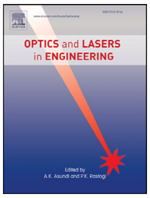
Contents lists available at ScienceDirect

Optics and Lasers in Engineering

journal homepage: www.elsevier.com/locate/optlaseng



Coherent plug-and-play artifact removal: Physics-based deep learning for imaging through aberrations

Casey J. Pellizzari[a,*], Timothy J. Bate[a], Kevin P. Donnelly[a], Gregery T. Buzzard[b], Charles A. Bouman[c], Mark F. Spencer[d]

[a] *Department of Physics and Meteorology, United States Air Force Academy, USAFA, CO, USA*
[b] *Department of Mathematics, Purdue University, West Lafayette, IN, USA*
[c] *Departments of Electrical and Computer Engineering, and Biomedical Engineering, Purdue University, West Lafayette, IN, USA*
[d] *Air Force Research Laboratory, Directed Energy Directorate, Kirtland Air Force Base, NM, USA*

ARTICLE INFO

Keywords:
Coherent detection
Digital holography
Computational imaging
Deep learning
Convolutional neural network
Speckle phenomena

ABSTRACT

Current image-correction frameworks for sensors that employ optically coherent detection attempt to estimate phase errors in the data, like those caused by aberrations, and simultaneously reconstruct digitally enhanced images. In practice, these frameworks struggle to account for the effects of speckle. To address this gap, we develop a novel image-correction framework referred to as Coherent Plug-and-Play Artifact Removal (CPnP-AR), which couples a neural-network despeckler with a physics-based measurement model. We also develop the experimental protocol needed to quantitatively evaluate performance relative to multiple state-of-the-art frameworks. The results show that CPnP-AR produces higher-quality images and more accurate estimates of the phase errors for a wide range of objects, specifically without the need for object-dependent parameter tuning. This improvement in overall robustness is a key step towards employing this novel image-correction framework for numerous applications of interest.

1. Introduction

Sensors that employ optically coherent detection measure the amplitude and phase of a scattered signal. In practice, phase errors in the data, like those caused by aberrations, degrade one's ability to use this data for the purposes of imaging. Therefore, we use image-correction frameworks to estimate and remove these phase errors and produce digitally-corrected images.

In addition to phase errors in the data, the use of coherent illumination unfortunately introduces speckle, which manifest as high-spatial-frequency artifacts in images. Practically speaking, fully formed speckle occurs when the surface-height standard deviation of the object is greater than half the wavelength of the coherent illumination [8]. When this occurs, the reflected optical field has a seemingly random phase at each point. This random phase leads to constructive or destructive interference when we integrate the amplitude and phase of the signal over a finite region. Images formed from the intensity of these reflected optical fields contain exponentially distributed multiplicative noise that we call speckle [8]. This is the case for numerous applications of interest, and the resulting speckle limits the effectiveness of image-correction frameworks.

A straightforward but measurement-intensive approach to reduce the effects of speckle is to average multiple, independent speckle realizations. This approach assumes that the object is relatively static between measurements but that each measurement has an independent speckle realization. We may obtain such data through small relative movements between the object and the sensor or by changing the illumination wavelength. However, such multi-shot data may not always be available. For example, when the object or phase errors are changing rapidly, or if multiple wavelengths are not available, speckle averaging is not possible. In those cases, we may only obtain a single measurement in which the object and turbulence are static. In this paper, we focus on this constraining case to produce high-quality images from single-shot data.

Speckle averaging, nonetheless, gives insight into the underlying problem at hand. In general, we model the spatially discretized version of the scattered signal, known as the reflection coefficient, $g \in \mathbb{C}^M$, as a complex-valued, zero-mean Gaussian random variable [8]. A speckled image is then formed from the observed amplitude squared, $|g|^2$. However, the expected value of $|g|^2$, known as the reflectance, $r \in \mathbb{R}^M$, does not contain speckle and is typically smoother than $|g|^2$ [24]. Simply put, r is analogous to the real-valued quantity that we observe in incoherent images and hence is the speckle-free quantity of interest in this paper.

[*] Corresponding author.
E-mail address: casey.pellizzari@afacademy.af.edu (C.J. Pellizzari).

https://doi.org/10.1016/j.optlaseng.2023.107496
Received 6 December 2022; Received in revised form 11 January 2023; Accepted 20 January 2023

In previous works, we developed image-correction frameworks that use a model-based Bayesian framework to find the joint *maximum a posterior* (MAP) estimate of the phase errors and r from single-shot measurements. These frameworks include Synthetic Aperture Lidar Model-Based Iterative Reconstruction (SAL-MBIR) [19,24], and Digital Holography MBIR (DH-MBIR) [18,20,22]. When compared to existing image-correction frameworks, SAL-MBIR and DH-MBIR produce higher-quality images and more accurate estimates of the phase errors with less data and at lower signal levels. However, both frameworks use relatively simple analytical functions to model the distribution of reflectances, $p(r)$, which leads to artifacts in the estimates of reasonably complex images [21,23].

Alternatively, the image-correction framework referred to as Coherent Plug-and-Play (CPnP) [23] uses a neural network image prior to further reduce artifacts. CPnP uses a generalization of Bayesian inversion based on Plug-and-Play (PnP) methods [6,7,9,11,12,28,37], in which a black-box denoiser, such as a convolutional neural network (CNN) trained to remove additive white Gaussian noise (AWGN), is used as an image prior. CPnP produces higher-quality images and more accurate estimates of the phase errors than the MBIR frameworks found in Refs. [18,20,22,24]; however, CPnP is limited in the type of image prior that can be used, and the resulting reconstructions still show residual artifacts caused by speckle.

In CPnP and other PnP-based methods, a neural network denoiser is applied iteratively in place of one of the proximal maps associated with the alternating direction method of multipliers (ADMM) algorithm, which is the foundation for PnP. A difficulty with using a pre-trained denoiser is that the intermediate images in the iterative loop often do not have the same distribution as that used to train the denoiser. To improve the empirical performance of PnP, one approach is to modify the denoiser during application. This can be done by changing the assumed noise level in each iteration as in Chan et al. [7] or by using additional input-specific training as in Tirer and Giryes [36]. These methods require significant parameter tuning or on-the-fly training. An alternative method in Liu et al. [15] uses a CNN trained as part of a deep unrolling (DU) architecture that includes the forward model in end-to-end training [11]. However, this method requires that the CNN be trained together with the forward model, destroying the modularity of PnP that allows for changes to the forward model without changing or retraining the prior model.

In this paper, we develop a new PnP image-correction framework for coherent imaging that allows us to move beyond the limitations of AWGN-denoiser prior models while still maintaining the benefit of PnP modularity. Our framework, referred to as CPnP-Artifact Removal (CPnP-AR), uses a CNN trained to remove speckle instead of AWGN and applies this so-called “despeckler” to input images using a novel resampling approach. Through this resampling approach, CPnP-AR maintains the modularity of PnP in that the despeckling CNN is trained on natural images without reference to the forward model. In addition, CPnP-AR better matches the distribution of images in the PnP loop to the distribution of training images by resampling before despeckling. That is, instead of applying the despeckler directly on an image estimate in the PnP loop, first this image estimate is used to generate a speckled image by mimicking the speckle-formation process. Only then is the despeckler applied and the rest of the loop continued. Not only does the resulting framework significantly improve image quality compared to existing frameworks, we find that CPnP-AR can better regularize a wide-range of image types with fixed hyperparameters. Thus, CPnP-AR is more robust to variations in the data than existing frameworks.

Another key contribution of this paper is the development of an experimental protocol, conducted at optical wavelengths using digital holography (DH), to obtain high-quality estimates of the true reflectances and phase errors. Such estimates are difficult to measure experimentally due to uncertainties in illumination uniformity, object complexity, and detection sensitivity. The use of such estimates as baseline truth data is in contrast to previous experiments involving optically-coherent detection data [18,22,24,27,35], which lacked accurate estimates of the true reflectances and phase errors. Our experimental protocol allows us to quantitatively compare image-correction frameworks and validate the performance of CPnP-AR.

In what follows, we provide a brief overview of MBIR and CPnP in Section 2. We then describe our novel image-correction framework CPnP-AR. Hereafter, we refer to the original CPnP algorithm as CPnP-Denoising (CPnP-DN) to help distinguish it from CPnP-AR. In Section 3, we describe our custom-built experimental testbed, which uses DH data. We provide our results in Section 4, where we leverage this testbed and our methodology for obtaining the baseline truth data needed to quantitatively evaluate the performance of CPnP-AR relative to existing image-correction frameworks. We do so for a wide range of objects with varying contrast levels. Our conclusion then follows in Section 5.

2. Image-correction framework

In this section, we briefly describe our physics-based measurement model and our general approach to jointly estimating the reflectance and the phase errors. We then provide an overview of CPnP-DN found in Pellizzari et al. [23], and we present our novel framework, CPnP-AR. Please see Appendices A–D for more details on the physics-based measurement model, CPNP-DN, CNN architectures and training, and algorithmic parameters, respectively.

2.1. Physics-based measurement model

Sensors that employ optically-coherent detection measure the complex-valued field of a scatered signal. In general, we model this measurement, $y \in \mathbb{C}^M$, as

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

Here, $g \in \mathbb{C}^M$ is the rasterized unknown reflection coefficient for the illuminated object, $w \in \mathbb{C}^M$ is complex-valued Gaussian measurement noise with variance σ_w^2, and $A_\phi \in \mathbb{C}^{M\times M}$ is a linear transform describing propagation, measurement geometry, and phase errors, $\phi \in \mathbb{R}^M$ [20,22,24]. While Eq. (1) represents a generic model for optically-coherent detection, the structure of A_ϕ will depend on the sensor modality at hand. In this paper, we consider the model for a sensor employing DH, as described in Appendix A.

Given y, we wish to jointly estimate the speckle-free image, $r = E[|g|^2]$, and the phase errors, ϕ, from single-shot data. Fig. 1 shows an example reflectance image along with averages of multishot data to estimate $r = E[|g|^2]$. Since our goal is to use single-shot data, the $N = 1$ case in Fig. 1 indicates the need for a prior distribution, so we seek the MAP estimate of r and ϕ, given y. With the physically-plausible assumption that r and ϕ are independent, we seek to solve the problem

$$\begin{aligned}(\hat{r}, \hat{\phi}) &= \arg\max_{r,\phi} p(y, r, \phi) \\ &= \arg\max_{r,\phi} p(y|r, \phi)\ p(r)\ p(\phi).\end{aligned} \tag{2}$$

However, since r is the expected value of $|g|^2$, it is non-linearly related to y, making this MAP estimate difficult to obtain [24].

