SIAM J. IMAGING SCIENCES
Vol. 14, No. 1, pp. 198–223

© 2021 Society for Industrial and Applied Mathematics

Patch-Based Holographic Image Sensing*

Alfred Marcel Bruckstein†, Martianus Frederic Ezerman‡, Adamas Aqsa Fahreza§, and San Ling§

Abstract. Holographic representations of data enable distributed storage with progressive refinement when the stored packets of data are made available in any arbitrary order. In this paper, we propose and test patch-based transform coding holographic sensing of image data. Our proposal is optimized for progressive recovery under random order of retrieval of the stored data. The coding of the image patches relies on the design of distributed projections ensuring best image recovery, in terms of the ℓ_2 norm, at each retrieval stage. The performance depends only on the number of data packets that have been retrieved thus far. Several possible options to enhance the quality of the recovery while changing the size and number of data packets are discussed and tested. This leads us to examine several interesting bit-allocation and rate-distortion trade-offs, highlighted for a set of natural images with ensemble estimated statistical properties.

Key words. holographic representation, mean squared error estimation, stochastic image data, Wiener filter

AMS subject classifications. 60G35, 68U10, 94A08, 94A12

DOI. 10.1137/20M1324041

1. Introduction.

Holographic sensing and representations aim to capture and describe signals, images, videos, and other information sources in a way that enables their recovery at various levels of resolution. The quality of the recovered data is dependent only on the size of the data portion that is available. Often, a mere preview of the visual information at some crude resolution level is sufficient to decide whether one needs to have the full detailed description. In such cases, holographic storage of the data is clearly beneficial.

Holographic representations of images were first proposed by Bruckstein, Holt, and Netravali in [6]. Their ideas exploited the redundancy in images, based on either subsampling strategies or Fourier transforms using a random phase mask, to ensure diffusion of low frequency information over all portions of the data. These ideas were then tested on individual

*Received by the editors March 9, 2020; accepted for publication (in revised form) November 17, 2020; published electronically February 18, 2021.
https://doi.org/10.1137/20M1324041
Funding: The work of the authors was supported by the National Research Foundation Singapore and the Israel Science Foundation under their joint program NRF2015-NRF-ISF001-2597 and by Nanyang Technological University grant M4080456.

†Department of Computer Science, Technion, Israel Institute of Technology, Haifa 32000, Israel, and School of Physical and Mathematical Sciences, Nanyang Technological University, 21 Nanyang Link, Singapore 637371 (freddy@cs.technion.ac.il, ambruckstein@ntu.edu.sg).
‡Corresponding author. School of Physical and Mathematical Sciences, Nanyang Technological University, 21 Nanyang Link, Singapore 637371 (fredezerman@ntu.edu.sg).
§School of Physical and Mathematical Sciences, Nanyang Technological University, 21 Nanyang Link, Singapore 637371 (adamas@ntu.edu.sg, lingsan@ntu.edu.sg)

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

images, but no theoretical evaluations on the expected achievable quality in the image recovery process were provided.

Several follow-up works, e.g., [11, 9, 10], came up with improved transforms for more efficient holographic encoding processes. Another approach, given in [7], proposed jittered quantization to achieve better precision with multiple instances of *compressed* image data exploiting differently quantized transform coefficients. A subsampling method for generating holographic data streams that ensure a uniform spatial spread of the sampled locations using low discrepancy sampling patterns can be found in [5].

In information theory, the issue of multiple representations has often been raised. Notable among them are the works of Goyal et al. in [12] and Kutyniok et al. in [15]. They dealt with coding into packets of data when communication processes may drop some of the packets in unpredictable ways. An earlier work [21] of Vaishampayan discussed multiple representations of a random variable via shifted or staggered quantization tables, a very nice idea resulting in better data fidelity as more and more quantized data are provided. Servetto et al. continued on this trend of ideas in [20] by proposing some high-performance wavelet-based multiple description image coding algorithms.

The insight in multiple description coding (MDC) is to encode an information source into multiple packets which are then sent through parallel channels, each of which is *errorless* and *bandwidth constrained*. There are a number of receivers that, upon receiving a subset of the packets, are tasked with the reconstruction. Most treatments in the literature focus on providing optimized performance in dealing with packet erasures, in various channel models, via the formalism of tight frames.

A general problem in MDC is to characterize the *achievable rate region* under specific distortion constraints on chosen reconstruction models. Detailed investigations can be found, e.g., in the works by Goyal, Kovačević, and Kelner [13] Venkataramani, Kramer, and Goyal [22] and in the two-part works [18] and [19] of Pradhan, Puri, and Ramchandran. Reaching their clearly stated performance predictions, however, remains an unattainable practical goal. This is mostly due to two factors. First, designing optimally suited tight frames for arbitrary numbers of erasures is highly nontrivial, although a clearer picture is available for one or two erasures. This fact is also evident in [15]. Second, attempts to establish multierasure information theoretic achievability results have not been successful.

Recently we provided an in-depth analysis of a holographic sensing paradigm, in the classical setting of Wiener filtering, on general random vector data in [4]. In the proposal, probings are performed by projection operators that eventually enable graceful successive refinement of the random vectors. The probing projection designs ensure the nice property that only the *number of available probings* determines the quality of the recovery. The *order* of their arrival is irrelevant.

Another interesting approach to holographic coding of data combines standard data compression procedures with shifts in the image plane to obtain a set of compressed versions of images that form the distributed packets of data [8]. In this case, again, random combinations of the compressed versions will yield improvement in the quality of the decompressed images, depending on the number of versions available.

In this paper, we apply the paradigm in [4] on images, seen as ensembles of smaller patches which are interpreted as realizations of a random process, in place of random vectors. In this important application, instead of merely repeating the relevant process in [4], we carefully and

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

deeply exploit special statistical properties of image data vectors. Furthermore, we formulate explicit probing policies guided by either estimated or modeled statistical image properties. We solve the problem of optimal probing allocations *when all packets are available*, and then discuss on how to distribute them so as to achieve a graceful and smooth degradation *when the packet numbers are decreased in random orderings*. Our strategies fulfill the goals of holographic representations on images.

We design the probing projections, based on the statistics derived from a given data set of images, in sections 2 and 3. The probings are guided either by the joint statistical properties of all the images in the set or by the properties of the patches of the image that is being considered for distributed holographic storage or communication. We ran our projection design process, along with the associated optimal bit allocation rule, on a set of natural images from various sources, described in section 4, and tested the image retrieval quality for various levels of noise, assumed to affect the holographic data acquisition process. The lowest noise level was actually modeling relatively noise-free projections. The higher levels were considerably deteriorating the projections that provide the distributed packets of images. The implemented system allows for modifications on the size of the image patches, the computation of the statistics for the patch ensemble to be encoded, and the number of holographic projections desired. Then, based on the ensemble's autocorrelation's spectrum of eigenvalues, we derive the projection operators and compute the predicted performance curves for recovery from any number of projections packets selected in either random or incremental orderings. The theoretical predictions are then compared to actual image recovery performance. We describe the implementation routines and report the outcomes in sections 5 and 6.

Our present approach, while suboptimal, has excellent properties. Its reliance on straightforward projection operators makes it simple and easy to understand. It yields clearly stated and comprehensible performance results. It provides a readily implemented scheme for holographic representations of images in the presence of noise. In terms of comparison to prior frame-based proposals, we have been unable to find comparable implementation routines that can be adapted with reasonable efforts to measure our approach against. We share a basic implementation software publicly to pave the way for comparative analysis in the future.

The following notational conventions are used throughout. Let $0 < k < \ell$ be integers. Denote by $[\![\ell]\!]$ the set $\{1, 2, \ldots, \ell\}$ and by $[\![k, \ell]\!]$ the set $\{k, k+1, \ldots, \ell\}$. Let $\mathbb{Z}$, $\mathbb{N}$, and $\mathbb{R}$ denote, respectively, the set of integers, the set of positive integers, and the field of real numbers. Vectors are expressed as columns and denoted by bold lowercase letters. Matrices are represented by either bold uppercase letters or upper Greek symbols. An $n \times n$ diagonal matrix with diagonal entries $v_j : j \in [\![n]\!]$ is denoted by $\mathrm{diag}(v_1, v_2, \ldots, v_n)$. The identity matrix is $\mathbf{I}$ or $\mathbf{I}_n$ if the dimension n is important. Concatenation of vectors or matrices is signified by the symbol | between the components. The transpose of a matrix $\mathbf{A}$ is $\mathbf{A}^\top$.

2. Our distributed sensing model and recovery procedure.

The data of interest in our investigation is a random process, i.e., a set of random column vectors $\{\mathbf{x}_\omega : \omega \in \Omega\}$ indexed by a set $\Omega \subset \mathbb{N}$. The process, of dimension M over $\mathbb{R}$, is characterized by its expectation $\overline{\mathbf{x}}$ and its $M \times M$ autocorrelation matrix

$$\mathbf{R}_{xx} := \mathbb{E}(\mathbf{x}_\omega - \overline{\mathbf{x}})(\mathbf{x}_\omega - \overline{\mathbf{x}})^\top. \tag{2.1}$$

It is assumed that $\overline{\mathbf{x}}$ is stored and available, and we deal with the centered case of $\overline{\mathbf{x}} = \mathbf{0}$ and consider the diagonalization of $\mathbf{R}_{xx}$, which is a positive definite symmetric matrix, with

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

spectral decomposition given by $\mathbf{R}_{xx} = \Psi\Lambda\Psi^\top$. Here $\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_M)$, with $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M > 0$, is a diagonal matrix and $\Psi = [\Psi_1|\Psi_2|\ldots|\Psi_M]$ is a unitary matrix whose columns, Ψ_j for $j \in [\![M]\!]$, are the eigenvectors corresponding to the respective eigenvalues λ_j.

Let $m < M$ be a fixed integer. We denote by $\mathbf{P}_k$ the projection matrix $\mathbf{U}_k\mathbf{U}_k^\top$, where $\mathbf{U}_k$ is the $M \times m$ matrix that displays the kth element of an orthonormal basis of the m-dimensional subspace V_k of $\mathbb{R}^M$ that $\mathbf{P}_k$ projects onto. The assumed model to sense realizations of the process $\{\mathbf{x}_\omega : \omega \in \Omega\}$ is the set of $N \in \mathbb{N}$ vectors

$$\{\mathbf{z}_k := \mathbf{U}_k^\top \Psi^\top \mathbf{x}_\omega + \mathbf{n}_k : k \in [\![N]\!]\}.$$

Here $\{\Psi^\top \mathbf{x}_\omega\}$ consists of column vectors with M entries with autocorrelation

$$\mathbb{E}[\Psi^\top \mathbf{x}_\omega \mathbf{x}_\omega^\top \Psi] = \Psi^\top \mathbf{R}_{xx} \Psi = \Lambda.$$

Hence, the realizations of the random process $\{\Psi^\top \mathbf{x}_\omega\}$ are vectors with uncorrelated entries having variances $\lambda_1, \lambda_2, \ldots, \lambda_M$. The noise vectors $\mathbf{n}_k$ affecting the probings are realizations of a random process with zero mean and autocorrelation $\sigma_n^2 \mathbf{I}_m$ and are assumed to be independent of $\mathbf{x}_\omega$. This model of sensing implements, first, a *decorrelating transform coding* on the data vector $\mathbf{x}_\omega$, then projects $\Psi^\top \mathbf{x}_\omega$ via $\mathbf{U}_k^\top$ onto an m-dimensional subspace of $\mathbb{R}^M$ for a fixed $m < M$.

