH-MBIR and C-PnP reconstructions in the middle and bottom rows, respectively. Additionally, we provide the true reflectance functions in Fig. 4. These results show a stark difference between the two algorithms. Similar to Fig. 2, the DH-MBIR reconstructions contain high-spatial-frequency variations that resemble salt and pepper noise. We conjecture that these artifacts may be caused by extremely dark and bright speckles in the initial estimate which are not regularized sufficiently by the QGGMRF model. Conversely, the C-PnP algorithm produces much more-natural looking images with no noticeable speckle variations.

While Figs. 5–6 show a qualitative improvement in reconstruction quality when using C-PnP, we also provide a quantitative comparison in Figs. 7–8. In Fig. 7, we plot image quality, measured using SSIM, as a function of the phase-error strength, D/r_0, for the four different SNR values. The results show that C-PnP produces higher quality images when compared with DH-MBIR. In fact, C-PnP produces a higher SSIM in all cases. Furthermore, over the 160 realizations, the average C-PnP SSIM value was $2.2\times$ higher than the average DH-MBIR SSIM. This doubling of image quality is significant; however, it is not

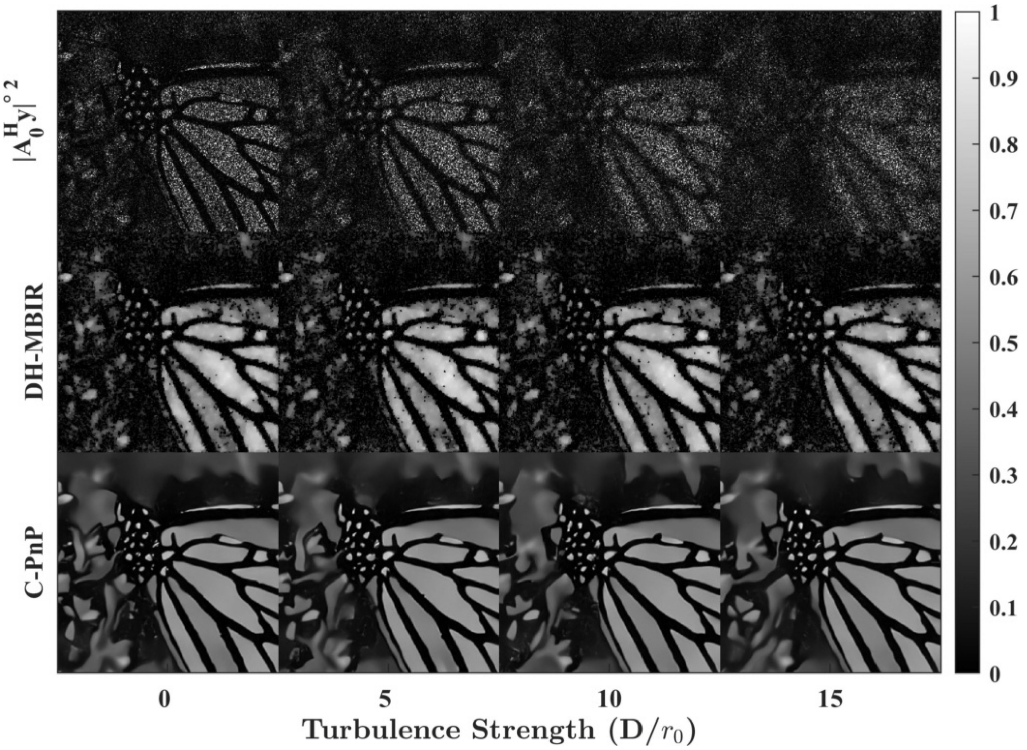

Fig. 5. Example results as a function of phase-error strength, D/r_0, for SNR=10. The top row, labeled $|A_0^H y|^{\circ 2}$, shows a back projection of the data with no phase error correction. The middle row shows the DH-MBIR reconstructions obtained using an analytical QGGMRF image model. These DH-MBIR images contain artifacts in the form of high-spatial-frequency variations. The bottom row shows the C-PnP reconstructions obtained using a CNN image model. The C-PnP images more-closely resemble the true reflectance function shown in Fig. 4.