Instead of solving (2) directly, our work in Pellizzari et al. [20,22–24] uses the expectation maximization (EM) algorithm to iteratively construct and minimize a surrogate function, Q, in place of the MAP cost function. With further details described in Appendix A, we obtain a surrogate function for $-\log p(y, r, \phi)$, given by

$$\begin{aligned}Q(r, \phi; r', \phi') &= E_g\left[-\log p(y, g, r, \phi) \mid y, r', \phi'\right] \\ &= Q_1(\phi; r', \phi') + Q_2(r; r', \phi') + Q_3(r) + Q_4(\phi)\end{aligned} \tag{3}$$

where each of the four terms are given as in Pellizzari et al. [23] by

$$Q_1(\phi; r', \phi') = -E_g\left[\log p(y|g, \phi) \mid y, r', \phi'\right] \tag{4}$$

$$Q_2(r; r', \phi') = -E_g\left[\log p(g|r) \mid y, r', \phi'\right] \tag{5}$$

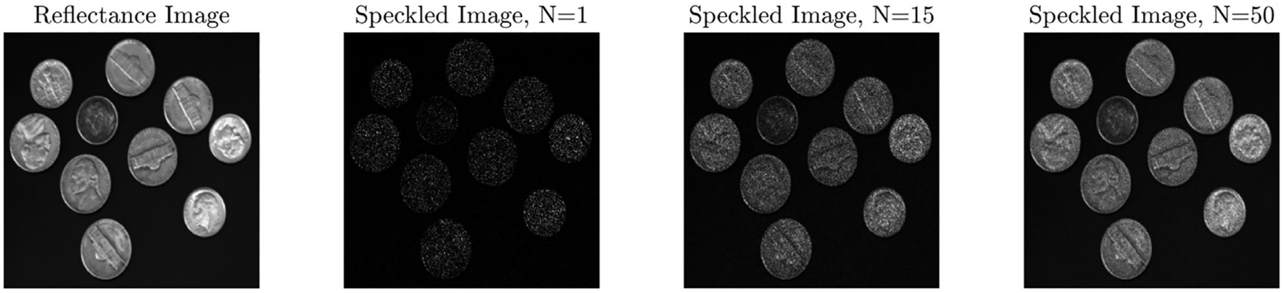


Fig. 1. Example reflectance image, r, and corresponding speckle averaged images, produced according to $\frac{1}{N}\sum_{i=1}^{N}|g_i|$, where i is the index for each independent speckle realization. Here, we show the images produced by averaging $N = 1$, 15, and 50 independent speckle realizations. As N grows large, the average converges to r and is analogous to what we observe in incoherent images.

$$Q_3(r) = -\log p(r) \tag{6}$$

$$Q_4(\phi) = -\log p(\phi). \tag{7}$$

Here, the conditional expectation $E_g[\cdot \mid y, r'\phi']$ indicates that E_g is computed using the posterior distribution of g, as given in Appendix A, with the current estimates, (r', ϕ'), in place of (r, ϕ), while the density functions $p(y|r,\phi)$, $p(y,g|r,\phi)$, and $p(g|y,r,\phi)$ do not depend on r' and ϕ'.

DH-MBIR, CPnP-DN, and CPnP-AR, are all designed using iterations of the form

$$(r', \phi') \leftarrow \arg\min_{r,\phi} \; Q(r,\phi; r', \phi'). \tag{8}$$

However, they solve the problem differently and they each use a different model for $p(r)$. These key differences have a significant impact on the resulting image quality and algorithm robustness. We direct the reader to Appendix A for more information about the physics-based measurement model and this surrogate function.

2.2. CPnP-DN

CPnP-DN is an adaptation of the Multi-Agent Consensus Equilibrium (MACE) framework [6], which uses a set of balance equations to integrate multiple heterogeneous agents [4,11]. Agents are functions, $F : \mathbb{R}^M \to \mathbb{R}^M$, that take an initial estimate as an input and produce a “better” estimate as an output. Here, better means that the output is more consistent with one or more models assigned to that agent [11]. Each agent, F_i, maintains its own version of the estimate, $x_i = (r_i, \phi_i)$, called a state vector. MACE seeks solutions x_i^* so that all the output points $F_i(x_i^*)$ are identical (consensus) and the update steps $F_i(x_i^*) - x_i^*$ sum to 0 (equilibrium).

CPnP-DN and MACE use the idea of proximal maps to convert the cost function formulation of Eq. (3) into an agent update formulation. As described in Sreehari et al. [32], Venkatakrishnan et al. [37], PnP starts with the ADMM algorithm for minimizing a sum of functions, as in Eq. (3). In ADMM, each term $f(x)$ in this sum is converted to a proximal map, given by

$$F(x_{in}) = \underset{x}{\operatorname{argmin}}\left\{\frac{1}{2\sigma^2}||x - x_{in}||^2 + f(x)\right\}. \tag{9}$$

Equation (9) is an agent that maps the input, x_{in}, closer to the minimizer of $f(x)$. The constant σ^2 is a kind of step size, maintaining proximity to this input. As σ^2 increases, the proximal map output gets closer to the minimizer of f, thus making this agent more insistent on its preference.

The key insight of PnP methods is that we can replace these proximal maps with more general maps that are not based on optimization. In particular, the proximal map for the prior model can be interpreted as a MAP estimate for an AWGN forward model [11,32,37]. Thus, we may replace this proximal map with a state-of-the-art CNN denoiser designed to remove AWGN.

Using a CNN denoiser as an agent provides two primary benefits. First, there is no optimization step; instead, we simply apply the denoiser. Second, we inherit the implicit image prior model, $p(r)$, learned by the CNN, which better captures subtle characteristics that explicit analytical models lack.

For CPnP-DN, we define three agents that reduce the cost associated with one or more of the four model terms in the surrogate function in Eq. (3). Fig. 2 shows DH-MBIR, in which the surrogate functions are iteratively minimized in turn, as well as CPnP-DN and CPnP-AR. DH-MBIR can be reformulated using PnP as in Fig. 2(b), in which case agent F_1 updates the phase-error estimate by minimizing $Q_1 + Q_4$ with respect to ϕ, while agents F_2 and F_3 are proximal maps for Q_2 and Q_3, respectively. The consensus solution, r^*, ϕ^*, is found using Mann iterations to a specified convergence tolerance.

In CPnP-DN [23], the agent F_3 is replaced with a CNN designed to remove AWGN, $\mathcal{DN}(\cdot)$, as shown in Fig. 2(c). In this paper, we use the U-in-U Network (UinUNet) architecture [1], trained on a single normalized noise level, $\sigma = 0.1$, and use an L2-loss function. In Appendix B and Appendix C, we provide additional details about CPnP-DN and UinUNet.

2.3. CPnP-AR

As noted above, the images encountered at various iterations of the PnP reconstruction process generally do not follow the AWGN assumption precisely. In the current setting, the presence of speckle in our images tempts us to include a prior agent in the form of a CNN despeckler, which can be trained for state-of-the-art performance for removing speckle [13,26,29]. While such CNNs encode a great deal of image prior information, using a despeckler naively as an agent also fails to match the distribution of images encountered during CPnP reconstruction.

To overcome this mismatch, we introduce a new agent for artifact removal (AR) in the form of a two-stage CNN despeckler, as shown in Fig. 2(d). The first stage takes an input state vector, (r_3, ϕ_3), which represents a candidate reconstruction in the iteration process, and projects it to a speckled image that is consistent with measured data and estimated phase error. In the second stage, a CNN despeckler maps this speckled image into the set of clean images that are consistent with the learned prior model.

More precisely, we define our AR agent as

$$F_3(r_3) = \mathcal{DS}(E_g[\,|g|^2 \mid y,\ \phi_3, r_3]). \tag{10}$$

This agent consists of two steps: computing the expected value of the speckled image given the input state vector and measured data y, then despeckling that image with a CNN $\mathcal{DS}(\cdot)$ trained to remove speckle from a generic set of images. The expected value $E_g[\,|g|^2 \mid y,\ \phi_3, r_3]$ is shown in Appendix A to be the backprojection of the data y modulated by the influence of ϕ_3 and r_3, hence it is a speckled image. We evaluate the expectation in Eq. (10) with the same approach used to evaluate our surrogate function, except that we condition the expectation on the state vector, (r_3, ϕ_3), rather than the current estimate, (r', ϕ'). With this new

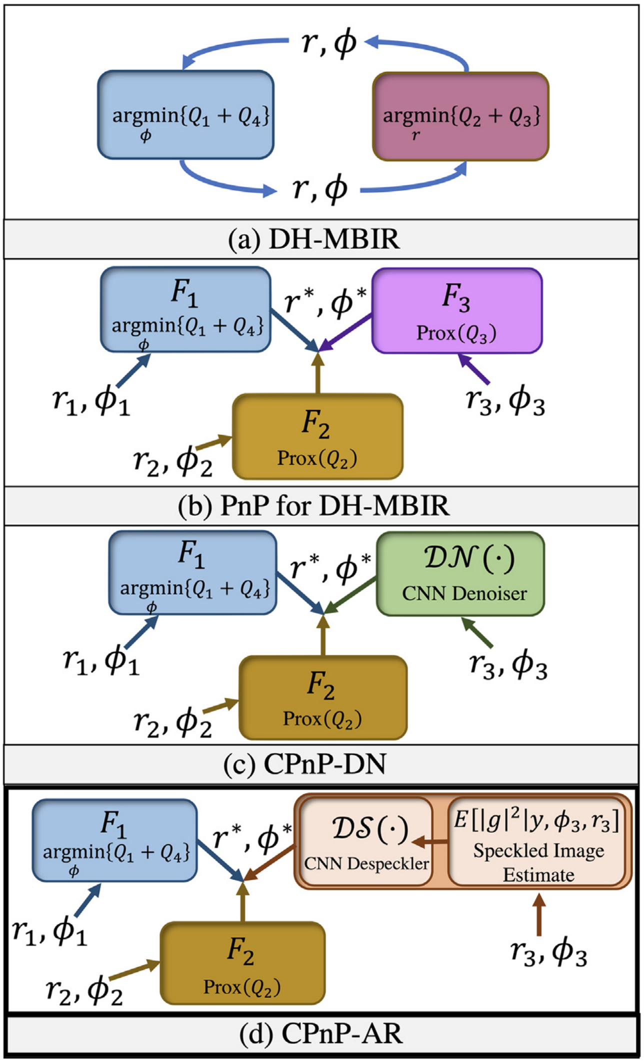


Fig. 2. (a) DH-MBIR uses an iterative coordinate descent minimization to update r and ϕ. (b) Reformulating DH-MBIR using PnP converts Q_2 and Q_3 to proximal maps, while $Q_1 + Q_4$ is minimized separately. Each agent has its own state vector, (r_i, ϕ_i). (c) CPnP-DN uses a CNN trained to remove AWGN in place of prior term proximal map F_3. (d) CPnP-AR replaces the CNN denoising agent with a two-step operation that first transforms the state vector, r_3, ϕ_3 into a speckled image and then applies a CNN trained to remove speckle.

agent, the CPnP-AR algorithm is identical to the CPnP-DN algorithm, except that we use Eq. (10) for F_3 in place of an AWGN CNN denoiser. While this change seems small, it is a fundamentally-different process for incorporating image priors. By using an appropriate resampling method, like the expectation in Eq. (10), we are no longer restricted to using AWGD operators, allowing us to incorporate a wider-range of mapping functions that better model the problem at hand.