There are many possible ways to define the projections, e.g., $\mathbf{U}_k$ can be any $M \times m$ matrix whose m column vectors, each having M entries, are orthogonal. We choose $\mathbf{U}_k$ to have columns selected from the standard basis

$$\left\{\mathbf{b}_1 := \begin{bmatrix}1 & 0 & 0 & \ldots & 0\end{bmatrix}^\top, \mathbf{b}_2 := \begin{bmatrix}0 & 1 & 0 & \ldots & 0\end{bmatrix}^\top, \ldots, \mathbf{b}_M := \begin{bmatrix}0 & \ldots & 0 & 1\end{bmatrix}^\top\right\}$$

of $\mathbb{R}^M$. In this case the projection $\mathbf{P}_k$ is a diagonal $M \times M$ matrix with m ones at specific locations on the diagonal as determined by the selection of the standard basis vectors in $\mathbf{U}_k$. Thus, our sensing model provides N vectors, each of dimension m, that convey the sensing information on the vectors $\Psi^\top \mathbf{x}_\omega$.

When all N vectors $\{\mathbf{z}_k : k \in [\![N]\!]\}$ are available, we can recover $\mathbf{x}_\omega$ by computing the best linear estimate under the given probing model at the noise level σ_n^2 in all of the observations. This recovery process provides the expected least-squared optimal solution for the recovery of the data $\{\mathbf{x}_\omega : \omega \in \Omega\}$ when both the data and the noise, seen as random processes, are Gaussian.

The best estimator for $\mathbf{x}_\omega$, given any set of ℓ out of N packets

$$\left\{\mathbf{z}_{k_j} = \mathbf{U}_{k_j} \Psi^\top \mathbf{x}_\omega + \mathbf{n}_{k_j} : j \in [\![\ell]\!]\right\}, \tag{2.2}$$

is expressed, in the classical Wiener filter, by

$$\widehat{\mathbf{x}}_\omega := \mathbf{R}_{x z_{\mathrm{combi}}} \left(\mathbf{R}_{z_{\mathrm{combi}} z_{\mathrm{combi}}}\right)^{-1} \mathbf{z}_{\mathrm{combi}} \tag{2.3}$$

from the observation

$$\mathbf{z}_{\mathrm{combi}} = \underbrace{\begin{bmatrix}\mathbf{z}_{k_1} \\ \mathbf{z}_{k_2} \\ \cdots \\ \mathbf{z}_{k_\ell}\end{bmatrix}}_{(\ell \cdot m) \times 1} = \underbrace{\begin{bmatrix}\mathbf{U}_{k_1}^\top \\ \mathbf{U}_{k_2}^\top \\ \cdots \\ \mathbf{U}_{k_\ell}^\top\end{bmatrix}}_{(\ell \cdot m) \times M} \Psi^\top \mathbf{x}_\omega + \underbrace{\begin{bmatrix}\mathbf{n}_{k_1} \\ \mathbf{n}_{k_2} \\ \cdots \\ \mathbf{n}_{k_\ell}\end{bmatrix}}_{(\ell \cdot m) \times 1}.$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

The matrices $\mathbf{R}_{xz_{\text{combi}}}$ and $\mathbf{R}_{z_{\text{combi}}z_{\text{combi}}}$ are readily obtained, respectively, as follows:

$$\begin{aligned}\mathbf{R}_{xz_{\text{combi}}} &= \mathbb{E}\left[\mathbf{x}_\omega \mathbf{z}_{\text{combi}}^\top\right] = \mathbf{R}_{xx}\left[\Psi\mathbf{U}_{k_1}|\Psi\mathbf{U}_{k_2}|\dots|\Psi\mathbf{U}_{k_\ell}\right] \\ &= \mathbf{R}_{xx}\Psi\left[\mathbf{U}_{k_1}|\mathbf{U}_{k_2}\dots|\mathbf{U}_{k_\ell}\right], \text{ and} \\ \mathbf{R}_{z_{\text{combi}}z_{\text{combi}}} &= \mathbb{E}\left[\mathbf{z}_{\text{combi}}\mathbf{z}_{\text{combi}}^\top\right] = \begin{bmatrix}\mathbf{U}_{k_1}^\top \\ \mathbf{U}_{k_2}^\top \\ \dots \\ \mathbf{U}_{k_\ell}^\top\end{bmatrix}\Lambda\left[\mathbf{U}_{k_1}|\mathbf{U}_{k_2}\dots|\mathbf{U}_{k_\ell}\right] + \sigma_n^2\mathbf{I}_{(\ell\cdot m)}.\end{aligned}$$

Since $\mathbf{R}_{xx} = \Psi\Lambda\Psi^\top$, the best reconstructed $\mathbf{x}_\omega$ from the given data is therefore

$$\widehat{\mathbf{x}}_\omega = \Psi\Lambda\left[\mathbf{U}_{k_1}|\mathbf{U}_{k_2}\dots|\mathbf{U}_{k_\ell}\right]\left(\begin{bmatrix}\mathbf{U}_{k_1}^\top \\ \mathbf{U}_{k_2}^\top \\ \dots \\ \mathbf{U}_{k_\ell}^\top\end{bmatrix}\Lambda\left[\mathbf{U}_{k_1}|\mathbf{U}_{k_2}\dots|\mathbf{U}_{k_\ell}\right] + \sigma_n^2\mathbf{I}_{(\ell\cdot m)}\right)^{-1}\mathbf{z}_{\text{combi}}. \tag{2.4}$$

The expected squared-error covariance in the reconstruction is therefore given by

$$\begin{aligned}\mathbf{R}_{ee} = \mathbb{E}\left[\left(\mathbf{x}_\omega - \widehat{\mathbf{x}}_\omega\right)\left(\mathbf{x}_\omega - \widehat{\mathbf{x}}_\omega\right)^\top\right] &= \left(\mathbf{R}_{xx}^{-1} + \frac{1}{\sigma_n^2}\Psi\left[\mathbf{U}_{k_1}|\mathbf{U}_{k_2}|\dots|\mathbf{U}_{k_\ell}\right]\begin{bmatrix}\mathbf{U}_{k_1}^\top \\ \mathbf{U}_{k_2}^\top \\ \dots \\ \mathbf{U}_{k_\ell}^\top\end{bmatrix}\Psi^\top\right)^{-1} \\ = \left(\Psi\Lambda^{-1}\Psi^\top + \frac{1}{\sigma_n^2}\Psi\sum_{j=1}^{\ell}\mathbf{P}_{k_j}\Psi^\top\right)^{-1} &= \Psi\left(\Lambda^{-1} + \frac{1}{\sigma_n^2}\sum_{j=1}^{\ell}\mathbf{P}_{k_j}\right)^{-1}\Psi^\top,\end{aligned} \tag{2.5}$$

where $\mathbf{P}_{k_j} = \mathbf{U}_{k_j}\mathbf{U}_{k_j}^\top$. Our choice of the basis for $\mathbb{R}^M$ implies $\mathbf{U}_{k_j} := [\mathbf{b}_{t_1}|\mathbf{b}_{t_2}|\dots|\mathbf{b}_{t_m}]$ is a selection of m out of M standard basis vectors of $\mathbb{R}^M$. This structure yields an explicit form for the expected mean squared error (MSE) since we have

$$\sum_{j=1}^{\ell}\mathbf{P}_{k_j} = \operatorname{diag}(s_1, s_2, \dots, s_M),$$

where s_i is the total number of times that the basis vector $\mathbf{b}_i$ appears in the combined matrix $[\mathbf{U}_{k_1}|\mathbf{U}_{k_2}|\dots|\mathbf{U}_{k_\ell}]$. Thus, the error covariance in (2.5) becomes

$$\begin{aligned}\mathbf{R}_{ee} &= \Psi\left(\Lambda^{-1} + \frac{1}{\sigma_n^2}\operatorname{diag}(s_1, s_2, \dots, s_M)\right)^{-1}\Psi^\top \\ &= \Psi\ \operatorname{diag}\left(\frac{\lambda_1}{1+\frac{\lambda_1}{\sigma_n^2}s_1}, \frac{\lambda_2}{1+\frac{\lambda_2}{\sigma_n^2}s_2}, \dots, \frac{\lambda_M}{1+\frac{\lambda_M}{\sigma_n^2}s_M}\right)\ \Psi^\top.\end{aligned}$$

The total MSE in estimating $\mathbf{x}_\omega$ is the trace of the error covariance matrix, i.e.,

$$\text{MSE}_{\text{total}} = \operatorname{Tr}(\mathbf{R}_{ee}) = \sum_{i=1}^{M}\frac{\lambda_i}{1+\frac{\lambda_i}{\sigma_n^2}s_i}, \tag{2.6}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

where s_i counts the number of times the ith entry of $\Psi^\top \mathbf{x}_\omega$ has been probed in the sensing process.

In summary, we have explicit expressions for the MSE incurred in the recovery procedure from any set of combined sensing results. We can therefore use this result to guide the design of the actual sensing process. The sensing process yields N vectors of length m to be distributed as packets of acquired sensing results. As mentioned earlier, the data consist of random vectors of M entries with zero mean and known autocorrelation of a stochastic process.

We now turn our focus to the design of the N projections $\mathbf{P}_1, \mathbf{P}_2, \ldots, \mathbf{P}_N$, i.e., on selecting the matrices $\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_N$ that satisfy the following holographic criteria. First, the distributed restoration must have the progressive refinement property as more packets are utilized. Second, the process must be smooth in the sense that it achieves maximal uniformity and low MSE estimation when any ℓ out of N packets of representation are made available for every $\ell \in [\![N]\!]$.

3. The holographic sensing design.

This section explicitly describes the steps of the projection design optimization in terms of the packet allocation process to ensure that the recovery process from randomly ordered packets meets the holographic requirements. The design is presented here in three subsections to mirror the progression in our optimization strategies. First, we think of the design as a general *resource allocation problem*, where we are given some number K of probings to allocate. Second, we include several practical considerations. Third, we propose a simple way to build the projection matrices for the actual sensing.

3.1. The optimal probing trade-offs.

We regard the problem of sensing design as a resource allocation, i.e., rate distortion trade-off process. We have random vectors $\mathbf{y}_\omega = \Psi^\top \mathbf{x}_\omega \in \mathbb{R}^M$ with uncorrelated entries of variances $\lambda_1, \ldots, \lambda_M$ and we can probe batches of m entries, with some small additive independent noise of zero mean and variance σ_n^2. We want to recover $\mathbf{x}_\omega$, from a certain number of m-probes, via the optimal Wiener filter estimation process. The combined total expected MSE in estimating the entries of $\mathbf{x}_\omega$ depends on the number of times $s_1, s_2, \ldots, s_M$ each entry is probed since

$$\mathrm{MSE}\left(\Lambda, \sigma_n^2, \{s_1, s_2, \ldots, s_M\}\right) = \sum_{i=1}^{M} \frac{\lambda_i}{1 + \frac{\lambda_i}{\sigma_n^2} s_i}, \tag{3.1}$$

as we have derived in (2.6). A natural question arises. How should we design the probes so as to obtain the smallest MSE, given that we select m locations to probe with each $\mathbf{U}_k \mathbf{U}_k^\top = \mathbf{P}_k$ projection matrix?