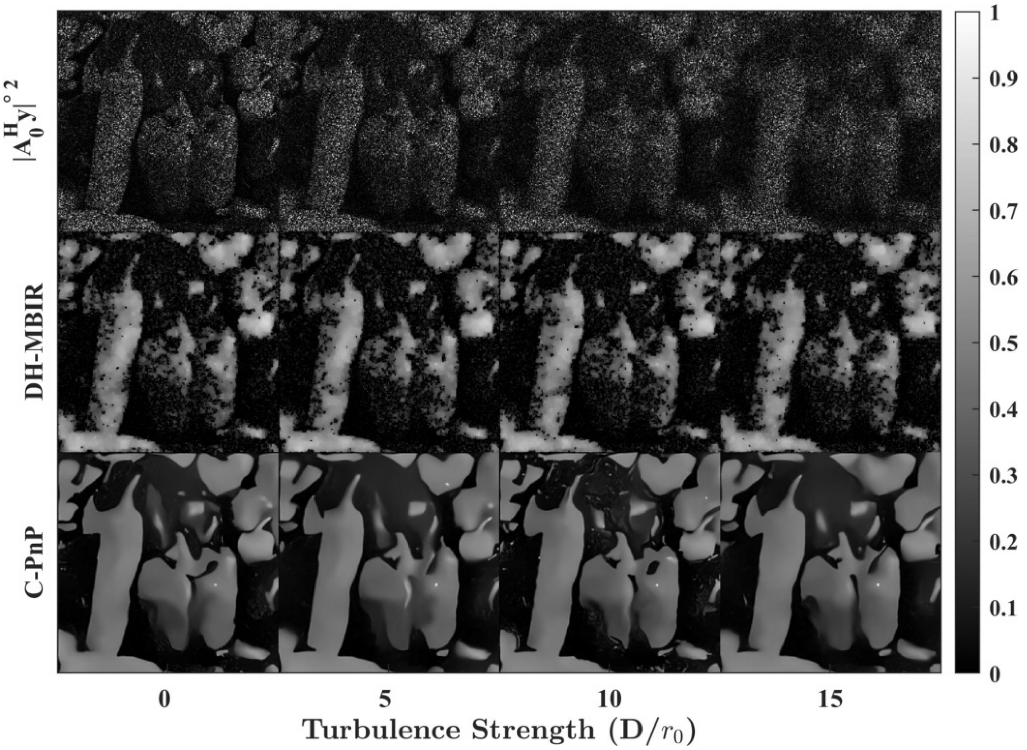

Fig. 6. Example results as a function of phase-error strength, D/r_0, for SNR=10. The top row, labeled $|A_0^H y|^{\circ 2}$, shows a back projection of the data with no phase error correction. The middle row shows the DH-MBIR reconstructions obtained using an analytical QGGMRF image model. These DH-MBIR images contain artifacts in the form of high-spatial-frequency variations. The bottom row shows the C-PnP reconstructions obtained using a CNN image model. The C-PnP images more-closely resemble the true reflectance function shown in Fig. 4.

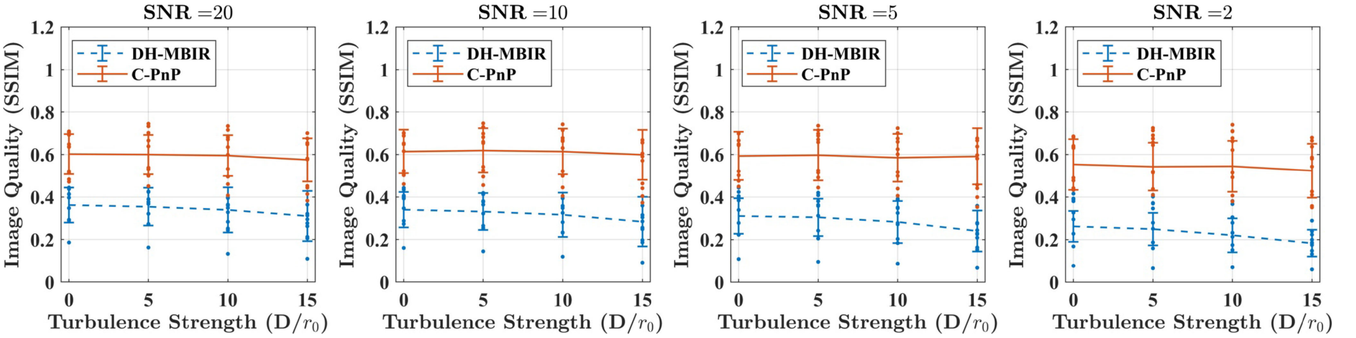

Fig. 7. Image reconstruction quality as a function of turbulence strength, D/r_0, for SNRs of 20, 10, 5, and 2. Here, the lines show the average value taken over ten different images and the error bars show the corresponding standard deviation. We also plot each data point to show the distribution of values. The results show that, by incorporating a more-advanced image model learned by a CNN, we obtain significantly higher quality images.