In contrast to deep unrolling methods, in which the CNN is trained while embedded in a full reconstruction algorithm, our AR agent decouples the learned image model from the forward model through our two-step process. During the resampling step, we directly use the forward model to evaluate $E_g[|g|^2\ |y, \phi', r_3]$. This step provides us with an estimate of a speckled image based on the current state vector. During the second step, we input this speckled image into our CNN despeckler, $\mathcal{DS}(\cdot)$. In our approach, $\mathcal{DS}(\cdot)$ does not use information about the forward model during training. Instead, we train the CNN using a generic set of speckled images, $|g_i|^2$, for $i \in 1, \ldots, K$, where K is the number of training images. We generate each training pair by drawing from the distribution

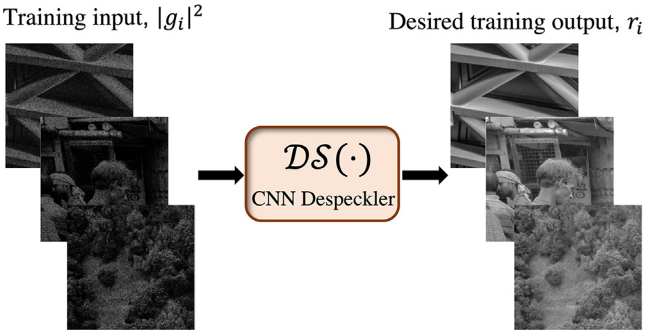


Fig. 3. Example data pairs used for training our CNN despeckler. On the left, we show the training input, a random draw of a speckled image, $|g_i|^2$. On the right, we show the corresponding reflectance image, r_i, used to generate the speckled input. We trained our CNN to map $|g_i|^2 \rightarrow r_i$.

$$p(g) \sim CN(0, D(r), 0) \tag{11}$$

where $D(\cdot)$ denotes an operator that produces a diagonal matrix from its vector argument and $CN(\mu, C, \Gamma)$ indicates a multivariate complex normal distribution with mean, μ, covariance matrix, C, and pseudo-covariance matrix, Γ [8]. We then train the CNN to map $|g_i|^2 \rightarrow r_i$, as shown in Fig. 3.

This generic model is applicable to any coherent imaging application with rough-surface scattering (e.g., LIDAR, ultrasound, active imaging, etc.) [8]. Thus, if the forward model changes, we do not need to retrain the CNN. Instead, we need only use this new forward model when computing the conditional expectation and then apply the original CNN. For our CNN despeckler, we use the same architecture, loss function, and training images that we used for our AWGN denoiser. For further details, see Appendices C and D.

3. Experimental testbed

In this section, we describe our custom-built experimental testbed used for collecting DH data, which is the sensor modality we chose to validate the performance of CPnP-AR. We also describe our methodology for obtaining the baseline truth data needed to rigorously evaluate performance using quantitative metrics. Please see Appendix E for more details about out testbed and methodology.

3.1. Optical setup

Fig. 4 shows a diagram of our custom-built experimental testbed. As shown, we used a 300 mW, 532 nm continuous-wave laser (Oxxius LBX-532S-300) as the master oscillator (MO) with an optical isolator to protect the laser from back reflections. Additionally, we used a half-wave plate and a polarized beam splitter (PBS) to create the signal and reference, and we included a variable neutral density (VND) filter in each leg of the optical setup to control the power. For ease of use, we also used a ten meter polarization-maintaining patch fiber for both legs. We positioned the object being imaged 4.6 m away from the imaging lens and adjusted the divergence of the outgoing beam using the lens at the output of the signal fiber. Our outgoing beam had an approximately Gaussian profile, and we adjusted its divergence until the illumination within the camera field of view was relatively uniform. Since the outgoing beam was not exactly uniform, some images have noticeably darker regions near the corners.

Starting with a one-inch-diameter, 400 mm-focal-length lens (Thorlabs best form), we used the off-axis image plane recording geometry (IPRG) [31]. We positioned the output from the reference fiber next to the imaging lens, and we placed the camera (Point Grey Chameleon

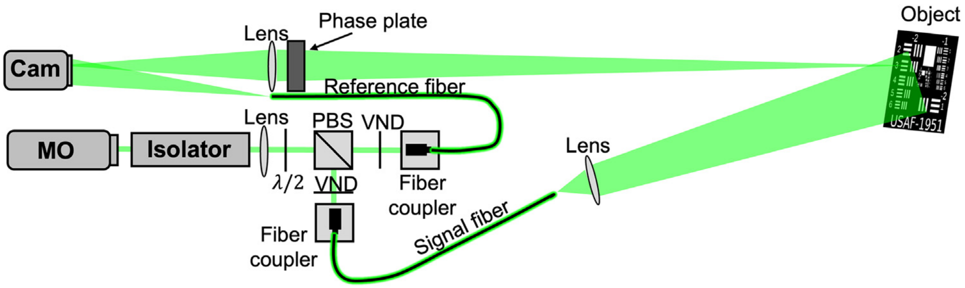


Fig. 4. Diagram of our optical setup depicting the beam path from the MO through the signal and reference legs to the camera.

CMLN-13S2M) in the image plane. We spatially demodulated each 16-bit digital hologram to obtain a 344×344-pixel, complex-valued field in the pupil plane of our sensor. For details about the demodulation process used for the off-axis IPRG, we direct the reader to any of the following resources: [18,22,25,30,31]. To add aberrations to the optical setup, we placed a phase plate as close as possible to the imaging lens. Made by Lexitek [14], this phase plate matched the statistics of realistic atmospheric turbulence, which allowed us to experimentally quantify the strength of the phase errors. Using baseline truth data and the methodology in Ref. [27], we obtained a D/r_0 of 10.7, where D is the aperture diameter and r_0 is the coherence length of our phase plate.

3.2. Baseline truth data

Unique to this paper, we obtained high-quality estimates of the true reflectance, for the objects tested, and the true phase errors, for the phase plate tested. We used this baseline truth data to quantitatively evaluate the performance of CPnP-AR relative to other image-correction frameworks, including Image Sharpening (IS) [34], DH-MBIR, and CPnP-DN. While these frameworks are designed for, and tested with, single-shot data, in our laboratory experiments we have the benefit of using multi-shot approaches to obtain our high-quality images and phase errors to serve as truth data.

To obtain baseline truth data for the reflectance r (for each object tested), we used speckle averaging rather than incoherent illumination in order to better capture the dynamic range of our DH data and to match the estimated reflectance to our illumination wavelength. For these baseline speckle-averaged images, we recorded 20 digital holograms, each with independent speckle realizations, by rotating the object on a precision rotation stage (Thorlabs PR01). We determined that with an object distance of 4.6 m, 2.8 mrad of rotation resulted in independent speckle realizations on a frame-to-frame basis. Note that we averaged the magnitude squared of the complex images obtained from the DH data, not the recorded digital holograms. More precisely, we first demodulated each of the 20 holograms, h_i for $i \in [1, 20]$, to obtain the complex measurement, y_i, given by Eq. (1). Next, we formed our 20 images according to $|A^H y_i|^2$, where the superscript H represents the Hermitian transpose. Note that the phase errors, ϕ, are 0 in this case. Lastly, we averaged these 20 images to obtain our baseline reflectance function.

To avoid blurring caused by the frame-to-frame rotation, we registered the 20 real-valued images prior to averaging using MATLAB's *imregister* function with default monomodal settings. We fixed the first image and registered the remaining 19 images to the fixed image. Lastly, to reduce the residual speckle variation arising from a sample size of 20 independent speckle realizations, we applied a Gaussian spatial filter, with a width $\sigma = 1.5$ pixels, to each averaged image.

To obtain baseline truth data for the phase errors (for each position of the phase plate tested), we explored two methods: (1) using a point source in the object plane and (2) applying DH-MBIR to multi-shot data with a high-contrast object. We found that the second approach worked better due to the dynamic-range limitations of the off-axis IPRG. Therefore, we obtained the baseline truth data by following the methodology from Horst et al. [10]. We used a transmissive chrome-on-glass USAF 1951 bar chart backed by a sheet of white paper, and we obtained five high signal-to-noise ratio (SNR) measurements, each with independent speckle realizations. We then used DH-MBIR with multi-shot data, as in Bate et al. [3], to produce high-quality estimates of the phase errors.

3.3. Quantitative metrics

To quantitatively evaluate image quality, we used MATLAB's structural similarity index measure (SSIM) and peak SNR (PSNR) functions with default settings [16,17]. Before using these functions, we first normalized our baseline truth reflectance to the range $[0, 1]$. Next, using Eq. (16) in Pellizzari et al. [22], we removed any multiplicative scalar offset between each reconstruction and our normalized baseline truth reflectance.

To assess the quality of phase errors, we used peak Strehl ratio, S_p. We define S_p as

$$S_p = \frac{\{\mathrm{PSF_c}\}_{\max}}{\{\mathrm{PSF_d}\}_{\max}}, \tag{12}$$

where $\{\cdot\}_{\max}$ indicates that we take the maximum value of the argument, $\mathrm{PSF_c}$ is the point-spread function (PSF) of the imaging system after correction of phase errors, and $\mathrm{PSF_d}$ is the diffraction-limited PSF [23]. Thus, the peak Strehl ratio is a normalized measure of how close imaging-system performance is to the diffraction limit, neglecting the effects of tilt.

4. Results and discussion

Figs. 5 and 6 show results for the 16 objects tested. The far left column shows the baseline truth data, r_{truth}. In the second column, we show the back projections, $|A_0^H y|^2$. These back projections represent the raw images with no correction (i.e., with $\phi = 0$). They contain speckle, measurement noise, and are corrupted by phase errors. In the remaining four columns, we show the results for IS, DH-MBIR, CPnP-DN, and CPnP-AR, respectively. To improve visibility for plotting purposes, we reduced the dynamic range by taking the square root of each image.