It is clear that if we have N projections and each of them selects m locations, then we have $\sum_{i=1}^{M} s_i = N \cdot m$. For any subset of size ℓ out of the N projections, i.e., for $\mathbf{U}_{k_1}, \mathbf{U}_{k_2}, \ldots, \mathbf{U}_{k_\ell}$ where $k_j \in [\![N]\!]$, we have a total of $\ell \cdot m$ probings. This leads us to consider the following questions.

Question 1: What is the lowest MSE attainable with a given total number of K probings?
Question 2: What is the optimal distribution of probings, i.e., the allocation of the $\ell \cdot m$ probings in total to the set $\{s_i\}$ of nonnegative integers that yields the lowest MSE?

We are thus led to the following optimization. Let K denote the total number of probings to be suitably distributed into ℓ probing projections. Given $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_M > 0$ and

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

σ_n^2, minimize, over all suitable $\{s_j : j \in [\![M]\!]\}$ allocations, the MSE in (3.1), subject to the requirements that $0 \le s_j \le N$ must be an integer for all $j \in [\![M]\!]$ and $\sum_{j=1}^{M} s_j = K$.

To address this resource allocation problem, we begin with the last condition on s_j, i.e., we relax the optimization over $\mathbb{R}$, instead of $\mathbb{Z}$. To make this relaxation clear, we replace $s_j \in \mathbb{Z}$ by $\zeta_j \in \mathbb{R}$. This reduces the problem to a straightforward Lagrangian optimization problem, requiring us to solve

$$\frac{\partial}{\partial \zeta_j}\left(\sum_{j=1}^{M} \frac{\lambda_j}{1+\frac{\lambda_j}{\sigma_n^2}\zeta_j} + \gamma\left(K - \sum_{j=1}^{M}\zeta_j\right)\right) = 0. \tag{3.2}$$

Solving it yields, for each $j \in [\![M]\!]$,

$$\frac{\lambda_j^2 \cdot \sigma_n^2}{\left(\sigma_n^2 + \lambda_j\zeta_j\right)^2} = \gamma \iff \left(\sigma_n^2 + \lambda_j\zeta_j\right)^2 = \frac{1}{\gamma}\lambda_j^2 \cdot \sigma_n^2,$$

which implies

$$\sigma_n^2 + \lambda_j\zeta_j = \frac{1}{\sqrt{\gamma}}\lambda_j \cdot \sigma_n.$$

Thus, the solution to the optimization problem is the set

$$\left\{\zeta_j = \frac{\sigma_n}{\sqrt{\gamma}} - \frac{\sigma_n^2}{\lambda_j} : j \in [\![M]\!]\right\}. \tag{3.3}$$

Now, adding the constraint $\sum_{k=1}^{M}\zeta_k = K$ means

$$\begin{aligned}\frac{\sigma_n}{\sqrt{\gamma}} \cdot M - \sigma_n^2\sum_{k=1}^{M}\frac{1}{\lambda_k} = K &\iff \frac{1}{\sqrt{\gamma}} = \frac{1}{M\cdot\sigma_n}\left(K + \sigma_n^2\sum_{k=1}^{M}\frac{1}{\lambda_k}\right)\\ &\iff \frac{1}{\sqrt{\gamma}} = \frac{K}{M\cdot\sigma_n} + \frac{\sigma_n}{M}\sum_{k=1}^{M}\frac{1}{\lambda_k}.\end{aligned}$$

Thus, we obtain

$$\zeta_j = \frac{K}{M} + \sigma_n^2\left(\frac{1}{M}\sum_{k=1}^{M}\frac{1}{\lambda_k} - \frac{1}{\lambda_j}\right), \tag{3.4}$$

which is very similar to bit allocation in transform coding.

The result we have just obtained is quite nice. The ζ_js have to be distributed about the even distribution of the K probings to the M entities in the probed vector, i.e., in the ratio $\frac{K}{M}$ according to how far their inverse variance is from the average inverse variances of all entries. The distribution, in other words, depends on the *harmonic mean of the variances*. Clearly, this result has not yet ensured that the other conditions imposed on the s_js, namely

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

s_j must be a nonnegative integer, upper bounded by N, for each j, are satisfied. Looking at the solutions for ζ_js, however, we realize that the function

$$\rho(j) = \frac{1}{M}\sum_{k=1}^{M}\frac{1}{\lambda_k} - \frac{1}{\lambda_j} \tag{3.5}$$

is a decreasing function of j. We therefore need to have

$$\begin{aligned}\frac{K}{M} + \sigma_n^2\left(\frac{1}{M}\sum_{k=1}^{M}\frac{1}{\lambda_k} - \frac{1}{\lambda_M}\right) \geq 0 \iff \frac{K}{M} &\geq \sigma_n^2\left(\frac{1}{\lambda_M} - \frac{1}{M}\sum_{k=1}^{M}\frac{1}{\lambda_k}\right) \\ \iff K &\geq M\cdot\sigma_n^2\left(\frac{1}{\lambda_M} - \frac{1}{M}\sum_{k=1}^{M}\frac{1}{\lambda_k}\right)\end{aligned} \tag{3.6}$$

to guarantee that ζ_M and, hence, $\zeta_1, \ldots, \zeta_{M-1}$ will all be nonnegative. If the above conditions are satisfied, then we will also have

$$\zeta_1 = \frac{K}{M} + \sigma_n^2\left(\frac{1}{M}\sum_{k=1}^{M}\frac{1}{\lambda_k} - \frac{1}{\lambda_1}\right) \leq K. \tag{3.7}$$

To see this, we notice that

$$\frac{1}{\lambda_1} + (M-1)\frac{1}{\lambda_M} \geq \sum_{k=1}^{M}\frac{1}{\lambda_k} = \frac{M-1}{M}\sum_{k=1}^{M}\frac{1}{\lambda_k} + \frac{1}{M}\sum_{k=1}^{M}\frac{1}{\lambda_k}$$

implies

$$(M-1)\left(\frac{1}{\lambda_M} - \frac{1}{M}\sum_{k=1}^{M}\frac{1}{\lambda_k}\right) \geq \frac{1}{M}\sum_{k=1}^{M}\frac{1}{\lambda_k} - \frac{1}{\lambda_1}.$$

Hence,

$$\frac{1}{\lambda_M} - \frac{1}{M}\sum_{k=1}^{M}\frac{1}{\lambda_k} \geq \frac{1}{M-1}\left(\frac{1}{M}\sum_{k=1}^{M}\frac{1}{\lambda_k} - \frac{1}{\lambda_1}\right).$$

Substituting this last expression into the inequality in (3.6) gives us

$$K \geq \frac{M}{M-1}\cdot\sigma_n^2\left(\frac{1}{M}\sum_{k=1}^{M}\frac{1}{\lambda_k} - \frac{1}{\lambda_1}\right),$$

which becomes the inequality in (3.7) after a simple rearrangement. Thus, we have determined a condition on K to ensure the nonnegativity of all the optimal values assigned to the nonincreasing sequence of real values $\zeta_1, \zeta_2, \ldots, \zeta_m$ whose sum is K.

Reflecting on our journey so far, minimizing the total MSE in estimating $\mathbf{y}_\omega$, hence $\mathbf{x}_\omega$, from K probings of its entries is achieved by distributing the probing projections according to the set of integers $s_1, s_2, \ldots, s_M$ determined via $\min_{\{s_j\}}\sum_{j=1}^{M}\frac{\lambda_j}{1+\frac{\lambda_j}{\sigma_n^2}s_j}$ such that $\sum_{j=1}^{M}s_j = K$.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

The minimal value is attained when the integer s_j, for each $j \in [\![M]\!]$, is "near" the real positive value

$$\zeta_j = \frac{K}{M} + \sigma_n^2 \left(\frac{1}{M} \sum_{k=1}^{M} \frac{1}{\lambda_k} - \frac{1}{\lambda_j} \right) \in \mathbb{R}. \tag{3.8}$$

Note that by using the real-valued ζ_j to compute for MSE in (3.1) we obtain

$$\begin{aligned} \mathrm{MSE}_{\mathrm{best}} &= \sum_{j=1}^{M} \frac{\lambda_j}{1 + \frac{\lambda_j}{\sigma_n^2} \left(\frac{K}{M} + \sigma_n^2 \left(\frac{1}{M} \sum_{k=1}^{M} \frac{1}{\lambda_k} - \frac{1}{\lambda_j} \right) \right)} = \sum_{j=1}^{M} \frac{\lambda_j}{\frac{\lambda_j}{\sigma_n^2} \frac{K}{M} + \lambda_j \left(\frac{1}{M} \sum_{k=1}^{M} \frac{1}{\lambda_k} \right)} \\ &= \sum_{j=1}^{M} \frac{1}{\frac{K}{M \cdot \sigma_n^2} + \frac{1}{M} \sum_{k=1}^{M} \frac{1}{\lambda_k}} = M \cdot \frac{\sigma_n^2}{\frac{K}{M} + \sigma_n^2 \left(\frac{1}{M} \sum_{k=1}^{M} \frac{1}{\lambda_k} \right)}. \end{aligned} \tag{3.9}$$

Thus, the optimal assignment is achieved when all of the errors in estimating the components of the random vector $\mathbf{y}_\omega$ are equalized. This is a very natural and oft-encountered condition in distributed estimation. Keep in mind that this is the case when K is large enough to make all of the s_js nonnegative, i.e., $K \geq M \cdot \sigma_n^2 (\frac{1}{\lambda_M} - \frac{1}{M} \sum_{k=1}^{M} \frac{1}{\lambda_k})$ from (3.6). Such a K ensures that the best MSE in estimating *each component* of $\mathbf{y}_\omega$ is bounded above by λ_M, i.e., it stays no more than the smallest variance λ_M in the uncorrelated random vector $\mathbf{y}_\omega$.

The *coding gain*, defined to be the *reduction* in the MSE value as a consequence of using the optimal $\{\zeta_j\}$ in (3.8) instead of $\zeta_j := \frac{K}{M}$ for all $j \in [\![M]\!]$, is

$$\sum_{j=1}^{M} \frac{\lambda_j}{1 + \frac{\lambda_j}{\sigma_n^2} \frac{K}{M}} - \sum_{j=1}^{M} \frac{\lambda_j}{1 + \frac{\lambda_j}{\sigma_n^2} \left(\frac{K}{M} + \sigma_n^2 \left(\frac{1}{M} \sum_{k=1}^{M} \frac{1}{\lambda_k} - \frac{1}{\lambda_j} \right) \right)} \geq 0.$$

The gain is 0 if and only if $\lambda_1 = \lambda_2 = \cdots = \lambda_M$.

One can plot the function $\rho(j) = \frac{1}{M} \sum_{k=1}^{M} \frac{1}{\lambda_k} - \frac{1}{\lambda_j}$ (see (3.5)), given the λ_j values. The curve displaying the allocation of ζ_j is then influenced by $\frac{K}{M}$, which serves as the constant level around which each ζ_j lives, starting above it before decreasing as j goes to M. The shape is determined by $\sigma_n^2 \cdot \rho(j)$ where $\rho(j)$ depends only on the diagonal entries $\lambda_1, \lambda_2, \ldots, \lambda_M$ of Λ. A plot of $\rho(j)$ based on actual λ_j values calculated from a set of images, alongside the plots of ζ_j for several values of σ_n^2, can be found later in Figure 4.