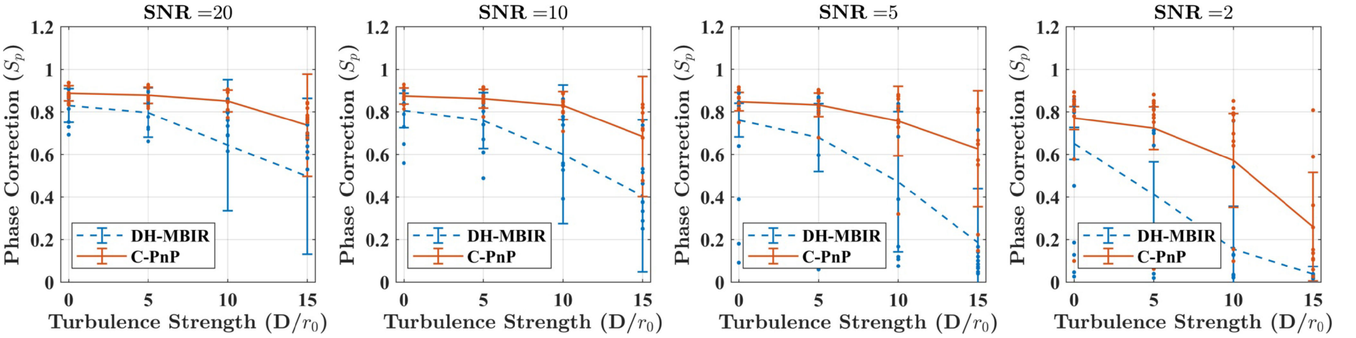

Fig. 8. Degree of phase-error correction, S_p, as a function of turbulence strength, D/r_0, for SNRs of 20, 10, 5, and 2. Here, the lines show the average value taken over ten different images and the error bars show the corresponding standard deviation. We also plot each data point to show the distribution of values. These plots show that, in a joint estimation framework, a more-sophisticated image model results in higher-quality phase-error corrections.

surprising given that we designed the algorithm to incorporate a more-advanced image model.

In Fig. 8, we plot the quality of the phase-error correction, measured using S_p, as a function of the phase-error strength, D/r_0, for the four different SNR values. These plots show a less-intuitive result in the quality of the phase-error estimate. Specifically, C-PnP produces a higher peak Strehl ratio in nearly all cases. Over the 160 realizations, the average C-PnP peak Strehl ratio was 2.9× higher than the average DH-MBIR peak Strehl ratio. While this difference is less pronounced for lower values of D/r_0, it becomes more significant as the turbulence strength increases. This nearly 3× improvement in average peak Strehl ratio may be surprising since the two algorithms both estimate ϕ using the same prior model, $p(\phi)$. However, since we are jointly estimating the two quantities using an iterative process, a better estimate of r results in a better estimate of ϕ. The two are jointly tied together through the EM surrogate function.

In addition to producing more-accurate reconstructions, we also find that the C-PnP algorithm consistently produces results that solve the MACE equations given in Eq. (22). Figure 9 shows the normalized MACE equation error, ϵ, for all 160 reconstructions. As shown in the plot, we found that SNR is a major factor in how well the output of the C-PnP algorithm solves the MACE equations. However, even for the cases where SNR $= 2$, ϵ was still less than 0.03, and the highest error measured was 0.042. This error metric decreased by several orders of magnitude as the SNR increased. Thus, we surmise that the C-PnP algorithm consistently solves the MACE equations, as defined in Eq. (22).

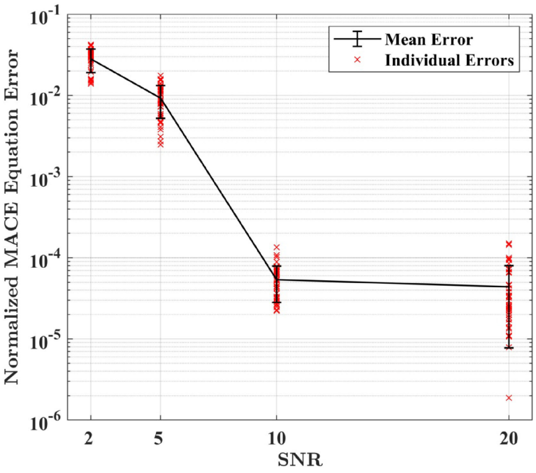

Fig. 9. Normalized MACE equation error, ϵ, as a function of SNR for all 160 reconstructions. These results show that the C-PnP algorithm consistently solves the MACE equations given in Eq. (22).