The results in Figs. 5 and 6 show the difficulty of regularizing coherent images. With no regularization, IS produces a speckled image. Using a simple model for $p(r)$, DH-MBIR does reduce the speckle variations; however, it does so only in certain regions that contain high signal levels. It also leaves "hole-like" artifacts throughout the image. We found that further increasing the regularization with DH-MBIR over-blurred the image.

With an AWGN denoiser, CPnP-DN does obtain better regularization and perceptual image quality than DH-MBIR. However, our results highlight a significant challenge with CPnP-DN that was not observed in our earlier simulated work—the optimal amount of regularization varies based on the object. Here, we see that by fixing our hyperparameters, we get over-regularization in some images and under-regularization in others. As an example, Object 5 appears over-regularized while Object 7 is under-regularized. While we could obtain better performance by adjusting the regularization for each image, such a method would be

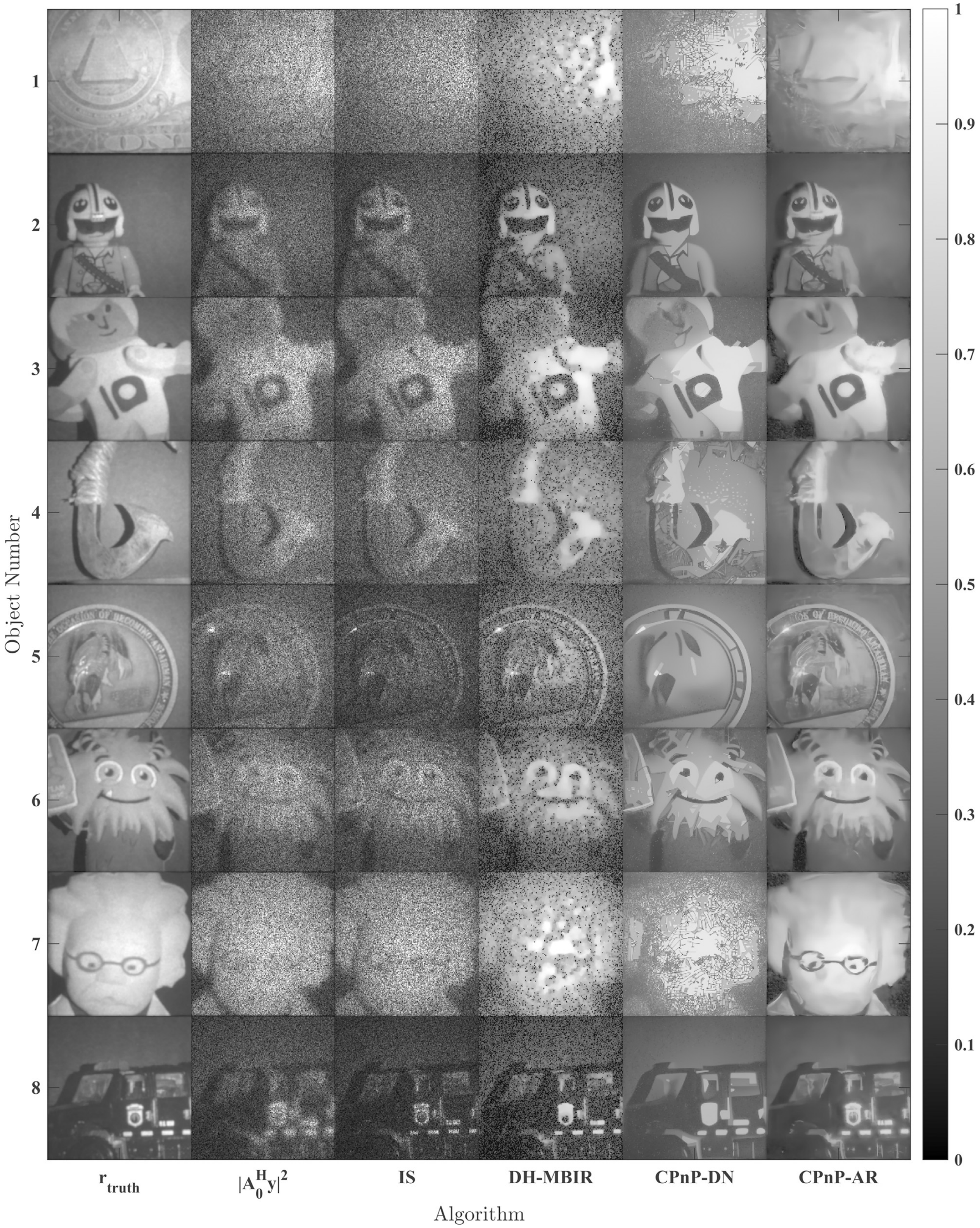


Fig. 5. Results for objects 1–8 shown on a normalized and unitless scale. To improve the visibility of these high-dynamic range images, we show the square root of the reconstructions here. The far-left column shows the baseline truth data, r_{truth}, whereas the second column shows the blurry and noisy back projections, $|A_0^H y|^2$. The remaining columns show the results from IS, DH-MBIR, CPnP-DN, and CPnP-AR, respectively.

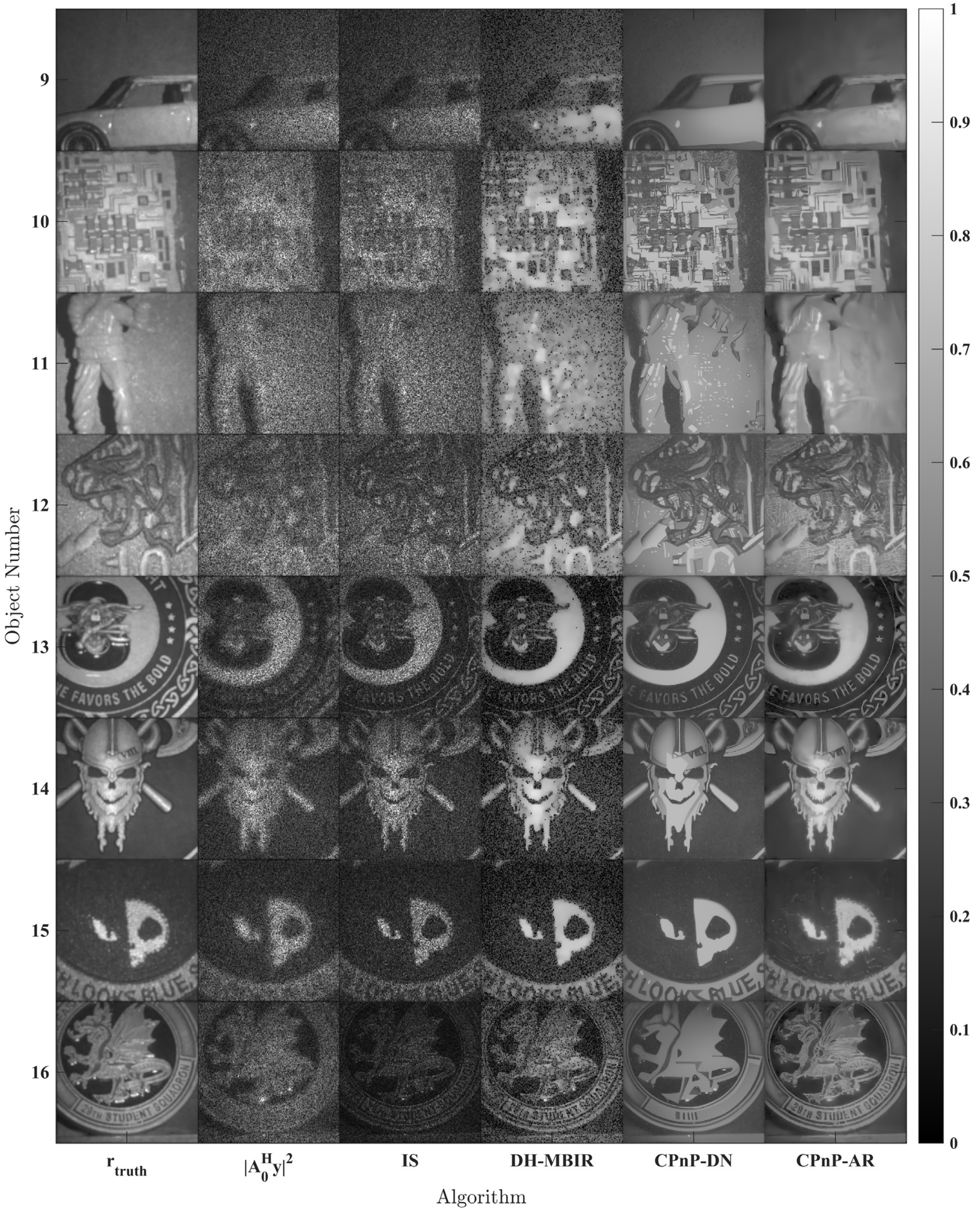


Fig. 6. Results for objects 9–16 shown on a normalized and unitless scale. To improve the visibility of these high-dynamic range images, we show the square root of the reconstructions here. The far-left column shows the baseline truth data, r_{truth}, whereas the second column shows the blurry and noisy back projections, $|A_0^H y|^2$. The remaining columns show the results from IS, DH-MBIR, CPnP-DN, and CPnP-AR, respectively.

Table 1
Quantitative performance measures for each algorithm on all 16 objects.