3.2. Practical calibrations. Given the nonincreasing sequence of variances

$$S := \lambda_1, \lambda_2, \ldots, \lambda_M,$$

the probing distribution is governed by the function $\rho(j)$ in (3.5). Letting

$$\Delta := \frac{1}{M} \sum_{j=1}^{M} \frac{1}{\lambda_j},$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

i.e., Δ is the *harmonic mean* of the elements in S, we write $\rho(j) = \Delta - \frac{1}{\lambda_j}$. Together with the assumed noise level and the total number K of probings, $\rho(j)$ provides us with the resource allocation strategy that achieves the best MSE performance.

In practical situations the sequence S of λ_js, the noise's standard deviation σ_n^2, and the total number of probings K are given to us as system parameters. It may happen, therefore, that K is not large enough or σ_n^2 not small enough to ensure that all ζ_js are nonnegative. When this is the case, we shall allocate no probings for a subsequence of small λ_js at the tail end of S. We accomplish this by repeatedly solving the probing allocation problem for leading subsequences of S, say, for $\lambda_1, \lambda_2, \ldots \lambda_L$ for an $L < M$. We recalculate $\zeta_1, \zeta_2, \ldots, \zeta_L$ for various L until we arrive at the *largest* L that makes

$$\zeta_i := \frac{K}{L} + \sigma_n^2 \left(\frac{1}{L} \sum_{j=1}^{L} \frac{1}{\lambda_j} - \frac{1}{\lambda_i} \right) > 0 \text{ for all } i \in [\![L]\!] \,, \tag{3.10}$$

implying $s_i \geq 0$, and fix $\zeta_{L+1} = \cdots = \zeta_M = 0$. The formula in (3.10) is derived by solving the suitably modified optimization problem (see (3.2))

$$\begin{aligned} \theta\left(\zeta_1, \ldots, \zeta_L, 0, \ldots, 0\right) &= \sum_{j=1}^{M} \frac{\lambda_j}{1 + \frac{\lambda_j}{\sigma_n^2}\zeta_j} + \gamma \left(\sum_{j=1}^{M} \zeta_j - K \right) \\ &= \sum_{j=1}^{L} \frac{\lambda_j}{1 + \frac{\lambda_j}{\sigma_n^2}\zeta_j} + \sum_{j=L+1}^{M} \lambda_j + \gamma \left(\sum_{j=1}^{L} \zeta_j - K \right). \end{aligned} \tag{3.11}$$

The allocation of the K probings is then done to the leading L entries of the vector $\mathbf{y}_\omega$ to achieve the goal of having holographic reconstructions for $K = N \cdot m$. In our setup, N is the number of packets created by the projections for each $\mathbf{y}_\omega$ and m is the fixed size of each probing packet, i.e., the number of entries in each packet, which is the dimension of the image vector after a projection. These N packets are to be stored or distributed in the environment or sent via some lossy communication channel that drops packets or randomizes their delivery time.

For practical examples that we encounter in the statistics of natural image patches, the elements in S are highly skewed toward the leading part, i.e., the first variance λ_1 is, relative to the rest of the entries, much larger. The next few variances tend to be high, then all the remaining variances decrease rapidly as the index goes to M. We use the following three modes of determining s_j from ζ_j for $j \in [\![L]\!]$. Their respective advantages become apparent in their actual deployment.

Mode 1: As much as possible, this mode assigns s_j to be the closest integer to ζ_j for $j \in [\![M]\!]$. It starts by listing the absolute distances from ζ_j to its nearest integer $\lfloor \zeta_j + \frac{1}{2} \rfloor$ and sorts these distances from smallest to largest, keeping tabs on the respective indices. Guided by this sorted list, the mode assigns s_j to be the nearest integer to ζ_j, choosing between $\lfloor \zeta_j \rfloor$ and $\lceil \zeta_j \rceil$, as i goes through the indices in the list. The enforcement decreases in priority, i.e., it may reverse the assignment to keep the condition $\sum_{j=1}^{M} s_j = K$ satisfied.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Mode 2: The idea is to initially assign as s_j the integer part $\lfloor \zeta_j \rfloor$ of ζ_j, for all $j \in [\![M]\!]$, and define an integer $\delta_1 := \sum_{j=1}^{M} (\zeta_j - \lfloor \zeta_j \rfloor)$. Since λ_1 has the highest variance coefficient, we assign a new value $s_1 \leftarrow \min(N, s_1 + \delta_1)$. With the updated s_1, let $\delta_2 := K - \sum_{j=1}^{M} s_j$. If $\delta_2 > 0$, then assign $s_2 \leftarrow \min(N, s_2 + \delta_2)$. One repeats this process until there is no other leftover quantity to assign.

Mode 3: This mode simply assigns $s_1 = \cdots = s_m = N$ and $s_k = 0$ for $k > m$.

3.3. Designing the packet projections. After obtaining the optimal allocation that yields the best distribution of the given K probings in terms of minimizing the MSE, i.e., having

$$s_1 \geq s_2 \geq \cdots \geq s_L \geq 0 \text{ and } s_{L+1} = \cdots = s_M = 0$$

for some $1 < L < M$, we need to probe only L first entries of $\mathbf{y}_\omega$. Recall that the vector has a given autocorrelation $\mathbf{R}_{yy} = \Lambda$. We now proceed to designing the N packets, each of size m, that will optimally distribute the $K = N \cdot m$ probings with the goal of reaching the best-possible recovery with the desired progressive refinement property.

For ℓ out of the total N packets, the number of available probings is $K_\ell := \ell \cdot m$. We combine (3.1) and the optimal ζ_js to infer that the best theoretically possible expected error for ℓ packets is given by

$$\text{MSE}_{\text{best},\ell} = \sum_{j=1}^{L} \frac{\lambda_j}{1 + \frac{\lambda_j}{\sigma_n^2} \zeta_j} + \sum_{j=L+1}^{M} \lambda_j. \tag{3.12}$$

Given specific Λ, m, and σ_n^2, we can explicitly compute $\text{MSE}_{\text{best},\ell}$ for each $\ell \in [\![N]\!]$. Plots for actual images will be given in section 5.

We shall never attain this optimal performance for every ℓ-set of projections, since we optimize the allocation of s_js for best recovery when *all* packets are available. We should, however, strive to design the N packets of projections in such a way that a random selection of ℓ out of N packets, i.e., selecting any one of the $\binom{N}{\ell}$ possible combinations of ℓ distinct packets, exhibits *smoothness*. In our context, it means that we recover the vector $\mathbf{y}_\omega$ as uniformly well as possible and as close as it can be to the $\text{MSE}_{\text{best},\ell}$ value.

Given the sequence S whose M elements are the diagonal entries of Λ, σ_n^2, and $K = N \cdot m$, we want to find the best way to assign the projections into N boxes satisfying the following conditions.

Condition 1: The projections sample the first, the second, and so on, up to the Lth entry of $\mathbf{y}_\omega$.

Condition 2: The $s_j : j \in [\![L]\!]$ add up, by design, to a total of $N \cdot m$ projections, i.e.,

$$s_1 + s_2 + \cdots + s_L = N \cdot m.$$

Condition 3: Choices of any ℓ out of N boxes, for each $\ell \in [\![N]\!]$, yield nearly as good a recovery of $\mathbf{y}_\omega$ as possible, without choices that yield significantly bad MSE.

We address the problem as follows. Every projection $\mathbf{P}_k = \mathbf{U}_k \mathbf{U}_k^\top$ is determined by a set of m basis vectors selected from the standard basis $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_M\}$ of $\mathbb{R}^M$. The numbers $s_1, s_2, \ldots, s_L$ tell us how many times each vector $\mathbf{b}_i$ will appear in the combined N projections,

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

written as $\sum_{k=1}^{N} \mathbf{P}_k$. The key to a good projection is to have as many low-indexed probings as possible since the various projections have different influence on the expected MSE of recovery. However, we want every pair of projections to provide similarly good results. The same should hold for any triplet, any quadruplet, and so on, of projections.

There are several options to distribute the $N \cdot m$ balls of L different colors into N boxes, each with capacity m. There are s_j balls of label j for each $j \in [\![L]\!]$,

$$\underbrace{1 \quad 1 \quad \dots \quad 1}_{s_1 \text{ balls labelled } 1} \quad \underbrace{2 \quad 2 \quad \dots \quad 2}_{s_2 \text{ balls labelled } 2} \quad \dots\dots \quad \underbrace{L \quad L \quad \dots \quad L}_{s_L \text{ balls labelled } L}.$$

Here is a simple yet effective distribution process. What we want is to have, in each of the N boxes, balls with low labels. Such balls contribute the most to the *reduction* of the expected MSE since they correspond to probing the low-indexed components of $\mathbf{y}_\omega$, which have the highest variances.

We mark the N boxes B_1 to B_N and stack them vertically with B_1 on top and B_N at the bottom. We then distribute the s_1 balls, labeled 1, one by one to boxes B_1 to B_{s_1}. As each ball is distributed, move the just-filled box from the very top to the bottom. Once all of these s_1 balls are taken care of, the (still empty) top box is now B_{s_1+1} and the bottom box is B_{s_1}. Repeat the distribution process on the remaining balls, with the boxes moved from top to bottom as before. Notice that at every instance after the top box is moved to the bottom position, the boxes are ordered in decreasing MSE, with the box on top corresponding to the highest current MSE, if the recovery process is to commence immediately. This way of allocating the probings to the projection boxes will eventually result in a complete allocation of $K = N \cdot m$ probings to the N boxes while making all of these N boxes as similar as possible in terms of their respective MSE measure of recovery quality.

4. The image data set used for testing. For a software implementation in section 5, we use a data set of 49 images available at https://github.com/adamasstokhorst/holographic/tree/master/img. Most of them are either part of the standard images used in image processing or taken from free-to-use online repositories https://www.pexels.com/ and https://pixabay.com/, with no attributions required. Figure 1 presents 6 of the 49.

The image `flood` in Figure 1(b) was taken by Fardin Oyan. Originally titled *Happiness on a Rainy Day*, it won him the Young Environmental Photographer of the Year 2018 award. Two images, `oldlady` in Figure 1(d) and `buffalo`, were included in the data set with the permission of their photographer, Rarindra Prakarsa. The data set also contains two iconic images, used under the "fair use principle" for academic purposes. The award-winning photographer Steve McCurry shot *Procession of Nuns* in Rangoon, Burma (now Yangon, Myanmar), in 1994. We call the image `monks`, for brevity. The image `refugee` of a Syrian refugee carying his daughter while crossing the border of Macedonia and Greece in 2015 was taken by the late decorated photographer Yannis Behrakis. The images `dragon` in Figure 1(a) and `baby` belong to the corresponding author.

5. A software implementation. To demonstrate the efficacy and versatility of holographic sensing on images, we design and thoroughly test an implementation of a distributed image sensing system. The images are then shown to be progressively recovered with the recovery

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

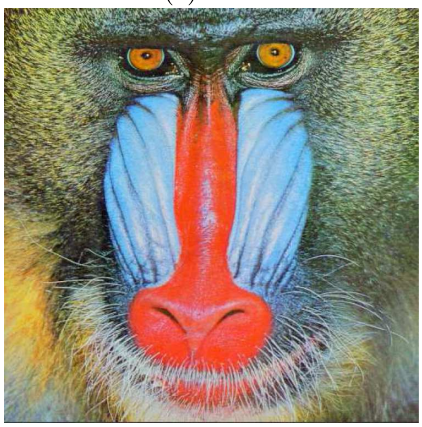
Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

(a) `dragon`

(b) `flood`

(c) `merlion`

(d) `oldlady`



(e) `mandrill`

(f) `fruit`

Figure 1. *Six of the images from the image data set, to be used as illustrations.*

being insensitive to and independent of the order in which the packets became available. Our software is written in `python 2.7` with `numpy` [2, 17], Python Imaging Library `PIL` [16], and `matplotlib` [1, 14] as the minimum required libraries.