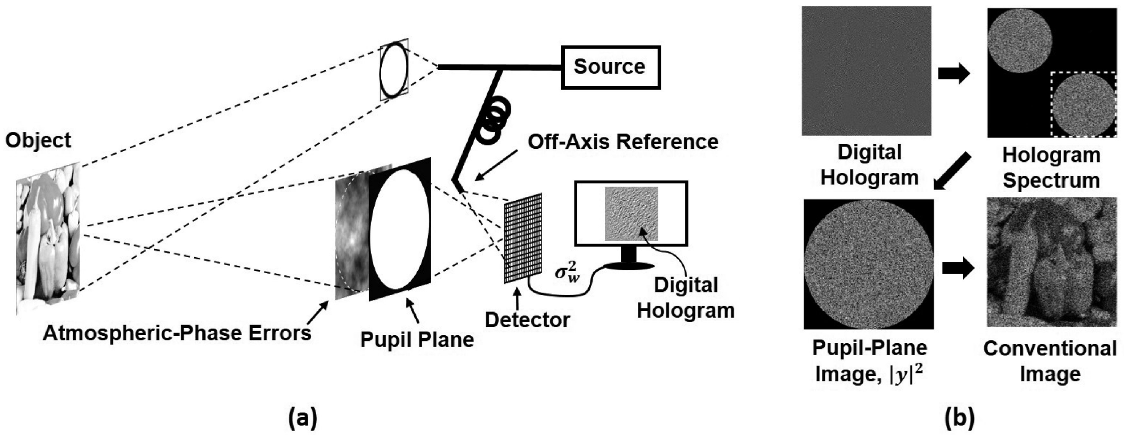

Fig. 10. Overview of DH system using an off-axis IPRG. (a) We use a laser source to flood illuminate an object. In this example, atmospheric turbulence corrupts the the scattered signal light, resulting in isoplanatic phase errors, ϕ. The scattered light passes through the pupil-aperture function, a, where it becomes collimated. From the pupil plane, we focus the collimated light onto a detector array where we mix it with a reference field to form the hologram. In (b), we show conventional image-processing steps for a DH system using an off-axis IPRG. Here, we show the magnitude squared of any complex-valued quantities. To extract the signal, we start with a real-valued digital hologram (top left), then take a 2D FFT to obtain the complex-valued holographic spectrum (top right). Note that the digitized hologram has low contrast. Next, we filter and demodulate a section of the spectrum (bottom left). This subset of the spectrum, y, represents a complex image of the signal field in the pupil plane. Finally, for basic image formation, we take an inverse FFT (IFFT) of y to form a complex image (bottom right).

VI. Conclusion

In conclusion, we presented a new approach for single-shot DH-image reconstruction that couples a CNN-based image model with stochastic physics-based models. Using the EM algorithm, we extended the MACE framework to overcome the intractable forward model associated with reconstructing real-valued speckle-free images from coherent data. We also extended the framework to include the joint estimation of atmospheric-phase errors. We designed three agents, two image agents that ensure that the resulting images are both natural looking and are consistent with the data we measured, and a phase-error agent that ensures we produce a focused image. Our approach balances the influence of these three agents to produce a comprehensive framework capable of overcoming a combination of speckle variations, measurement noise, and phase errors.

When compared to the DH-MBIR algorithm from [11], C-PnP produces significantly higher-quality images. Over the range of images, SNRs, and turbulence strengths we considered, C-PnP consistently produces reconstructions with higher SSIM values. On average, the C-PnP SSIM was 2.2 times higher than the DH-MBIR SSIM value. This improvement in image quality also lead to an improvement in phase-error estimation. On average, the C-PnP peak Strehl ratio was 2.9 times higher than the DH-MBIR peak Strehl ratio, even though the two algorithms use the same approach for jointly estimating ϕ. This surprising result shows that improved image estimates, as part of a joint reconstruction framework, lead to better atmospheric-phase-error estimates.

Appendix A
Review of Digital Holography Sensor Model

In this section, we provide a brief overview of the DH sensor concept and measurement model. For a more-comprehensive explanation, we direct the reader to [11], [35]. Fig. 10(a) illustrates an example DH system using an off-axis image-plane recording geometry (IPRG) [11], [35]. The field reflected by the object encounters unknown atmospheric-phase errors, ϕ. Note that ϕ is a high-dimensional phase function caused by variations in the index of refraction of the atmosphere[36]. After reaching the sensor, the reflected field is imaged onto a detector array where it is mixed with a reference field, to form the digital hologram [1]. The intensity of the interference pattern obtained when mixing our signal field, S, with our reference field, R, is given by

$$\begin{aligned} I &= |S+R|^2, \\ &= |S|^2 + |R|^2 + S^*R + SR^*, \end{aligned} \tag{33}$$

where * indicates a the complex conjugate operator. In Eq. (33), the first two terms are commonly known as the zero-order terms and last two terms are known as the cross terms. The cross terms are of particular interest since they contain information about the amplitude and phase of the complex-valued signal. By offsetting our reference from the primary axis of propagation, we modulate the interference pattern in a way that makes separating the four terms in Eq. (33) trivial [10], [11].

Fig. 10(b) shows the basic processing steps used to recover complex image information from the real-valued digital hologram [11]. We first extract a discrete version of the complex-valued signal from the spectrum of the digital hologram. The top right plot in Fig. 10(b) shows the hologram spectrum where the cross terms appear as two diagonally-separated circles. Note that we suppressed the zero-order terms for easier viewing. We spatially filter one of the cross term to obtain the complex image spectrum, $y \in \mathbb{C}^M$, shown in the bottom left plot of Fig. 10(b). Due to the discrete Fourier transform (DFT) relationship between the pupil plane and the image/detector array, this spectral information, y, is a complex image of the signal at the pupil plane. Finally, we can form a basic image by digitally back propagating y to the image plane using the appropriate Fresnel or Fraunhofer diffraction model and then taking the magnitude squared.

In our model for the complex-valued data, y, given by Eq. (1), we assume our DH sensor is shot-noise-limited and we model our measurement noise as

$$p(w) \sim CN(0, \sigma_w^2 I, 0), \tag{34}$$

where, I is the identity matrix and σ_w^2 is the variance of the measurement noise.