	Peak Strehl Ratio				SSIM				PSNR			
Object #	IS	DH-MBIR	CPnP-DN	CPnP-AR	IS	DH-MBIR	CPnP-DN	CPnP-AR	IS	DH-MBIR	CPnP-DN	CPnP-AR
1	0.02	0.01	0.02	**0.04**	0.16	0.32	0.37	**0.67**	12.6	17.9	17.9	**23.5**
2	0.04	0.47	0.65	**0.66**	0.65	0.80	0.87	**0.90**	24.6	30.1	28.9	**31.1**
3	0.09	0.06	0.51	**0.67**	0.37	0.52	0.76	**0.84**	14.6	22.2	23.6	**25.7**
4	0.07	0.04	**0.33**	0.29	0.33	0.59	0.72	**0.83**	19.1	25.3	25.8	**27.8**
5	0.30	0.73	**0.83**	0.80	0.24	0.45	**0.77**	0.74	19.4	22.9	**25.7**	**25.7**
6	0.08	0.19	0.67	**0.71**	0.45	0.64	0.81	**0.91**	19.1	25.5	27.0	**31.7**
7	0.03	0.01	0.02	**0.12**	0.16	0.26	0.19	**0.75**	11.5	16.3	15.5	**23.0**
8	0.40	0.75	0.74	**0.77**	0.76	0.82	0.79	**0.85**	23.4	26.7	25.6	**27.2**
9	0.07	0.06	0.12	**0.16**	0.68	0.80	0.91	**0.92**	24.2	29.6	33.5	**33.9**
10	0.23	0.74	0.85	**0.86**	0.18	0.33	0.33	**0.37**	15.3	**18.9**	18.2	18.8
11	0.07	0.16	0.50	**0.66**	0.21	0.50	0.64	**0.76**	17.4	23.0	23.6	**25.1**
12	0.43	0.63	0.66	**0.67**	0.27	0.50	**0.63**	0.59	18.3	24.3	24.5	**25.0**
13	0.71	0.89	**0.92**	**0.92**	0.36	0.51	**0.61**	0.58	18.5	20.9	21.6	**21.6**
14	0.48	0.84	**0.90**	**0.90**	0.59	0.76	0.84	**0.91**	19.6	27.5	25.5	**30.1**
15	0.79	0.88	0.89	**0.90**	0.65	0.76	0.83	**0.85**	20.9	27.0	26.4	**27.9**
16	0.77	0.85	**0.90**	**0.90**	0.29	0.51	**0.69**	0.62	22.0	24.7	**26.9**	26.3
Mean	0.29	0.46	0.59	**0.63**	0.40	0.57	0.67	**0.76**	18.8	23.9	24.4	**26.5**
STD	**0.27**	0.35	0.31	0.29	0.20	0.18	0.20	**0.15**	**3.7**	3.9	4.3	3.8

practical only if we could automate the process. Furthermore, for many objects, we see that the optimal amount of regularization varies locally within the image. As an example, for Objects 1 and 7, the bright regions within the images are over-regularized, while the dimmer regions are under-regularized.

Using the AR agent, CPnP-AR produces reconstructions with higher perceptual image quality compared to the other algorithms. We observe that CPnP-AR does a better job of regularizing different types of images with a fixed set of hyperparameters. Furthermore, within a single image, it does better than CPnP-DN at regularizing both the bright and dim regions.

In Table 1, we present our quantitative results for peak Strehl Ratio and our image-quality metrics. These results reinforce the qualitative results shown in Figs. 5 and 6. On average, the two CPnP algorithms produce similar peak Strehl ratios, with both performing significantly better than DH-MBIR and IS. CPnP-AR provides a slight advantage over CPnP-DN in this category. Also, both algorithms obtain a significant improvement in image quality compared to DH-MBIR and IS. In this category, CPnP-AR provides a significant advantage over CPnP-DN.

5. Conclusion

In this paper, we developed a new image-correction framework: CPnP-AR, which incorporates the benefits of a neural-network despeckler in a way that maintains the independence of prior-model training and forward-model implementation. We also developed a custom-built experimental testbed and the associated methodology needed to quantitatively evaluate performance. The results show that CPnP-AR significantly outperforms competing methods: IS, DH-MBIR, and CPnP-DN, most notably without the need for additional parameter tuning. Overall, CPnP-AR produced higher-quality images and moreaccurate estimates of the phase errors for a wide range of objects. This improvement in overall robustness is a key step towards employing CPnP-AR for numerous applications of interest. While CPnP-AR performed significantly better than the other methods for low-contrast images, there is still room for improvement. One possible approach to better reconstruct low-contrast objects might be to train the image model using a wider-range of image contrasts, or to integrate multiple image models trained on different contrast levels.

Data and source code availability

The data and source code underlying the results may be obtained from the corresponding author upon reasonable request.

Declaration of Competing Interest

Authors declare that they have no conflict of interest.

CRediT authorship contribution statement

Casey J. Pellizzari: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing – original draft, Writing – review & editing. **Timothy J. Bate:** Data curation, Formal analysis, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **Kevin P. Donnelly:** Data curation, Formal analysis, Investigation. **Gregery T. Buzzard:** Conceptualization, Methodology, Validation, Writing – review & editing. **Charles A. Bouman:** Conceptualization, Funding acquisition, Methodology, Validation, Project administration, Resources, Writing – original draft. **Mark F. Spencer:** Conceptualization, Funding acquisition, Investigation, Methodology, Project administration, Resources, Supervision, Validation, Writing – original draft, Writing – review & editing.

Data availability

Data will be made available on request.

Acknowledgments

We would like to thank the Joint Directed Energy Transition Office for sponsoring this research. This work was also supported by NSF under Grant CCF-1763896. The views expressed in this article are those of the authors and do not necessarily reflect the official policy or position of the United States Air Force Academy, the Air Force, the Department of Defense, or the U.S. Government. This work is approved for public release; distribution is unlimited. Public Affairs release approval # USAFA-DF-2022-780. Additionally, we thank Mr. John R. Coe for his support of our laboratory experiments.

Appendix A. Problem Formulation

In this Appendix, we expand our description of the problem at hand starting with our data model and surrogate function. We also specify the physics-based forward model, A_ϕ, and provide information about the distribution of our data. Thereafter, we describe the posterior distribution of g, given r', ϕ' used in our evaluation of the EM algorithm.

A1. Data model and surrogate function

As described in the main body of the paper, we model the complex-valued measurement, $y \in \mathbb{C}^M$, for a sensor employing optically-coherent detection as

$$y = A_\phi g + w, \tag{13}$$

where $g \in \mathbb{C}^M$ is the rasterized unknown reflection coefficient for the illuminated object, $w \in \mathbb{C}^M$ is complex-valued measurement noise with variance σ_w^2, and $A_\phi \in \mathbb{C}^{M\times M}$ is a linear transform describing propagation, measurement geometry, and phase errors, $\phi \in \mathbb{R}^M$ [20,22].

Given y, we wish to jointly estimate the speckle-free image, $r = E[|g|^2]$, and the phase errors, ϕ, from single-shot DH data. With the physically plausible assumption that r and ϕ are independent, we seek to solve the problem

$$(\hat{r}, \hat{\phi}) = \arg\max_{r,\phi}\ p(y \mid r, \phi)\ p(r)\ p(\phi). \tag{14}$$

Instead of solving (14) directly, we use the expectation maximization (EM) algorithm as in Refs. [20,22–24] to iteratively construct and minimize a surrogate function, Q in place of the MAP cost function. Those papers show that the assumptions on y, r, g yield the following conditional distributions

$$y \mid g, \phi \sim CN(A_\phi g, \sigma_w^2 I, 0) \tag{15}$$

$$g \mid r \sim CN(0, \mathcal{D}(r), 0) \tag{16}$$

$$g \mid y, r, \phi \sim CN(\mu_{r,\phi}, C_{r,\phi}, 0), \tag{17}$$

where $A_\phi, \mu_{r,\phi}, C_{r,\phi}$ are defined below. These distributions determine the corresponding density functions $p(\cdot \mid \cdot)$ used below. To relate these distributions to $p(y \mid r, \phi)$ as in (14), we note that the EM algorithm as in Ref. [4] uses the fact that

$$\begin{aligned} -\log p(y \mid r, \phi) = {} & E_g\left[-\log p(y, g \mid r, \phi) \mid y, r^{'}, \phi^{'}\right] \\ & - E_g\left[-\log p(g \mid y, r, \phi) \mid y, r^{'}, \phi^{'}\right]. \end{aligned} \tag{18}$$

Here the conditional expectation $E_g[\cdot \mid y, r'\phi']$ indicates that the distribution of g for the expectation E_g is given by (17) with (r', ϕ') in place of (r, ϕ), while the density functions $p(y|r,\phi)$, $p(y, g|r, \phi)$, and $p(g|y, r, \phi)$ do not depend on r' and ϕ'. The second term in (18) can be shown to achieve its maximum when $(r, \phi) = (r', \phi')$, so that the first term serves as a surrogate function for $-\log p(y \mid r, \phi)$. That is, instead of minimizing (18) directly, we can fix (r', ϕ'), minimize the first term in (18) over (r, ϕ), use these to update (r', ϕ'), and iterate.

After including the prior terms, we obtain a surrogate function for $-\log p(y, r, \phi)$ given by

$$\begin{aligned} Q(r, \phi; r', \phi') &= E_g\left[-\log p(y, g, r, \phi) \mid y, r', \phi'\right] \\ &= Q_1(\phi; r', \phi') + Q_2(r; r', \phi') + Q_3(r) + Q_4(\phi) \end{aligned} \tag{19}$$

where each of the four terms are given as in Pellizzari et al. [23] by

$$Q_1(\phi; r', \phi') = -E_g\left[\log p(y|g, \phi) \mid y, r', \phi'\right] \tag{20}$$

$$Q_2(r; r', \phi') = -E_g\left[\log p(g|r) \mid y, r', \phi'\right] \tag{21}$$

$$Q_3(r) = -\log p(r) \tag{22}$$

$$Q_4(\phi) = -\log p(\phi) \tag{23}$$

and where we have used $p(y, g, \mid r, \phi) = p(y \mid g, \phi)\ p(g \mid r)$, and where $p(y \mid g, \phi)$ and $p(g \mid r)$ are determined by (15) and (17).

DH-MBIR, CPnP-DN, and CPnP-AR, are all designed using iterations of the form

$$(r', \phi') \leftarrow \arg\min_{r,\phi}\ Q(r, \phi; r', \phi'). \tag{24}$$

A2. Forward model

While Eqs. (1) and (15) represent a generic model for sensors that employ optically-coherent detection, the structure of A will depend on the application at hand. In this paper, we consider the model for a sensor employing DH. In practice, DH data starts with a master oscillator (MO) and arises from the interference between a strong reference from a local oscillator and the scattered signal from the coherent illumination of a distant object [30]. The associated spatially modulated irradiance patterns or "holograms" are digitized using a camera, and the resulting digital holograms encode information about the amplitude and phase of the scattered signal. This information is decoded by spatially demodulating the digital holograms [25]. In so doing, we gain access to the path-integrated amplitude and phase of the scattered signal, which allows the decoded images to be digitally corrected.

In Appendix A of Ref. [22], we described a generalized form of A_ϕ for DH. In this paper, we restrict to a single Fresnel propagation between the object and the digital-holographic sensor. We also restrict to cases in which the phase errors are concentrated near the pupil plane of our digital-holographic sensor, resulting in a shift-invariant point spread function (PSF) in the image domain. The resulting model for A_ϕ is given by

$$A_\phi = D(a) D(e^{iP\phi}) \Lambda_1 F^{-1} H F \Lambda_0 \tag{25}$$

where $D(\cdot)$ denotes an operator that produces a diagonal matrix from its vector argument, $a \in \mathbb{R}^M$ is a binary vector that represents the transparency of the circular aperture in the pupil plane, $F \in \mathbb{C}^{M\times M}$ is a 2D discrete Fourier transform (DFT) matrix, $H \in \mathbb{C}^{M\times M}$ is the free-space transfer function, and Λ_0, $\Lambda_1 \in \mathbb{C}^{M\times M}$ are diagonal matrices that apply the quadratic-phase factors in each plane for the Fresnel propagation. Furthermore, we use an interpolation matrix, $P \in \mathbb{R}^{M\times L}$, where $L \le M$, that allows us to model the phase errors, $\phi \in \mathbb{R}^L$, on a low-resolution grid. In this paper, we used nearest-neighbor interpolation. For complete details on the structure of these matrices, we direct the reader to Pellizzari et al. [22].