The software implementation routines fall into three types, namely preparatory routines, recovery routines, and performance analysis routines. We welcome readers who are interested in implementing the tools for themselves to access the source files at https://github.com/adamasstokhorst/holographic. Users can tweak the input parameters to better suit their favorite implementation scenarios.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

5.1. Preparatory steps. In the preprocessing stage, the system analyzes a database of images to obtain the necessary statistical data to use in the sensing design. Dealing with typically large size images, as is customary, we break them up into small patches of $r \times r$ pixels, where r can be 4, 8, or 16. Each two-dimensional image is row-stacked into vectors of $M = r^2$ entries. Taken sequentially, from all images in the database, these patches form our ensemble of Ω vectors. These vectors are then analyzed to get their average and, subsequently, the centered autocorrelation. One can easily handle both black-and-white 8-bit per pixel images and color images, which are simply triplets of RGB 8-bit per pixel color planes of red, blue, and green added together. The resulting centered ensemble $\{\mathbf{x}_\omega : \omega \in \Omega\}$ consists of vectors with M entries and zero mean.

The original value range for the centered images, for 8-bit quantization, is between 0 and 255, with quantization step 1. The range is then converted in our software to the interval -1 to 1, with the quantization step adjusted accordingly. We set the noise level at $\sigma_n^2 \in \{0.01, 0.25, 0.64, 1.00\}$ for implementation on actual images. Recall that in most of the process, we are looking at $\mathbf{y}_\omega$, instead of $\mathbf{x}_\omega$, and, thus, using the diagonalized $\mathbf{R}_{xx} = \Psi\Lambda\Psi^\top$ via $\mathbf{R}_{yy} = \Lambda$. Once we have the average patch vector and the autocorrelation $\mathbf{R}_{xx}$ of the ensemble, we perform, for this ensemble, the holographic sensing and recovery processes as described in sections 2 and 3.

The Λ and Ψ Matrices. The autocorrelation matrices are of size $M \times M$, for a given M. We typically use $M = 64$. The standard singular value decomposition function in `numpy` allows for a fast computation of Λ and Ψ. Figure 2 displays the orthonormal set of eigenvectors as 8×8 patches in a combined image as a typical example that can be replicated for other values of M. As expected, these eigenpatches look similar to the patches defined by the classical orthonormal basis used in the discrete cosine transform [3]. These are the patches to invoke in reconstructing the images from the data packets that represent the sensed images.

Figure 3 presents the ordered eigenvalues for the patches, of size 8×8, of each of the 49 images in the data set separately as well as for the complete ensemble, i.e., the `aggregate`

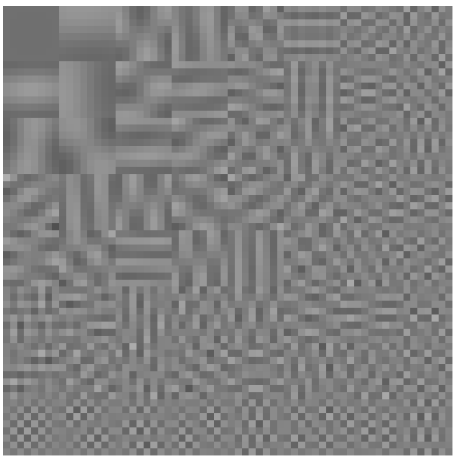


Figure 2. *A visual presentation of the orthonormal eigenvectors as 8×8 patches.*

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

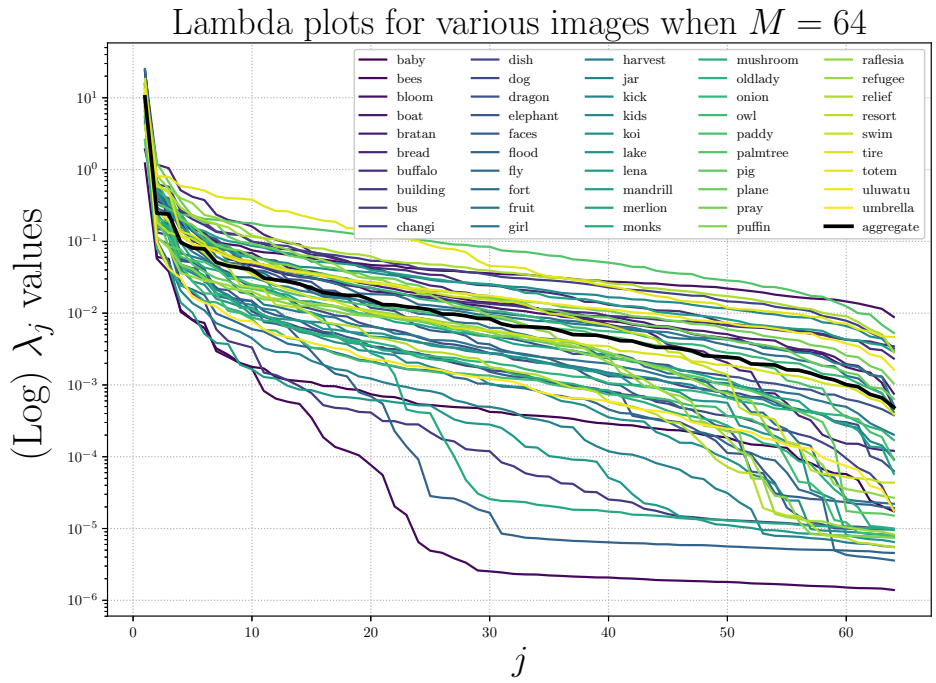


Figure 3. *The Λ profiles when $M = 64$. The vertical axis is labeled in logarithmic scale.*

Table 1
The First 8 Eigenvalues when $M = 64$.

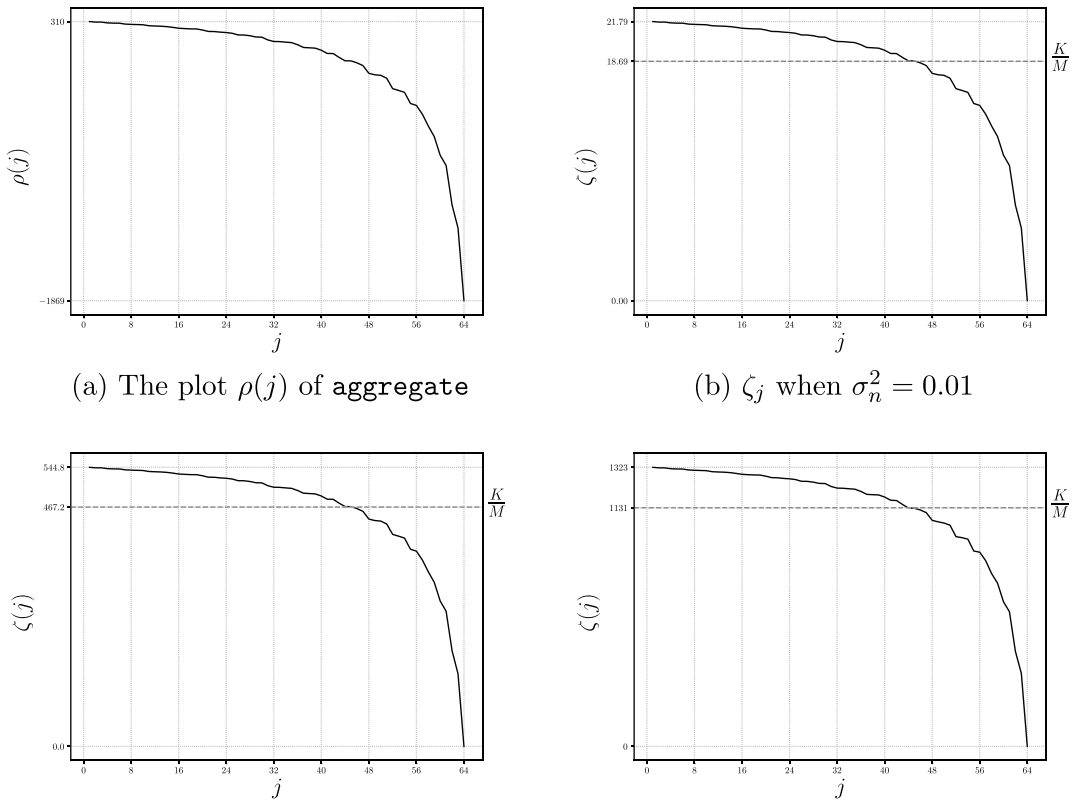
No.	Image	λ_1	λ_2	λ_3	λ_4	λ_5	λ_6	λ_7	λ_8
	`aggregate`	10.216	0.244	0.238	0.096	0.080	0.078	0.051	0.045
1	`fruit`	15.843	0.458	0.400	0.140	0.112	0.100	0.062	0.050
2	`merlion`	15.125	0.245	0.142	0.096	0.056	0.043	0.039	0.035
3	`dragon`	10.590	0.454	0.333	0.200	0.164	0.122	0.107	0.078
4	`flood`	8.714	0.216	0.167	0.047	0.042	0.041	0.017	0.016
5	`mandrill`	4.276	0.326	0.280	0.201	0.154	0.138	0.114	0.100
6	`oldlady`	2.578	0.080	0.043	0.025	0.020	0.014	0.014	0.012

in black. The aggregate λ plot is representative of the individual plots for the small indices $1, 2, \ldots, m$. Hence, it is reasonable to use the aggregate value in the design. The images in the data set have more varied individual λ_j values as j grows larger but this will not significantly influence the quality of recovery as $\sigma_n^2 > \lambda_j$ by then. Table 1 gives the first eight eigenvalues, rounded to three decimal places, of six images from Figure 1 and of `aggregate`.

The plot $\rho(j)$ for $j \in [\![64]\!]$ based on (3.5) using the λ_j values of `aggregate` is in Figure 4(a). If we are given just enough number K of probings to allocate such that $\zeta_j \geq 0$ for all $j \in [\![64]\!]$, as discussed in subsection 3.1, then the respective plots of ζ_j for $\sigma_n^2 \in \{0.01, 0.25, 0.64\}$ are those given in Figure 4(b)–(d). Each plot shows how the curve is influenced by $\frac{K}{M}$, which

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms



(a) The plot $\rho(j)$ of `aggregate`

(b) ζ_j when $\sigma_n^2 = 0.01$

(c) ζ_j when $\sigma_n^2 = 0.25$

(d) ζ_j when $\sigma_n^2 = 0.64$

Figure 4. *The plot of the function $\rho(j)$ and the respective distribution of ζ_j for $j \in [\![64]\!]$ based on the Λ profile of* `aggregate` *for the indicated σ_n^2.*

serves as the constant level around which each ζ_j lives, starting above it before decreasing as j goes to M. The shape is determined by $\sigma_n^2 \cdot \rho(j)$ since $\zeta_j = \frac{K}{M} + \sigma_n^2 \cdot \rho(j)$.