Using Eqs. (1), (3) and (34), we model the data from our DH sensor as a conditionally complex Gaussian random variable with distribution

$$p(y|r,\phi) \sim CN(0, C_{y|r,\phi}, 0), \tag{35}$$

where $C_{y|r,\phi}$ is a dense covariance matrix given by

$$C_{y|r,\phi} = A_\phi D(r) A_\phi^H + \sigma_w^2 I, \tag{36}$$

and the superscript H indicates the Hermitian transpose.

Appendix B
Theorems With Proof

Theorem III.1: For the agents defined in Eq. (21), any solution, $[\mathbf{r}^*, \phi^*]$ to the EM-based MACE equations given by Eq. (22) also has the property that $\nabla_{r,\phi} f_{\sum}(\bar{r}^*, \phi^*) = 0$, where $\mathbf{r}^* = [r_1^{*T}, r_2^{*T}]^T$ and $\bar{r}^* = (r_1^* + r_2^*)/2$.

Proof: If $[\mathbf{r}^*, \phi^*]$ is a solution to the EM-based MACE equations given by Eq. (22), then

$$\nabla_r Q_r(\bar{r}^*; \bar{r}^*, \phi^*) + \frac{1}{\sigma^2}\left(\bar{r}^* - r_1^*\right) = 0, \tag{37}$$

$$\nabla_r \beta f_2(\bar{r}^*) + \frac{1}{\sigma^2}\left(\bar{r}^* - r_2^*\right) = 0, \tag{38}$$

$$\nabla_\phi Q_\phi(\phi^*; \bar{r}^*, \phi^*) + \nabla_\phi f_3(\phi^*) = 0, \tag{39}$$

From the separable nature of Q, and from the gradient property of surrogate functions [30], we know that at the point of approximation, $\bar{r}^*, \phi^*$,

$$\begin{aligned}\nabla_r Q_r(\bar{r}^*; \bar{r}^*, \phi^*) &= \nabla_r Q(\bar{r}^*, \phi^*; \bar{r}^*, \phi^*) = \nabla_r f_1(\bar{r}^*, \phi^*),\\ \nabla_\phi Q_\phi(\phi^*; \bar{r}^*, \phi^*) &= \nabla_\phi Q(\bar{r}^*, \phi^*; r^*, \phi^*) = \nabla_\phi f_1(\bar{r}^*, \phi^*).\end{aligned} \tag{40}$$

Substituting Eq. (40) into Eqs. (37–39)

$$\nabla_r f_1(\bar{r}^*, \phi^*) + \frac{1}{\sigma^2}\left(\bar{r}^* - r_1^*\right) = 0, \tag{41}$$

$$\nabla_r \beta f_2(\bar{r}^*) + \frac{1}{\sigma^2}\left(\bar{r}^* - r_2^*\right) = 0, \tag{42}$$

$$\nabla_\phi f_1(\bar{r}^*, \phi^*) + \nabla_\phi f_3(\phi^*) = 0, \tag{43}$$

Adding Eqs. (41) and (42), we get

$$\nabla_r f_1(\bar{r}^*, \phi^*) + \nabla_r \beta f_2(\bar{r}^*) + \frac{2}{\sigma^2}\left(\bar{r}^* - \frac{r_1^* + r_2^*}{2}\right) = 0. \tag{44}$$

Next, we use $\bar{r}^* = (r_1^* + r_2^*)/2$ to simplify Eq. (44), leaving

$$\nabla_r f_1(\bar{r}^*, \phi^*) + \nabla_r \beta f_2(\bar{r}^*) = 0. \tag{45}$$

When Eqs. (43) and (45) are true, then

$$\nabla_\phi f_{\sum}(\bar{r}^*, \phi^*) = 0, \tag{46}$$

and

$$\nabla_r f_{\sum}(\bar{r}^*, \phi^*) = 0, \tag{47}$$

respectively. Finally, it is also true that

$$\nabla_{r,\phi} f_{\sum}(\bar{r}^*, \phi^*) = 0. \tag{48}$$

■

Theorem III.2: When the update steps in Eq. (24) converge to a fixed point, $[\mathbf{w}^*, \phi^*, \mathbf{r}^*]$, where $\mathbf{w}^* = [w_1^{*T}, w_2^{*T}]^T$ and $\mathbf{r}^* = [r_1^{*T}, r_2^{*T}]^T$, the fixed point is also a solution to the MACE equations defined in Eq. (22).