A3. Posterior distribution

As noted above, the conditional posterior distribution $p(g \mid y, r', \phi')$, as in (17), is the distribution used to evaluate the expectation with respect to g as in Eqs. (4), (5), and (10) of the main paper and (20) and (21) of this Appendix. In [20], we showed that this distribution is complex Gaussian with mean

$$\mu\left(r^{'}, \phi^{'}\right) = C\left(r^{'}, \phi^{'}\right) \frac{1}{\sigma_w^2} A_{\phi^{'}}^H y\ , \tag{26}$$

and covariance

$$C(r', \phi') = \left[\frac{1}{\sigma_w^2} A_{\phi'}^H A_{\phi'} + D(r')^{-1}\right]^{-1} \approx D\left(\frac{\sigma_w^2}{1 + \frac{\sigma_w^2}{r'}}\right). \tag{27}$$

In this paper, we indicate explicitly that μ and C are functions of inputs r', ϕ' since we will later use a state variable as input, rather than r', ϕ'. The approximation in Eq. (27) assumes that $A_{\phi'}^H A_{\phi'} \approx I$. In practice, we have found this approximation to work well [18,20,22].

Using this diagonal approximation, Eqs. (26) and (27) imply that $|g_i|^2$ has a non-central χ^2 distribution with

$$E_g[\ |g_i|^2 \mid y, r', \phi'] = C_{i,i}(r', \phi') + |\mu_i(r', \phi')|^2 \ \text{ for all } i \tag{28}$$

where μ_i is the ith element of the posterior mean and $C_{i,i}$ is the ith diagonal element of the posterior covariance.

Appendix B. CPnP-DN

In this Appendix, we give further details on CPnP-DN, particularly as it relates to the Multi-Agent Consensus Equilibrium (MACE) framework [6]. This framework allows us to integrate three heterogeneous agents, each of which is associated with one or more of the four model terms in the surrogate function Q in Eq. (3). We use agents

$$
\begin{aligned}
F_1(r', \phi') &= \underset{\phi}{\operatorname{argmin}} \left\{ Q_1(\phi; r', \phi') + Q_4(\phi) \right\} \\
F_2(r_2; r', \phi') &= \underset{r}{\operatorname{argmin}} \left\{ \frac{1}{2\sigma_2^2} \| r - r_2 \|^2 + Q_2(r; r', \phi') \right\} \\
F_3(r_3) &= \mathcal{DN}(r_3).
\end{aligned}
\tag{29}
$$

Agent F_1 updates the phase-error estimate by minimizing Q_1 and Q_4 with respect to ϕ. The output of F_1 is the MAP estimate of ϕ, given the data and our current estimate, r', ϕ'. Given the relatively simple structure of ϕ, we use a Gaussian Markov Random Field (GMRF) for the prior model, $p(\phi)$ used to define $Q_4(\phi)$ [23]. To conduct the optimization required for F_1, we use a quasi-Newton gradient descent method.

Unlike the update of ϕ, we split the forward- and prior-model terms for r between two agents, F_2 and F_3. As shown, F_2 is a proximal map for the data-fitting image model Q_2, but this proximal map has a relatively simple closed form solution, equivalent to rooting a 3rd order polynomial for each pixel [23]. Finally, for the image prior model, we use a CNN designed to remove AWGN with variance σ_3^2 [23]. In particular, we use the U-in-U network (UinUNet) architecture, described in Appendix C for $\mathcal{DN}(\cdot)$ [1].

We combine the agents in Eq. (29) by first defining $v = [\phi; r_2; r_3]$ to be a vector formed by stacking three state variables and defining

$$
\mathbf{F}(v; r', \phi') = \begin{bmatrix} F_1(r', \phi') \\ F_2(r_2; r', \phi') \\ F_3(r_3) \end{bmatrix}, \qquad \mathbf{G}(v) = \begin{bmatrix} \phi \\ \bar{r} \\ \bar{r} \end{bmatrix}.
\tag{30}
$$

Here, $\mathbf{F}$ is an operator formed by the application of the three agents in parallel, $\mathbf{G}$ is an operator that averages and stacks the reflectance state vectors, $\bar{r} = (r_2 + r_3)/2$, and r', ϕ' on one iteration is obtained by using the values of r_2, r_3, and ϕ from the previous iteration and taking $r', \phi' = [\bar{r}, \phi]$.

The MACE balance equation is then given by

$$
\mathbf{F}(v^*; r^*, \phi^*) = \mathbf{G}(v^*) \ ,
\tag{31}
$$

where $v^* = (\phi^*, r_2^*, r_3^*)$, and the final reconstruction is given by $r^* = (r_2^* + r_3^*)/2$, which represents an equilibrium and a consensus among the 3 agents [5].

To find this solution, we use the fact that v^* is a fixed point of the operator, $\mathbf{T} = (2\mathbf{G} - \mathbf{I})(2\mathbf{F} - \mathbf{I})$, where $\mathbf{I}$ is the identity map. We find this fixed point using Mann iterations of the form

$$
\begin{aligned}
z &\leftarrow \mathbf{F}(v) \\
v &\leftarrow v + 2\rho[\mathbf{G}(2z - v) - z]
\end{aligned}
\tag{32}
$$

where $\rho = \rho(\rho_\phi, \rho_r)$ is a linear map that multiplies the first of the three components in the stacked vector, $[\mathbf{G}(2z - v) - z]$, by ρ_ϕ and multiplies the other two components by ρ_r.

In Ref. [23], we showed that when these updates converge to a fixed point, that fixed point is also a solution to the MACE equation defined in Eq. (31). Ref. [23] also shows that when the CNN agent F_3 is replaced by a proximal map for the function Q_3, then any solution to the MACE equation is also a solution to the original MAP problem given by Eq. (2). However, since F_3 is not a proximal map, and since the first two terms in the surrogate function are non-convex, proper initialization can improve convergence and may improve the final solution point.

Following [18,20,22–24], we employ a set of nested iterations to improve performance. Specifically, we use an outer loop of N_1 iterations with periodic restarts using the current estimate of ϕ and simple backprojection to re-estimate r. We use an inner loop of N_2 iterations to find the solution given the initialization from the outer loop. These inner iterations first define the surrogate functions at the current base point, then take a step towards equilibrium using (32). This single step approach is based on the idea of a partial update as in Pellizzari et al. [23], Sridhar et al. [33].

In Fig. 7, we provide pseudo code for the general CPnP algorithm. This pseudo code applies to both CPnP-DN and CPnP-AR, which differ only in the behavior of agent F_3.

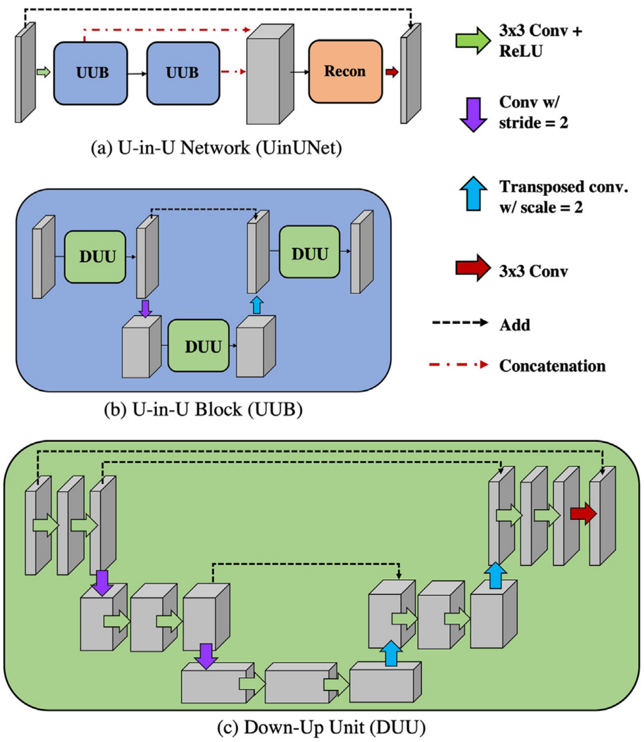
Appendix C. CNN architectures and training

To maintain a fair comparison between CPnP-DN and CPnP-AR, we used the same CNN UinUNet architecture for both frameworks, shown

Algorithm 1 General CPnP Algorithm

```
Input: $N_1$, $N_2$, $\sigma_2^2$, $\sigma_3^2$, $\sigma_w^2$, $y$
$\phi' \leftarrow 0$
for $i = 1 : N_1$ do   # Multi-start re-initialization
    $r' \leftarrow r_2 \leftarrow r_3 \leftarrow |A_{\phi'}^H y|^2$
    $v \leftarrow (\phi, r_2, r_3)$
    for $j = 1 : N_2$ do   # Find solution for this initialization
        # E-Step: Define $Q$ and $F$
        $Q(r, \phi; r', \phi') = E_g[-\log p(y, g, r, \phi) \mid y, r', \phi']$ # as in (20)–(23)
        $F_1, F_2, F_3 \leftarrow$ definition as in (29)
        # M-Step: Take a step towards consensus equilibrium
        $z \leftarrow \mathbf{F}(v)$
        $v \leftarrow v + 2\rho\,[\mathbf{G}(2z - v) - z]$
        # Update surrogate base point
        $(\phi', r') \leftarrow \left(v_1, \frac{1}{2}(v_2 + v_3)\right)$
    end for
end for
Return $r', \phi'$
```

Fig. 7. The general C-PnP algorithm consists of an outer loop for periodic restarts using the current estimate of ϕ and simple backprojection to re-estimate r, and an inner loop to iteratively conduct the joint estimation. Within the inner loop, we use a variant of the EM algorithm to define a surrogate for the cost function, convert this surrogate to agents, then we use Mann iterations to find a solution. Note that CPnP-DN and CPnP-AR are implemented using the same algorithm; they differ only in the form of the image agent F_3.