The distribution of probings. For the various levels of noise we next determine the projection packet designs to holographically sense the images in the data set, based on the learned second order statistics $\mathbf{R}_{xx}$. Since we aim to achieve the lowest MSE when all N packets, i.e., $K = N \cdot m$ probings, are available, we solve for ζ_j for $j \in [\![L]\!]$. Calibrations to conform to Modes 1 to 3 are subsequently carried out.

Table 2 presents the optimal sensing distributions obtained for the indicated noise levels, based on either the Λ of `aggregate` or the Λ of `fruit`, or that of `mandrill`, for illustrative comparison. The set $\{s_j : j \in [\![L]\!]\}$ is written in shorthand with $[x_1]^{y_1}[x_2]^{y_2} \dots [x_r]^{y_r}$ denoting $s_j = x_1$ for $j \in [\![y_1]\!]$ followed by $s_j = x_2$ for $j \in [\![y_1 + 1, y_1 + y_2]\!]$ and so on until $s_j = x_r$ for $j \in [\![L - y_r + 1, L]\!]$. We remove the superscript if $y_i = 1$. Notice that $L = \sum_{i=1}^{r} y_r$. For example, entry 3 in Table 2 for the `aggregate`'s statistics has the distribution $[8]^3[4][2]^2$ in Mode 1. We read this as $L = 6$ with $s_1 = s_2 = s_3 = 8$, $s_4 = 4$, and $s_5 = s_6 = 2$, when all N packets are available.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

Table 2

The assignments $s_1, \ldots, s_L$ based on the Λ profiles of `aggregate`*,* `fruit`*, and* `mandrill`*. For Mode 3, the distribution is always $[N]^m$, i.e., the first m indices are each probed N times.*

No.	(M, m, N)	σ_n^2	Distribution when using the statistics of `aggregate`	
			Mode 1	Mode 2
1	$(64, 4, 8)$	0.01	$[2]^4[1]^{24}$	$[8][7][1]^{17}$
2		0.25	$[7][6][5][4][3]^2[2][1]^2$	$[8]^2[5][3]^3[1]^2$
3		0.64	$[8]^3[4][2]^2$	$[8]^3[5][2][1]$
4		1.00	$[8]^4$	$[8]^4$
5	$(64, 4, 16)$	0.01	$[3]^2[2]^{23}[1]^{12}$	$[16][7][2]^{12}[1]^{17}$
6		0.25	$[10][9][8][7][6]^2[5][4]^2[3][1]^2$	$[16][8]^2[6]^3[4][3]^3[1]$
7		0.64	$[15][12]^2[8][7]^2[2][1]$	$[16][14][12][8][6]^2[2]$
8		1.00	$[16]^2[14][8][5]^2$	$[16]^2[15][7][5]^2$
9	$(64, 8, 8)$	0.01	$[3]^2[2]^{23}[1]^{12}$	$[8]^3[3][2]^{10}[1]^{17}$
10		0.25	$[8]^5[6][5][4]^2[3][1]^2$	$[8]^6[6][3]^3[1]$
11		0.64	$[8]^8$	$[8]^8$
12	$(64, 8, 16)$	0.01	$[4]^{12}[3]^{18}[2]^{10}[1]^6$	$[16][15][3]^{22}[2]^{12}[1]^7$
13		0.25	$[14][13]^2[11]^3[9][8]^2[7][6][5]^2[4][2][1]$	$[16]^3[11][10]^2[8]^2[7]^2[5]^2[4][3][2]$
14		0.64	$[16]^6[14][7][6][5]$	$[16]^6[14][7][6][5]$
15		1.00	$[16]^7[12][3][1]$	$[16]^7[13][2][1]$
No.	(M, m, N)	σ_n^2	Distribution for `fruit`	
			Mode 1	Mode 2
1	$(64, 4, 8)$	0.01	$[2]^6[1]^{20}$	$[8]^2[1]^{16}$
2		0.25	$[6][5]^2[4]^3[2][1]^2$	$[8][7][5][4][3]^2[1]^2$
3		0.64	$[8]^2[7][4][3][2]$	$[8]^3[4][2]^2$
4		1.00	$[8]^3[5][2][1]$	$[8]^3[7][1]$
5	$(64, 4, 16)$	0.01	$[3]^6[2]^{19}[1]^8$	$[16][5][2]^{15}[1]^{13}$
6		0.25	$[9][8]^2[7]^3[5][4]^2[3][2]$	$[14][8]^2[7][6]^2[4]^2[3][2]^2$
7		0.64	$[13][12]^2[9][8][7][3]$	$[16][14][11][8][7][6][2]$
8		1.00	$[16][13]^2[9][7][6]$	$[16]^2[13][8][6][5]$
9	$(64, 8, 8)$	0.01	$[3]^6[2]^{19}[1]^8$	$[8]^2[7][2]^{14}[1]^{13}$
10		0.25	$[8]^4[7]^2[5][4]^2[3][2]$	$[8]^6[5][4][3][2]^2$
11		0.64	$[8]^8$	$[8]^8$
12	$(64, 8, 16)$	0.01	$[4]^{21}[3]^{10}[2]^6[1]^2$	$[16][13][4]^7[3]^{17}[2]^8[1]^4$
13		0.25	$[13]^3[12][11]^2[9]^2[8][7]^2[5][4][3]^2$	$[16]^2[14][11]^2[10][9][8]^2[6]^2[4]^2[3][2]$
14		0.64	$[16]^6[13][7]^2[3][2]$	$[16]^6[14][7][6][3][2]$
15		1.00	$[16]^7[13][3]$	$[16]^7[13][3]$
No.	(M, m, N)	σ_n^2	Distribution for `mandrill`	
			Mode 1	Mode 2
1	$(64, 4, 8)$	0.01	$[1]^{32}$	$[8]^3[7][1]$
2		0.25	$[4]^3[3]^3[2]^3[1]^5$	$[8][7][3]^2[2]^3[1]^5$
3		0.64	$[7][6][5][4][3]^2[2][1]^2$	$[8]^2[5][4][3][2][1]^2$
4		1.00	$[8][7][6][5][3][2][1]$	$[8]^3[4][2]^2$
5	$(64, 4, 16)$	0.01	$[2]^{15}[1]^{34}$	$[16][11][1]^{37}$
6		0.25	$[6]^2[5]^4[4]^3[3]^4[2]^2[1]^4$	$[15][5]^3[4]^3[3]^4[2]^3[1]^4$
7		0.64	$[10][8]^2[7][6]^2[5][4][3]^2[2][1]^2$	$[16][8][7]^2[6][5][4][3]^2[2]^2[1]$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Table 2 (*cont.*)

8		1.00	$[13][10][9][8][7][6][4][3][2][1]^2$	$[16][10][9][8][6][5][4][3][2][1]$
9	$(64, 8, 8)$	0.01	$[2]^{15}[1]^{34}$	$[8]^3[5][1]^{35}$
10		0.25	$[6]^2[5]^4[4]^3[3]^4[2]^2[1]^4$	$[8]^3[6][4]^3[3]^4[2]^3[1]^4$
11		0.64	$[8]^4[7][6][5][4][3]^2[2][1]^2$	$[8]^6[5][3]^2[2]^2[1]$
12		1.00	$[8]^7[4][2][1]^2$	$[8]^7[5][2][1]$
13	$(64, 8, 16)$	0.01	$[3]^{19}[2]^{31}[1]^9$	$[16]^2[5][2]^{37}[1]^{17}$
14		0.25	$[9][8]^4[7]^3[6]^4[5]^2[4]^5[3]^2[2]^2[1]^2$	$[16][12][8][7]^3[6]^5[5]^3[4]^3[3]^2[2]^3[1]^2$
15		0.64	$[14][12]^2[11][10]^2[9][8]^2[7][6]^2[5][4][2]^2[1]^2$	$[16]^3[11][10][9][8][7]^2[6]^2[5][4]^2[1]^3$
16		1.00	$[16]^2[15][13][12][11][9][8][7][6]^2[4][3][2]$	$[16]^4[11][10][9][7]^2[6][5][4][3][2]$

Example 5.1. Let $(M, m, N) = (64, 8, 8)$. Using Mode 1 on the Λ profile of `aggregate`, we obtain the following packet sensing allocations for $\sigma_n^2 \in \{0.01, 0.25, 0.64\}$, where locations refer to the indices i where $\mathbf{P}_k$ has entry 1 on its diagonal:

k	Locations for $\sigma_n^2 = 0.01$	Locations for $\sigma_n^2 = 0.25$	Locations for $\sigma_n^2 = 0.64$
1	$1, 4, 8, 12, 16, 20, 24, 30$	$1, 2, 3, 4, 5, 6, 7, 9$	$1, 2, 3, 4, 5, 6, 7, 8$
2	$1, 4, 8, 12, 16, 20, 24, 31$	$1, 2, 3, 4, 5, 6, 7, 9$	$1, 2, 3, 4, 5, 6, 7, 8$
3	$1, 5, 9, 13, 17, 21, 25, 32$	$1, 2, 3, 4, 5, 6, 7, 9$	$1, 2, 3, 4, 5, 6, 7, 8$
4	$2, 5, 9, 13, 17, 21, 25, 33$	$1, 2, 3, 4, 5, 6, 8, 10$	$1, 2, 3, 4, 5, 6, 7, 8$
5	$2, 6, 10, 14, 18, 22, 26, 34$	$1, 2, 3, 4, 5, 6, 8, 10$	$1, 2, 3, 4, 5, 6, 7, 8$
6	$2, 6, 10, 14, 18, 22, 27, 35$	$1, 2, 3, 4, 5, 6, 8, 10$	$1, 2, 3, 4, 5, 6, 7, 8$
7	$3, 7, 11, 15, 19, 23, 28, 36$	$1, 2, 3, 4, 5, 7, 8, 11$	$1, 2, 3, 4, 5, 6, 7, 8$
8	$3, 7, 11, 15, 19, 23, 29, 37$	$1, 2, 3, 4, 5, 7, 9, 12$	$1, 2, 3, 4, 5, 6, 7, 8$

5.2. The sensing and recovery steps. Now that all of the ingredients to sense any input image are in place, we are ready to showcase the holographic image sensing in action. Several functionalities are included in the software for analytical purposes. Aside from allowing users to vary the basic parameters M, m, N, and σ_n^2, there are options to perform the sensing and complete the recovery process by using either the Λ of `aggregate` or that of the individual image. The modes can also be set as desired. Recall that we represent $\mathbf{y}_\omega$, instead of $\mathbf{x}_\omega$, holographically, as N packets of data $\mathbf{z}_k$ given by

$$\Big\{ \mathbf{z}_k = \mathbf{U}_k \Psi^\top \mathbf{x}_\omega + \mathbf{n}_k : k \in [\![N]\!] \Big\}.$$

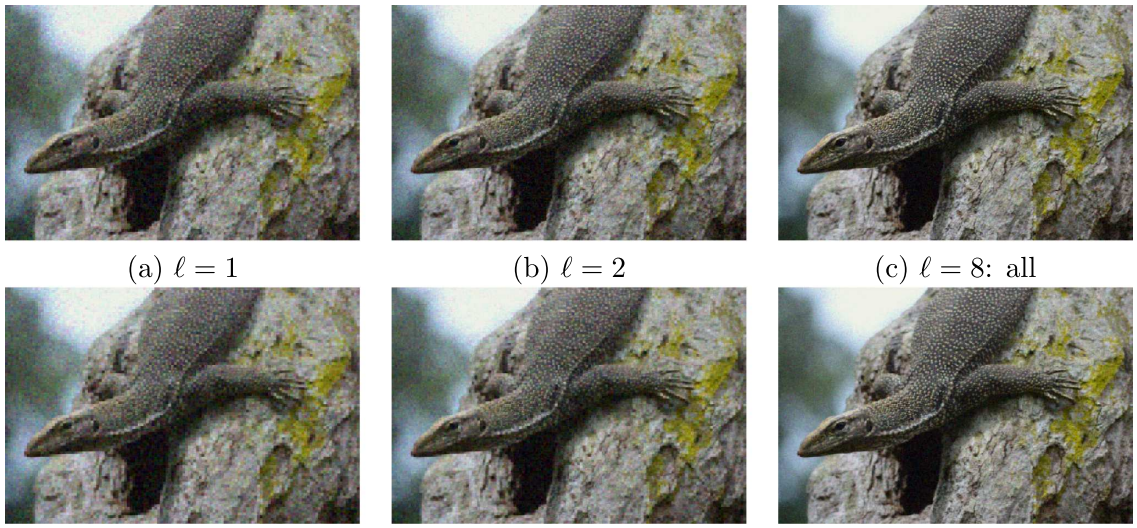
The packets can be stored for later recovery, once needed, or transmitted over some channels for reconstruction at another location.