Proof: If $[\mathbf{r}^*, \mathbf{w}^*, \phi^*]$ is a fixed point of the update steps defined in Eq. (24), then

$$\mathbf{w}^* = \begin{bmatrix}\tilde{F}_1(r_1^*; \bar{r}^*, \phi^*)\\ F_2(r_2^*)\end{bmatrix} \tag{49a}$$

$$\phi^* = \tilde{F}_3(\bar{r}^*, \phi^*), \tag{49b}$$

$$\mathbf{r}^* = \mathbf{r}^* + 2\rho\left[\mathbf{G}\left(2\mathbf{w} - \mathbf{r}^*\right) - \mathbf{w}\right]. \tag{49c}$$

From Eq. (49c), we get

$$\mathbf{G}\left(2\mathbf{w}^* - \mathbf{r}^*\right) = \mathbf{w}^*. \tag{50}$$

Using $\mathbf{w} = [w_1^{*T}, w_2^{*T}]^T$, $\mathbf{r}^* = [r_1^{*T}, r_2^{*T}]^T$, and our definition for the averaging operator, $\mathbf{G}$, we rewrite Eq. (50) as

$$\begin{aligned}\frac{1}{2}\begin{bmatrix}(2w_1^* - r_1^* + 2w_2^* - r_2^*)\\ (2w_1^* - r_1^* + 2w_2^* - r_2^*)\end{bmatrix} &= \begin{bmatrix}w_1^*\\ w_2^*\end{bmatrix},\\ \begin{bmatrix}w_1^* + w_2^* - \frac{r_1^*+r_2^*}{2}\\ w_1^* + w_2^* - \frac{r_1^*+r_2^*}{2}\end{bmatrix} &= \begin{bmatrix}w_1^*\\ w_2^*\end{bmatrix},\\ \begin{bmatrix}w_2^*\\ w_1^*\end{bmatrix} &= \begin{bmatrix}\frac{r_1^*+r_2^*}{2}\\ \frac{r_1^*+r_2^*}{2}\end{bmatrix}.\end{aligned} \tag{51}$$

Next, Eq. (49a), $\bar{r}^* = (r_1^* + r_2^*)/2$, and Eq. (51) together results in the following relationship

$$\begin{bmatrix}\tilde{F}_1(r_1^*; \bar{r}^*, \phi^*)\\ F_2(r_2^*)\end{bmatrix} = \begin{bmatrix}\bar{r}^*\\ \bar{r}^*\end{bmatrix}, \tag{52}$$

Finally, we include Eq. (49b) to get

$$\begin{bmatrix}\tilde{F}_1(r_1^*; \bar{r}^*, \phi^*)\\ F_2(r_2^*)\\ \tilde{F}_3(\bar{r}^*, \phi^*)\end{bmatrix} = \begin{bmatrix}\bar{r}^*\\ \bar{r}^*\\ \phi^*\end{bmatrix}, \tag{53}$$

which are the MACE equations defined in Eq. (22) ■

Appendix C
Update Equations

In this section, we provide details about how we establish the surrogate function and we provide the update equations used to implement our agents.

A. E-Step: Update for Surrogate Function

To evaluate the expectation with respect to g in Eq. (19), we start by defining the conditional posterior distribution of g. In [9], we showed that this distribution is complex Gaussian with mean

$$\mu = C\frac{1}{\sigma_w^2}A_{\phi'}^H y, \tag{54}$$

and covariance

$$C = \left[\frac{1}{\sigma_w^2}A_{\phi'}^H A_{\phi'} + D(r')^{-1}\right]^{-1} \approx D\left(\frac{\sigma_w^2}{1+\frac{\sigma_w^2}{r'}}\right). \tag{55}$$

The approximation in Eq. (55) assumes that $A_{\phi'}^H A_{\phi'} \approx I$. In practice, we have found this approximation to work well [9]–[11]. Furthermore, we can modify the spatial-filtering of the hologram spectrum to make the approximation exact [10]. However, this modification reduces the size of the data, y.

Using Eqs. (54–55), we can write Eq. (19) as

$$\begin{aligned} Q(r,\phi;r',\phi') &= Q_r(r;r',\phi') + Q_\phi(\phi;r',\phi') + c \\ &= -\frac{1}{\sigma_w^2}2\mathrm{Re}\left\{y^H A_\phi \mu\right\} \\ &\quad + \sum_{i=1}^{N}\frac{1}{r_i}\left(C_{i,i} + |\mu_i|^2\right) + \log|D(r)| + c \end{aligned} \tag{56}$$

where μ_i is the i^{th} element of the posterior mean, $C_{i,i}$ is the i^{th} diagonal element of the posterior covariance, and c is constant with respect to r and ϕ.

B. M-Step: Update for $\tilde{F}_1$

For the EM-based image agent, $\tilde{F}_1$, there is no coupling between the elements of r. Therefore, we update each element, r_s, independently of the others with a closed-form solution. Starting with Eq. (21a), we write the 1D cost function for the s^{th} element as [11], [21]

$$c\left(r_s; r_1, r', \phi'\right) = \log r_s + \frac{C_{s,s} + |\mu_s|^2}{r_s} + \frac{1}{2\sigma_1^2}\left(r_s^2 - 2r_s r_{1,s}\right), \tag{57}$$

where $r_{1,s}$ is the s^{th} element of the input state vector, r_1. Note that Eq. (57) depends on r' and ϕ' because C and μ depend on r' and ϕ'.