(a) U-in-U Network (UinUNet)

(b) U-in-U Block (UUB)

(c) Down-Up Unit (DUU)

Fig. 8. (a) Overall U-in-U Network (UinUnet) architecture [1]. In this paper, we used two U-in-U Blocks (UUBs) (b) UUB architecture. Each UUB has a single down-up structure and contains three Down-Up Units (DUU)s (c) DUU architecture. Each DUU resembles a conventional Unet.

in Fig. 8 and described in Abdelhamed et al. [1]. This residual-learning architecture uses a nested UNet structure and yielded state-of-the-art performance in the NTIRE 2020 real-denoising challenge. The UinUNet is built with U-in-U Blocks (UUBs), which use a “down-up” architecture along with several three-level “down-up units” (DUUs) that resemble conventional Unets. The output of the UUBs are concatenated and fed into a final ”Recon” block that applies a series of eight consecutive 3 × 3 convolutions and ReLU blocks. We used two U-in-U Blocks (UUBs) for our network.

To produce CNN agents for CPnP-DN and CPnP-AR, we trained two UinUnets with corresponding training pairs, (y, x), where y is the noisy input and x is the clean image used to evaluate the $L2$ loss function. For both CNNs, we obtained x from the 400 training images found at [2] and normalized all values to be $\in [0,\ 1]$.

For CPnP-DN, we generated noisy data $y_n \in \mathbb{R}^M$ from clean images x_n according to

$$y_n = x_n + \sigma w_n \ , \tag{33}$$

where $w_n \in \mathbb{R}^M$ is a random vector with elements that are zero-mean Gaussian with unit variance. In this paper, we used a fixed noise level of $\sigma = 0.1$. For CPnP-AR, we generated noisy data (speckled images) according to

$$y_n = \left| \frac{x_n}{2} \odot w_{n,1} + i \frac{x_n}{2} \odot w_{n,2} \right|^2 \ , \tag{34}$$

where $i = \sqrt{-1}$, $w_{n,1}$, $w_{n,2} \in \mathbb{R}^M$ are two zero-mean Gaussian vectors with unit variance, and $\odot$ indicates element-wise multiplication. Then y_n is a speckle realization using the values of x_n as the reflectance r.

We trained both CNNs in MATLAB using the Deep Learning Toolbox with the ADAM training routine. We trained for 30 epochs with a learning rate of 1×10^{-4} and a batch size of 4 images. We reduced our learning rate by half every 5 epochs.

Appendix D. Algorithmic parameters

In this Appendix, we provide additional details about the application and parameters for each of IS, DH-MBIR, CPnP-DN, and CPnP-AR. For each approach, we estimate the phase errors on a low-resolution, 86 × 86 grid and convert to a 344 × 344 grid using nearest neighbors interpolation. This step enables comparison with our baseline truth data. For IS, we used the steps outlined in Thurman [34] with a sharpness parameter $\beta = 0.5$, and we iteratively reduced the phase-error pixel binning according to $B_p = [86,\ 43,\ 8,\ 4]$.

DH-MBIR, CPnP-DN, and CPnP-AR share several characteristics, including the same phase-error estimation step and common parameters $N_1 = 40$, $N_2 = 250$, $\sigma_\phi = 0.25$, and $\sigma_w = 0.3$ (see Ref. [23] for all parameters not defined in this Appendix). Specifically, for DH-MBIR, we set the unitless regularizing parameter $\gamma = 5$. For CPnP-DN and CPnP-AR, we set $\rho_\phi = 0.5$ and $\rho_r = 0.8$. For CPnP-DN, we also set $\sigma_2 = 0.34$ and $\sigma_3 = 0.1$. Lastly, for CPnP-AR, we set $\sigma_2 = 0.05$. Note that the prior agent for CPnP-AR, given by Eq. (9), does not have a σ_3.

For each algorithm, we used a single set of parameters for all reconstructions. We selected the regularization parameters that produced the highest quality estimates when averaged over all 16 data sets. Using our truth data, we quantified the quality, q, of our estimate as the sum of the peak Strehl ratio and the SSIM, $(q = S_p + \text{SSIM})$. To maximize q with respect to the regularization parameters, we used MAT-

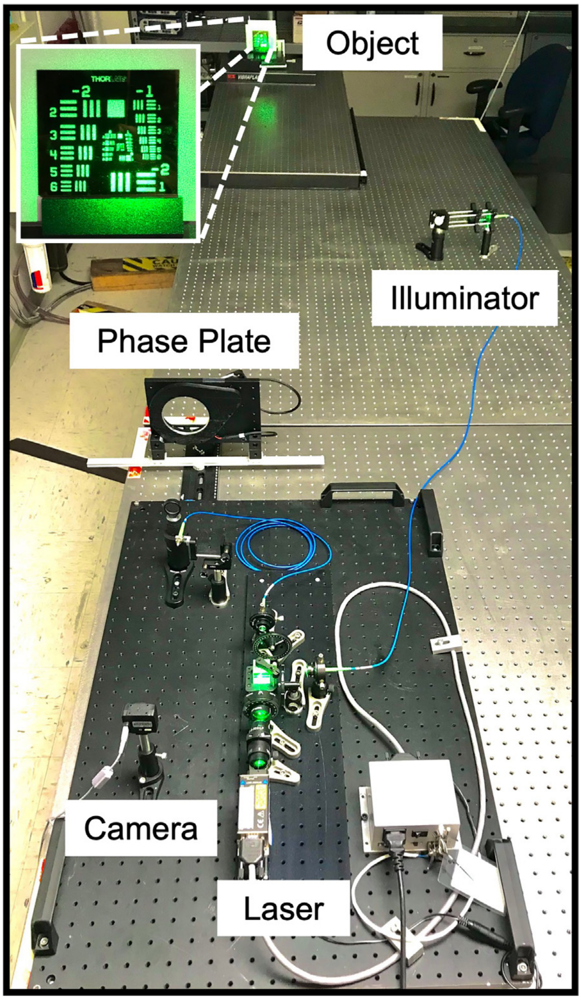


Fig. 9. Photo of our optical setup depicting the beam path from the laser to the camera. Note that we moved the phase plate closer to the imaging lens than what is depicted in the photo.

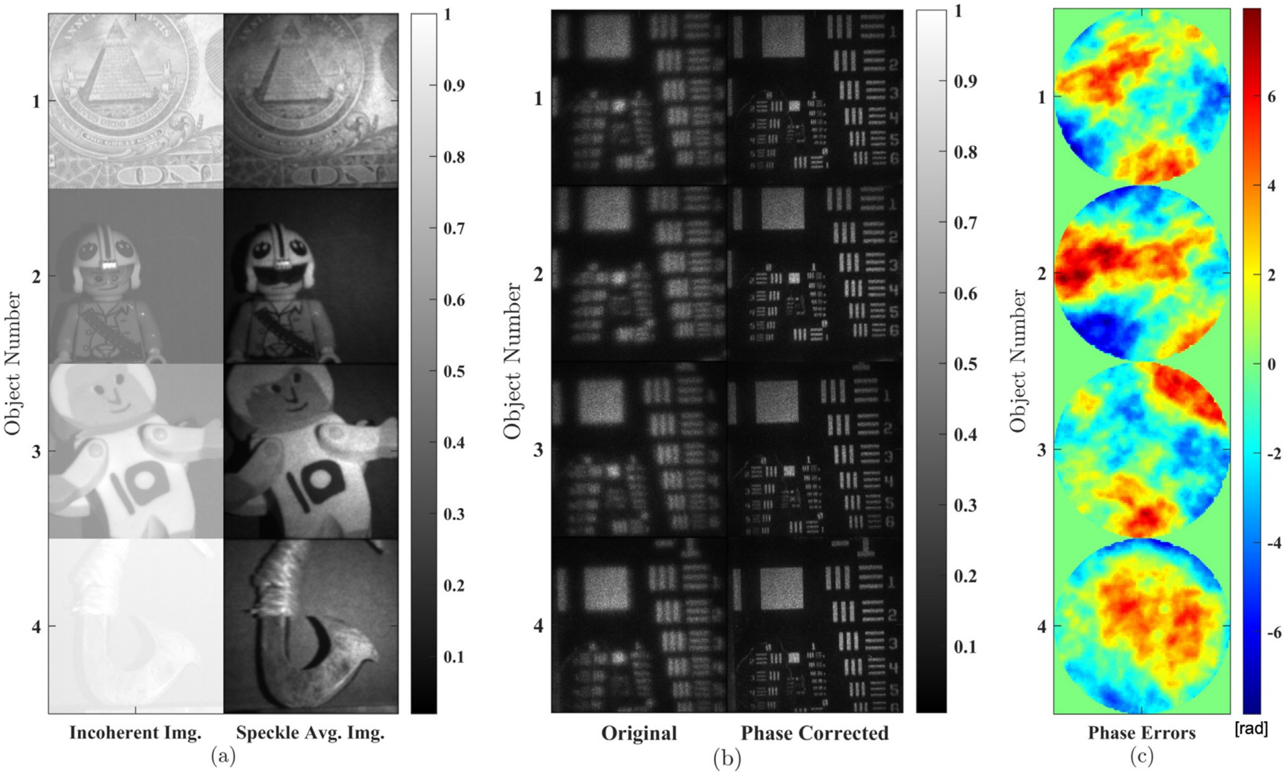


Fig. 10. On the left (a), we show example incoherent images and the corresponding speckle-averaged images using a normalized and unitless scale. Here, we used the speckle-averaged images as our baseline truth data for the reflectance (of the objects tested). In the middle, (b), we show a bar-chart image blurred with the phase errors from our rotatable phase plate and the corresponding phase-corrected image obtained using DH-MBIR with multi-shot data, also shown on a normalized scale. On the right, (c), we show baseline truth data for the phase errors with units of radians.

LAB's Particle-Swarm global-optimization function. Each particle in the population evaluated the algorithm with a unique set of regularization parameters. We used the parameters that produced the highest value of q. For all other parameters not mentioned here, we used the values found in Pellizzari et al. [23].

Appendix E. Experimental testbed

In this Appendix, we provide additional figures to supplement the description of our experimental testbed. Fig. 9 shows the digital-holographic sensor and bar chart in the object plane. This picture supplements Fig. 4 from the main text.

Fig. 10 shows four reference images and phase errors. In Fig. 10(a) we show examples of the reference reflectance functions obtained using incoherent illumination on the left and speckle averaging on the right. The speckle-averaged images obtained using coherent detection have a larger dynamic range and contrast than the incoherent images. Additionally, we see wavelength differences in the reflectivity between these two image types. This is most apparent with the helmet visor of object number 2, which is clear when viewed with the broad-band incoherent light but opaque when viewed with the narrow-band laser illumination. For these reasons, we used speckle averaging to produce our reference reflectance functions.

The left side of Fig. 10(b) shows the blurry bar-chart images corresponding to the first four data sets. We used DH-MBIR with multi-shot data to produce a high-quality estimate of the phase errors for each bar chart. To verify the quality of our estimate, we applied the conjugate of these estimated phase-errors to the blurry bar-chart data. This phase-only correction, shown on the right side of Fig. 10(b), produced a focused image in each case. In Fig. 10(c), we show the corresponding estimates of the phase errors.