In the recovery process, we assume that an arbitrary ℓ out of N packets have been made available. We have $\mathbf{U}_{\text{combi},\kappa} := (\mathbf{U}_{k_1} | \mathbf{U}_{k_2} | \ldots | \mathbf{U}_{k_\ell})$, where $\kappa := \{k_1, k_2, \ldots, k_\ell\}$ is the corresponding index set. The recovered estimate $\widehat{\mathbf{y}}_\omega$ of $\mathbf{y}_\omega$ is given by

$$\widehat{\mathbf{y}}_{\omega,\kappa} = \Lambda \; \mathbf{U}_{\text{combi},\kappa} \; \mathbf{M}^{-1} \; \mathbf{z}_{\text{combi},\kappa} \text{ where } \mathbf{M} := \mathbf{U}_{\text{combi},\kappa}^\top \; \Lambda \; \mathbf{U}_{\text{combi},\kappa} + \sigma_n^2 \mathbf{I}_{(\ell \cdot m)}. \tag{5.1}$$

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms



(a) $\ell = 1$ (b) $\ell = 2$ (c) $\ell = 8$: all

(d) $\ell = 1$ (e) $\ell = 2$ (f) $\ell = 8$: all

Figure 5. *Recovered* `dragon` *in the randomized procedure with* $(M, m, N, \sigma_n^2) = (64, 8, 8, 0.64)$ *on Mode* 1. *Recovery for the images in the upper row uses the* Λ *of* `dragon`*, with distribution* $[8]^7[5][2][1]$*, while for those in the lower row it uses the* Λ *of* `aggregate`*, with distribution* $[8]^8$.

The original vector $\mathbf{x}_\omega$ is thus estimated by $\widehat{\mathbf{x}}_{\omega,K} := \Psi\widehat{\mathbf{y}}_{\omega,K}$. To recover the whole image we assume that the location of each patch is known, i.e., the index set $\{\omega\}$ is known and in order. Reversing the prosess described at the start of subsection 5.1 yields the recovered image. We make available two procedures.

Incremental: Begin by randomly choosing any 1 out of the N packets and perform the image reconstruction. At each ℓ, as ℓ goes from 2 to N, choose 1 packet at random from among the remaining $N - (\ell - 1)$ packets. Combine the packet with the existing $\mathbf{z}_{\text{combi}}$ and use the result to output an improved image. In this build-up, once a packet had been chosen it remains in use for an image reconstruction as ℓ increases.

Randomized: As ℓ goes from 1 to N, randomly choose any one of the $\binom{N}{\ell}$ possible index sets κ of size ℓ. Use the corresponding $\mathbf{z}_{\text{combi},\kappa}$ in the image reconstruction. Here, a packet which has been included earlier may be dropped in the next iteration.

Figure 5 presents recovered `dragon` images using a run of the randomized procedure on Mode 1 for the supplied input parameters. For a fixed ℓ, we can see how the resulting images are very similar on recovery based on the two Λ profiles, one belonging to `dragon` itself while the other is that of `aggregate`. As the noise level rises, L tends to decrease, favoring the probings of the first few coordinates where the corresponding λ values are higher. When $(M, m, N, \sigma_n^2) = (64, 8, 8, 1.00)$, for example, all three modes based on the Λ profile of `aggregate` coincide, with sensing distribution $[8]^8$. Figure 6 presents incrementally recovered `dragon` images, on the specified input parameters, for $\ell \in \{1, 2, 8\}$.

5.3. Performance analysis.

To analyze the recovery performance, one can consider several types of MSE computations in each of the procedures. Let the parameters M, m, N, and σ_n^2 as well as the mode be given for a chosen image. Hence, we have the Λ and Ψ matrices as well

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

(a) $\ell = 1$ (b) $\ell = 2$ (d) $\ell = 8$: all

Figure 6. *Incrementally recovered* `dragon` *using the* Λ *of* `aggregate` *with* $(M, m, N, \sigma_n^2) = (64, 8, 8, 1.00)$. *The noise forces all three modes to use the distribution* $[8]^8$, *i.e.,* $s_j = 8$ *for* $j \in [\![8]\!]$ *and* $s_j = 0$ *for* $j > 8$.

as the probing distribution. First, there is the theoretical best $\mathrm{MSE}_{\mathrm{best},\ell}$ from (3.12) to be used as a benchmark. Based on this equation, for a fixed mode, one can compute for the best case MSE_ℓ by replacing each ζ_j by the corresponding s_j. Second, for each ℓ, one can compute the expected value

$$\mathbb{E}(\mathrm{MSE}, \ell) := \frac{1}{\binom{N}{\ell}} \sum_{i=1}^{\binom{N}{\ell}} \left(\frac{1}{|\Omega|} \sum_{\omega=1}^{|\Omega|} (\mathbf{x}_\omega - \widehat{\mathbf{x}}_{\omega,i})^2 \right) \tag{5.2}$$

with

$$(\mathbf{x}_\omega - \widehat{\mathbf{x}}_{\omega,i})^2 = (\mathbf{x}_\omega - \widehat{\mathbf{x}}_{\omega,i})^\top (\mathbf{x}_\omega - \widehat{\mathbf{x}}_{\omega,i}) .$$

The second index i of $\widehat{\mathbf{x}}_{\omega,i}$ refers to the ith recovery simulation. For each ℓ we compute the average MSE, taken over $\min(100, \binom{N}{\ell})$ simulations, since averaging over $\binom{N}{\ell}$ simulations may not be practical as N grows.

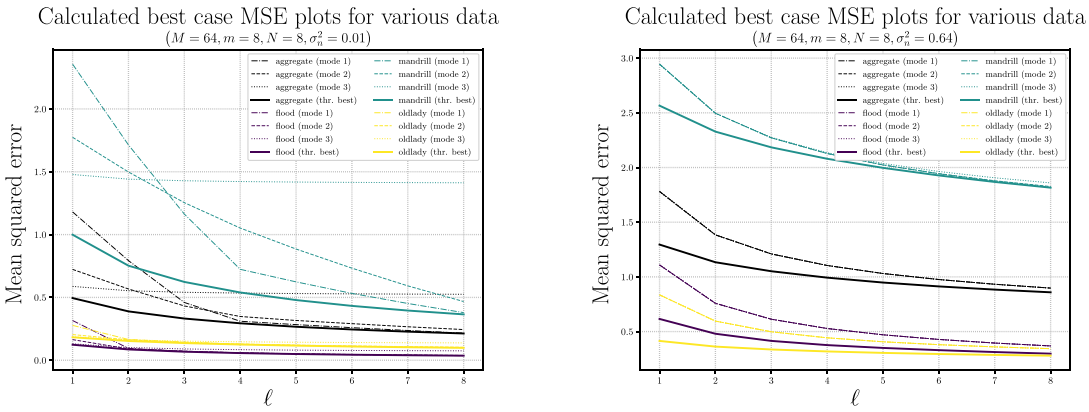
Figure 7 shows how the best case MSE curves of the three modes compare with respect to the theoretical best. Mode 1 comes closest to the theoretical ideal at $\ell = N$, as per the design. When ℓ is very small, however, its best case MSE is considerably higher than those of the other two modes. Mode 2 has been designed to improve the performance for such ℓ at the relatively small cost of decreased performance in the second half of ℓ values compared with Mode 1. Mode 3 has the best performance at $\ell = 1$ but then does not improve as much as the other two modes as more packets become available.

As we switch from Mode 1 to 2 and then to 3, the MSE variances among the simulations for small ℓ decreases, i.e., the recovery is smoother. There is less chance of recovering the image badly in the early stages of any single simulation. On the other hand, the gain in MSE reduction as ℓ goes toward N also lessens. Higher noise forces the plots of the three modes closer together, often to the point of merging into one plot. This is because the sensing mechanism assigns more probings to the portion of $\mathbf{y}_\omega$ with larger covariances to mitigate the effect of higher noise.

For an easier inspection on how a change in noise level affects the performance in an actual recovery, Figure 8 illustrates the trend on `dragon` for $(M, m, N) = (64, 8, 8)$ and $\sigma_n^2 \in \{0.01, 0.64\}$. For each mode, we run a single recovery procedure based on the Λ of `dragon` and the Λ of `aggregate`. Relevant plots for other images and parameter combinations can be similarly generated.

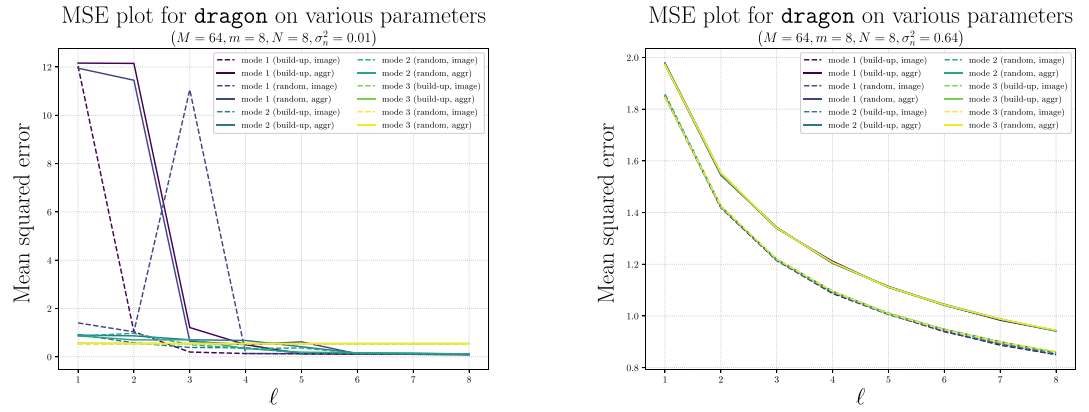
Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms



(a) MSE when $\sigma_n^2 = 0.01$ (b) MSE when $\sigma_n^2 = 0.64$

Figure 7. *A comparison of the plots for the theoretical best (thr. best), computed based on ζ_j for $j \in [\![M]\!]$, for $\ell \in [\![N]\!]$, in (3.12), with the computed best case* MSE *in the three modes, when ζ_j is replaced by the respective s_j. We use $(M, m, N) = (64, 8, 8)$ and $\sigma_n^2 \in \{0.01, 0.64\}$.*