To minimize Eq. (57), we differentiate with respect to r_s, set the left-hand side equal to zero, and multiply both sides by r_s^2. The result is a 3^{rd} order polynomial given by

$$0 = \alpha_1 r_s^3 + \alpha_2 r_s^2 + \alpha_3 r_s + \alpha_4, \tag{58}$$

where

$$\alpha_1 = \frac{1}{\sigma_1^2},\ \alpha_2 = \frac{-r_{1,s}}{\sigma_1^2},\ \alpha_3 = 1,\ \alpha_4 = -\left(C_{i,i} + |\mu_i|^2\right). \tag{59}$$

Thus, we update each element, r_s, by simply rooting Eq. (58). The cubic equation provides a general closed-form solution for the roots of a 3^{rd} order polynomial. If multiple real-valued roots exist, we use Eq. (57) to determine the root with the lowest cost.

C. M-Step: Update for $\bar{\tilde{F}}_3$

Similar to the reflectance update, we also use ICD to update the phase errors. Furthermore, we use a nearest-neighbors interpolation matrix, $P \in \mathbb{R}^{M\times L}$, as described in Section III-C3, that maps $\phi \in \mathbb{R}^L$ to the full $M \times M$ size required for use in the forward model. Our 1D cost function for element, ϕ_s, is given by

$$c(\phi_s; r', \phi') = -|\chi_s|\cos\left(\angle_s\chi - \phi_s\right) + \frac{1}{2\sigma_\phi^2}\sum_{j\in\partial s} b_{i,j}|\phi_s - \bar{\phi}_j|^2, \tag{60}$$

where

$$\chi_s = \frac{2}{\sigma_w^2}\left[P^T\left(y^H A_0 \mu\right)\right]_s. \tag{61}$$

In Eq. (60), $j \in \partial s$ is an index over neighboring phase samples and $\angle\chi$ indicates the the phase of the complex-valued scalar, χ. The operator A_0 indicates we apply A using zeros for ϕ. This acts to propagate μ to the pupil plane, but does not apply the phase errors. Intuitively, χ, is the difference between the fields on either side of the phase screen, summed over all high-resolution points that correspond to the single low-resolution point, ϕ_s, being updated.

Acknowledgment

Views expressed in this paper are those of the authors and do not reflect the official policy of the Air Force, the Department of Defense, or the U.S. Government. The authors would like to thank Dr. Douglas E. Thornton for his assistance in with this work.

References

[1] T.-C. Poon and J.-P. Liu, *Introduction to Modern Digital Holography: With MATLAB*. New York, NY, USA: Cambridge Univ. Press, 2014.

[2] S. T. Thurman and J. R. Fienup, "Phase-error correction in digital holography," *J. Opt. Soc. Amer. A*, vol. 25, no. 4, pp. 983–994, 2008.

[3] S. T. Thurman and J. R. Fienup, "Correction of anisoplanatic phase errors in digital holography," *J. Opt. Soc. Amer. A*, vol. 25, no. 4, pp. 995–999, 2008.

[4] J. C. Marron, R. L. Kendrick, N. Seldomridge, T. D. Grow, and T. A. Höft, "Atmospheric turbulence correction using digital holographic detection: Experimental results," *Opt. Express*, vol. 17, no. 14, pp. 11 638–11 651, 2009.

[5] A. E. Tippie and J. R. Fienup, "Multiple-plane anisoplanatic phase correction in a laboratory digital holography experiment," *Opt. Lett.*, vol. 35, no. 19, pp. 3291–3293, 2010.

[6] S. T. Thurman, "Phase-error correction in digital holography using single-shot data," *J. Opt. Soc. Amer. A*, vol. 36, no. 12, pp. D47–D61, 2019.

[7] S. Sotthivirat and J. A. Fessler, "Penalized-likelihood image reconstruction for digital holography," *J. Opt. Soc. Amer. A*, vol. 21, no. 5, pp. 737–750, 2004.

[8] D. J. Lee, C. A. Bouman, and A. M. Weiner, "Single shot digital holography using iterative reconstruction with alternating updates of amplitude and phase," *Electron. Imag.*, vol. 2016, no. 19, pp. 1–6, Sep. 2016.

[9] C. J. Pellizzari, M. F. Spencer, and C. A. Bouman, "Phase-error estimation and image reconstruction from digital-holography data using a Bayesian framework," *J. Opt. Soc. Amer. A*, vol. 34, no. 9, pp. 1659–1669, 2017.

[10] C. Pellizzari, M. T. Banet, M. F. Spencer, and C. A. Bouman, "Demonstration of single-shot digital holography using a Bayesian framework," *J. Opt. Soc. Amer. A*, vol. 35, no. 1, pp. 103–107, 2018.

[11] C. J. Pellizzari, M. F. Spencer, and C. A. Bouman, "Imaging through distributed-volume aberrations using single-shot digital holography," *J. Opt. Soc. Amer. A*, vol. 36, no. 2, pp. A20–A33, 2019.