Given our methodology for obtaining baseline truth data, we developed the following procedure to collect data for each object tested. First, we measured 20 speckle realizations without the phase plate in the optical setup. Second, we inserted the phase plate into the optical setup and took a single measurement with the object being tested. Next, we replaced this object with the transmissive chrome-on-glass 1951 USAF bar chart backed by a sheet of white paper and kept the phase-plate position fixed. Lastly, we obtained five independent speckle realizations by rotating the bar chart relative to the digital-holographic sensor. Finally, we used MATLAB's *imregister* function to align these images and then average them.

References

[1] Abdelhamed A, Afifi M, Timofte R, Brown MS. NTIRE 2020 challenge on real image denoising: dataset, methods and results. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops; 2020. p. 496–7.

[2] Azad A.. KerasDnCNN - keras implementation of DnCNN-S. https://github.com/aGIToz/KerasDnCNN; 2018.

[3] Bate T, Spencer MF, Pellizzari CJ. Model-based digital holographic imaging using mulit-shot data. In: Unconventional imaging and adaptive optics 2021, vol. 11836. International Society for Optics and Photonics; 2021. p. 118360D.

[4] Bouman CA. Foundations of computational imaging: a model-based approach, vol 180. SIAM; 2022.

[5] Buzzard G.T., Chan S.H., Sreehari S., Bouman C.A.. Plug-and-play unplugged: optimization free reconstruction using consensus equilibrium. arXiv preprint arXiv:1705.08983 2017.

[6] Buzzard GT, Chan SH, Sreehari S, Bouman CA. Plug-and-play unplugged: optimization-free reconstruction using consensus equilibrium. SIAM J Imaging Sci 2018;11(3):2001–20.

[7] Chan S.H., Wang X., Elgendy O.A.. Plug-and-Play ADMM for image restoration: fixed point convergence and applications. arXiv preprint arXiv:1605.01710 2016.

[8] Goodman JW. Speckle phenomena in optics, theory and applications. Englewood Colorado: Roberts and Company; 2006.

[9] He T, Sun Y, Chen B, Qi J, Liu W, Hu J. Plug-and-play inertial forward–backward algorithm for poisson image deconvolution. J Electron Imaging 2019;28(4):043020.

[10] Horst S, Radosevich CJ, Pellizzari CJ, Spencer MF. Measuring the fried parameter of transmissive phase screens using digital-holographic detection. In: Dolne JJ,

Spencer MF, Testorf ME, editors. Unconventional and indirect imaging, image reconstruction, and wavefront sensing 2019, vol. 11135. International Society for Optics and Photonics; SPIE; 2019. p. 83–90. doi:10.1117/12.2529680.

[11] Kamilov U.S., Bouman C.A., Buzzard G.T., Wohlberg B.. Plug-and-play methods for integrating physical and learned models in computational imaging. arXiv preprint arXiv:2203.17061 2022.

[12] Kamilov US, Mansour H, Wohlberg B. A plug-and-play priors approach for solving nonlinear imaging inverse problems. IEEE Signal Process Lett 2017;24(12):1872–6.

[13] Lattari F, Gonzalez Leon B, Asaro F, Rucci A, Prati C, Matteucci M. Deep learning for SAR image despeckling. Remote Sens 2019;11(13):1532.

[14] Lexitek. http://www.lexitek.com/optics.html.

[15] Liu J, Sun Y, Eldeniz C, Gan W, An H, Kamilov US. Rare: image reconstruction using deep priors learned without groundtruth. IEEE J Sel Top Signal Process 2020;14(6):1088–99.

[16] MathWorks. Peak signal-to-noise ratio (PSNR). 2022a. https://www.mathworks.com/help/images/ref/psnr.html?searchHighlight=psnr&s_tid=srchtitle_psnr_1.

[17] MathWorks. Structural similarity (SSIM) index for measuring image quality. 2022b. https://www.mathworks.com/help/images/ref/ssim.html?searchHighlight=ssim&s_tid=srchtitle_ssim_1.

[18] Pellizzari C, Banet MT, Spencer MF, Bouman CA. Demonstration of single-shot digital holography using a Bayesian framework. JOSA A 2017;35(1):103–7.

[19] Pellizzari C, Trahan R III, Zhou H, Williams S, Williams S, Nemati B, Shao M, Bouman CA. Optically coherent image formation and denoising using plug and play inversion framework. JOSA 2017.

[20] Pellizzari CJ, Spencer MF, Bouman CA. Phase-error estimation and image reconstruction from digital-holography data using a Bayesian framework. JOSA A 2017;34(9):1659–69.

[21] Pellizzari CJ, Spencer MF, Bouman CA. Optically coherent image reconstruction in the presence of phase errors using advanced-prior models. In: Long-range imaging III, vol. 10650. International Society for Optics and Photonics; 2018. p. 106500B.

[22] Pellizzari CJ, Spencer MF, Bouman CA. Imaging through distributed-volume aberrations using single-shot digital holography. JOSA A 2019;36(2):A20–33.

[23] Pellizzari CJ, Spencer MF, Bouman CA. Coherent plug-and-play: digital holographic imaging through atmospheric turbulence using model-based iterative reconstruction and convolutional neural networks. IEEE Trans Comput Imaging 2020;6:1607–21.

[24] Pellizzari CJ, Trahan R, Zhou H, Williams S, Williams SE, Nemati B, et al. Synthetic aperture ladar: a model-based approach. IEEE Trans Comput Imaging 2017;3(4):901–16.

[25] Poon T-C, Liu J-P. Introduction to modern digital holography: with MATLAB. Cambridge University Press; 2014.

[26] Pradeep S, Nirmaladevi P. A review on speckle noise reduction techniques in ultrasound medical images based on spatial domain, transform domain and CNNmethods. In: IOP conference series: materials science and engineering, vol. 1055. IOP Publishing; 2021. p. 012116.

[27] Radosevich CJ, Pellizzari CJ, Horst S, Spencer MF. Imaging through deep turbulence using single-shot digital holography data. Opt Express 2020;28(13):19390–401.

[28] Romano Y, Elad M, Milanfar P. The little engine that could: regularization by denoising (RED). SIAM J Imaging Sci 2017;10(4):1804–44.

[29] Shi F, Cai N, Gu Y, Hu D, Ma Y, Chen Y, et al. Despecnet: a CNN-based method for speckle reduction in retinal optical coherence tomography images. Phys Med Biol 2019;64(17):175010.

[30] Spencer MF. Spatial heterodyne. In: Guenther BD, Steel DG, editors. Encyclopedia of modern optics. Oxford: Elsevier; 2018. p. 369–400. ISBN 978-0-12-814982-9. doi:10.1016/B978-0-12-803581-8.09416-9.

[31] Spencer MF, Raynor RA, Banet MT, Marker DK. Deep-turbulence wavefront sensing using digital-holographic detection in the off-axis image plane recording geometry. Opt Eng 2017;56(3):031213.

[32] Sreehari S, Venkatakrishnan SV, Wohlberg B, Buzzard GT, Drummy LF, Simmons JP, et al. Plug-and-play priors for bright field electron tomography and sparse interpolation. IEEE Trans Comput Imaging 2016;2(4):408–23. doi:10.1109/TCI.2016.2599778.

[33] Sridhar V, Wang X, Buzzard GT, Bouman CA. Distributed iterative CT reconstruction using multi-agent consensus equilibrium. IEEE Trans Comput Imaging 2020;6:1153–66. doi:10.1109/TCI.2020.3008782.

[34] Thurman ST. Phase-error correction in digital holography using single-shot data. JOSA A 2019;36(12):D47–61.

[35] Tippie AE, Fienup JR. Multiple-plane anisoplanatic phase correction in a laboratory digital holography experiment. Opt Lett 2010;35(19):3291–3.

[36] Tirer T, Giryes R. Super-resolution via image-adapted denoising CNNs: incorporating external and internal learning. IEEE Signal Process Lett 2019;26(7):1080–4. doi:10.1109/LSP.2019.2920250.

[37] Venkatakrishnan SV, Bouman CA, Wohlberg B. Plug-and-play priors for model based reconstruction. In: Global conference on signal and information processing (GlobalSIP), 2013 IEEE. IEEE; 2013. p. 945–8.

Lt Col Casey Pellizzari is an Associate Professor in the Department of Physics, U.S. Air Force Academy, Colorado Springs, Colorado. He entered the Air Force in 2005 as a graduate from California State University of Fullerton. Lt Col Pellizzari earned a doctorate in electrical engineering from Purdue University and continues his research in computational sensing, directed energy, and space domain awareness.

Capt Timothy Bate is a Senior Instructor of Physics at the U.S. Air Force Academy and the Advisor-in-Charge for Physics majors. He graduated from Clemson University with B.S. degrees in electrical engineering and in computer cngineering in 2015 and a M.S. in applied physics at the Air Force Institute of Technology in 2020. Capt Bate's research focuses on nonlinear optics, computational sensing, and directed energy.

Cadet First Class Kevin Donnelly is an undergraduate student in the Department of Astronautical Engineering at the U.S. Air Force Academy. Upon graduation in 2023, he will attend Rochester Institute of Technology to attain a master's degree in Imaging Science before his follow-on assignment at the National Reconnaissance Office. Cadet Donnelly's current research focuses on computational sensing, non-linear attitude controllers, and high-altitude balloon launch platforms.

Gregory T. Buzzard is a Professor of Mathematics at Purdue University. His current research focuses on measurement and reconstruction algorithms in imaging. At a more fundamental level, he helped developed the foundations of Plug and Play, which uses Multi Agent Consensus Equilibrium to reconcile sensing data for applications such as denoising. The work found in this paper builds on these foundations in a highly novel way.

Charles A. Bouman received a B.S.E.E. degree from the University of Pennsylvania in 1981 and a M.S. degree from the University of California at Berkeley in 1982. From 1982 to 1985, he was a full staff member at MIT Lincoln Laboratory and in 1989 he received a Ph.D. in electrical engineering from Princeton University. He joined the faculty of Purdue University in 1989 where he is currently the Showalter Professor of Electrical and Computer Engineering and Biomedical Engineering.

Mark F. Spencer is currently the Directed Energy Staff Specialist at Headquarters U.S. Indo-Pacific Command as the first-ever liaison from the Air Force Research Laboratory, Directed Energy Directorate. He is also an Adjunct Associate Professor of Optical Sciences and Engineering at the Air Force Institute of Technology within the Department of Engineering Physics. Mark is a fellow of SPIE and a senior member of Optica.