[12] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, 2013, pp. 945–948.

[13] S. H. Chan, X. Wang, and O. A. Elgendy, "Plug-and-Play ADMM for image restoration: Fixed point convergence and applications," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 84–98, 2016.

[14] U. S. Kamilov, H. Mansour, and B. Wohlberg, "A plug-and-play priors approach for solving nonlinear imaging inverse problems," *IEEE Signal Process. Lett.*, vol. 24, no. 12, pp. 1872–1876, Dec. 2017.

[15] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (red)," *SIAM J. Imag. Sci.*, vol. 10, no. 4, pp. 1804–1844, 2017.

[16] G. T. Buzzard, S. H. Chan, S. Sreehari, and C. A. Bouman, "Plug-and-play unplugged: Optimization-free reconstruction using consensus equilibrium," *SIAM J. Imag. Sci.*, vol. 11, no. 3, pp. 2001–2020, 2018.

[17] T. He, Y. Sun, B. Chen, J. Qi, W. Liu, and J. Hu, "Plug-and-play inertial forward–backward algorithm for poisson image deconvolution," *J. Electron. Imag.*, vol. 28, no. 4, Jun. 2019, Art. no. 043020.

[18] M. T. McCann, K. H. Jin, and M. Unser, "Convolutional neural networks for inverse problems in imaging: A review," *IEEE Signal Proc. Mag.*, vol. 34, no. 6, pp. 85–95, Nov. 2017.

[19] G. Barbastathis, A. Ozcan, and G. Situ, "On the use of deep learning for computational imaging," *Optica*, vol. 6, no. 8, pp. 921–943, 2019.

[20] H. K. Aggarwal, M. P. Mani, and M. Jacob, "Modl: Model-based deep learning architecture for inverse problems," *IEEE Trans. Med. Imag.*, vol. 38, no. 2, pp. 394–405, Feb. 2019.

[21] C. J. Pellizzari, M. F. Spencer, and C. A. Bouman, "Optically coherent image reconstruction in the presence of phase errors using advanced-prior models," in *Long-Range Imag. III*, 2018, Art. no. 106500B.

[22] C. Pellizzari *et al.*, "Optically coherent image formation and denoising using a plug and play inversion framework," *Appl. Opt.*, vol. 56, no. 16, pp. 4735–4744, 2017.

[23] S. Sreehari, S. Venkatakrishnan, B. Wohlberg, L. F. Drummy, J. P. Simmons, and C. A. Bouman, "Plug-and-play priors for bright field electron tomography and sparse interpolation," *IEEE Trans. Comput. Imag.*, vol. 2, no. 4, pp. 408–423, 2016.

[24] S. Majee, T. Balke, C. A. Kemp, G. T. Buzzard, and C. A. Bouman, "4D x-ray CT reconstruction using multi-slice fusion," in *Proc. IEEE Int. Conf. Comput. Photogr.*, 2019, pp. 1–8.

[25] M. U. Ghani and W. C. Karl, "Integrating data and image domain deep learning for limited angle tomography using consensus equilibrium," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop.*, 2019, pp. 3922–3932.

[26] X. Wang *et al.*, "Consensus equilibrium framework for super-resolution and extreme-scale ct reconstruction," in *Proc. Int. Conf. High Perform. Comput., Netw., Storage Anal.*, 2019, pp. 1–23.

[27] V. Sridhar, X. Wang, G. T. Buzzard, and C. A. Bouman, "Distributed iterative CT reconstruction using multi-agent consensus equilibrium," *IEEE Trans. Comput. Imag.*, 2020.

[28] J. W. Goodman, *Speckle Phenomena in Optics, Theory, and Applications*. Englewood, Colorado, USA: Roberts and Company, 2006.

[29] C. J. Pellizzari *et al.*, "Synthetic aperature ladar: A model-based approach," *IEEE Trans. Comput. Imag.*, vol. 3, no. 4, pp. 901–916, Dec. 2017.

[30] C. A. Bouman, "Model based image processing," 2013. [Online]. Available: https://engineering.purdue.edu/ bouman/publications/pdf/MBIP-book.pdf

[31] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[32] "KerasDnCNN - keras implementation of DnCNN-S," Accessed: Sep. 20, 2018. [Online]. Available: https://github.com/aGIToz/KerasDnCNN

[33] J. D. Schmidt, *Numerical Simulation of Optical Wave Propagation, With Examples in Matlab*. Bellingham, WA, USA: SPIE, 2010.

[34] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 47–57, Mar. 2017.

[35] M. F. Spencer, R. A. Raynor, M. T. Banet, and D. K. Marker, "Deep-turbulence wavefront sensing using digital-holographic detection in the off-axis image plane recording geometry," *Opt. Eng.*, vol. 56, no. 3, 2017, Art. no. 031 213.

[36] L. Andrews and R. Phillips, *Laser Beam Propagation through Random Media*, 2nd ed. Bellingham, WA, USA: SPIE, 2005.