(a) A `dragon` recovery at $\sigma_n^2 = 0.01$ (b) A `dragon` recovery at $\sigma_n^2 = 0.64$

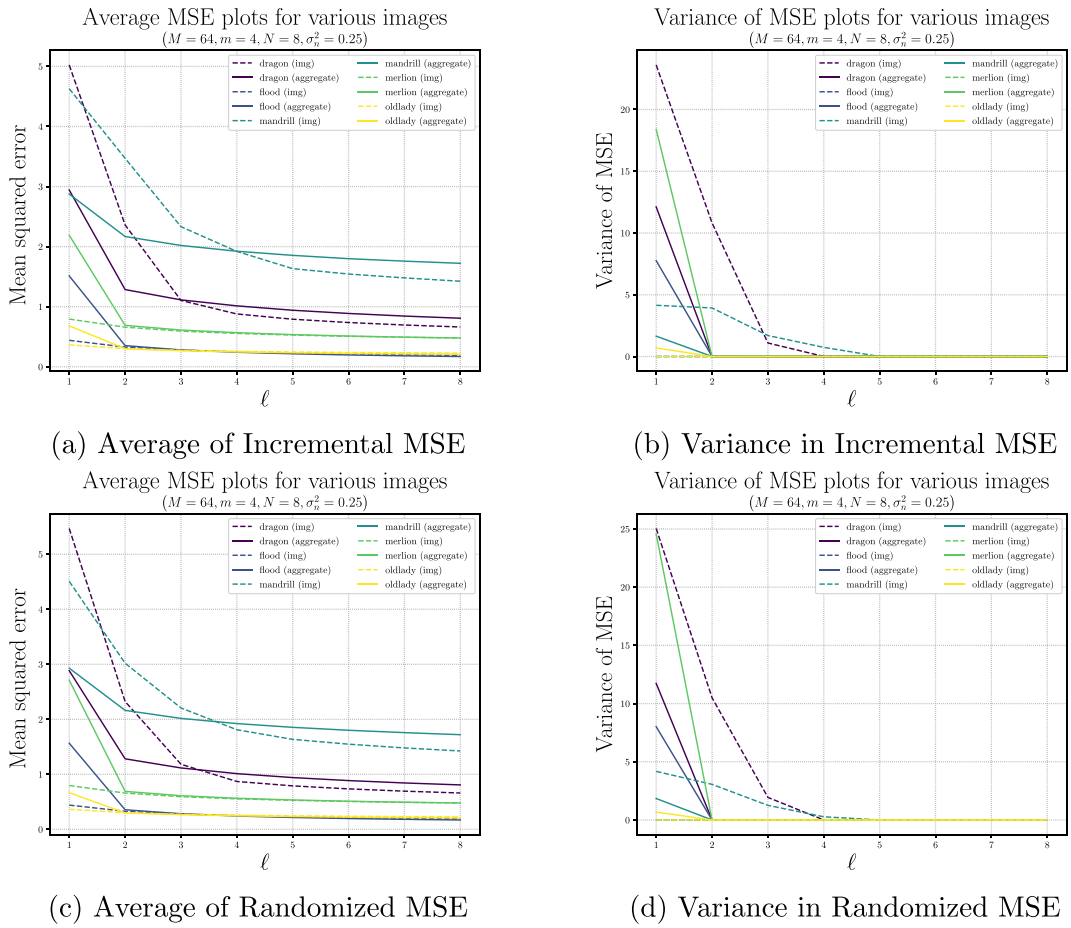
Figure 8. *The trend from a single instance of recovery for each of the indicated setting choices on* `dragon` *for $(M, m, N) = (64, 8, 8)$ and $\sigma_n^2 \in \{0.01, 0.64\}$.*

There are insights to gain from the average recovery performance. While Mode 1 is the closest to the original design philosophy of reaching the best MSE when all N packets are available, its recovery performance fluctuates rather widely for $\ell < \frac{N}{2}$. Figure 9 presents the average MSE and its variance, simulated on the indicated images, for both the incremental and randomized procedures when $(M, m, N, \sigma_n^2) = (64, 4, 8, 0.25)$.

6. Recovery and performance analysis on control images. In https://github.com/adamasstokhorst/holographic/tree/master/control we collect some natural images for control purposes. They are sensed and recovered by using the Λ of `aggregate` obtained in section 5, since their second order statistics are assumed to be unknown. We use two control images, namely `twobirds` and `boy`, in Figure 10 to illustrate and analyze their recovery. Their respective incrementally recovered images, for the indicated parameters, can be found in Figure 11.

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

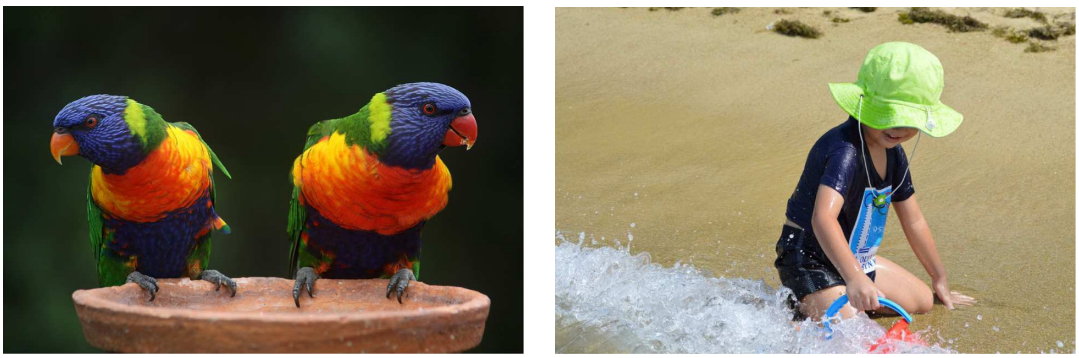
Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms



(a) Average of Incremental MSE (b) Variance in Incremental MSE

(c) Average of Randomized MSE (d) Variance in Randomized MSE

Figure 9. *The average* MSE *plots and their respective variances obtained from recovery simulations based on two options of second order statistics, namely the* `aggregate` *(aggregate) and the individual image (img). The plots in the upper row are based on the incremental recovery while those in the lower row come from the randomized one. We use Mode* 1 *with* $(M, m, N, \sigma_n^2) = (64, 4, 8, 0.25)$.

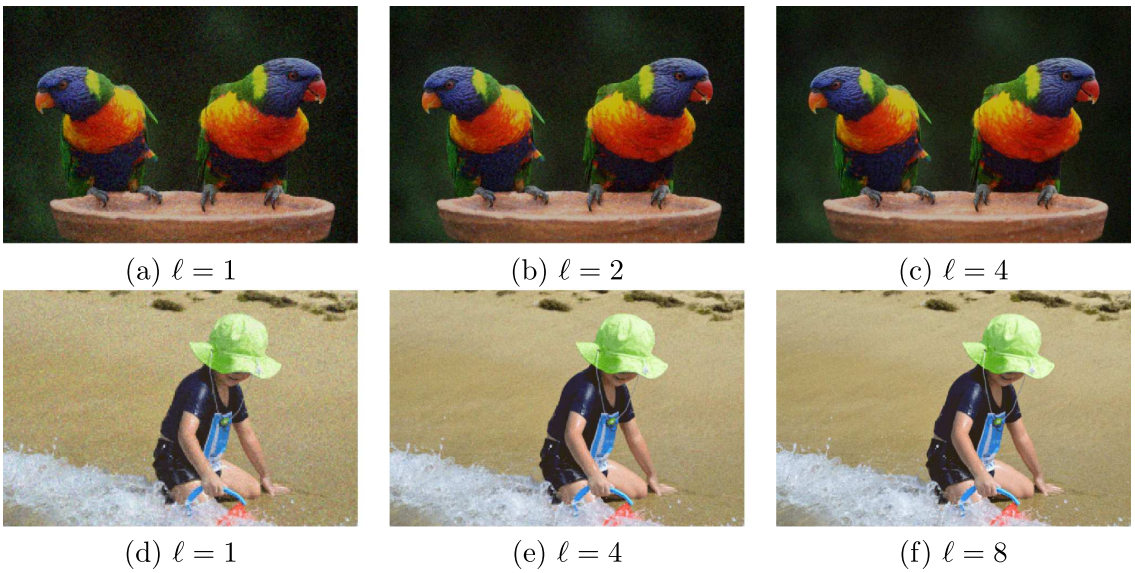


(a) `twobirds` (b) `boy`

Figure 10. *Two images from the control data set for illustrations in recovery and analysis. Taken from https://pixabay.com/users/magee-830963/,* `twobirds` *depicts two rainbow lorikeets. The image* `boy` *was taken by the corresponding author.*

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

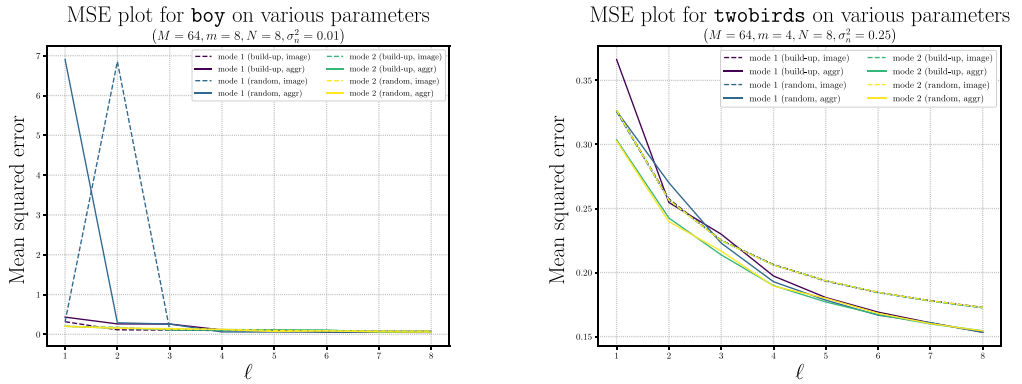
Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.



(a) $\ell = 1$ (b) $\ell = 2$ (c) $\ell = 4$

(d) $\ell = 1$ (e) $\ell = 4$ (f) $\ell = 8$

Figure 11. *Incrementally recovered* `twobirds` *and* `boy`. *On* `twobirds`, $(M, m, N, \sigma_n^2) = (64, 8, 8, 0.64)$. *By Table 2, entry 12, all modes have probing distribution* $[8]^8$. *Mode 2 with* $(M, m, N, \sigma_n^2) = (64, 4, 16, 0.64)$ *is used to recover* `boy`, *where the probing distribution is as given in Table 2, entry 7.*



(a) Recovery instances on `boy` (b) Recovery instances on `twobirds`

Figure 12. *The* MSE *plots capture the performance of a single recovery run each on Modes* 1 *and* 2, *for the indicated settings, on* `boy` *and* `twobirds`. *We use both the* Λ *profile of* `aggregate` *(aggr) and that of the image, as well as both the incremental (build-up) and randomized (random) procedure.*

We see in Figure 12 how the recovered images exhibit the expected properties of progressive refinement as more and more packets are used. The plots confirm that using the Λ of `aggregate` on control images results in very similar MSE values as using the individual image's profile, had it been known and used.

7. Concluding remarks.

We briefly reviewed the theory of holographic sensing and then explained in detail the process of designing the sensing packets before presenting the expected performance of the distributed holographic encodings of databases of images. We have outlined

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

the full design of the system implementing the idea of holographic representations and tested the system's performance on a data set of 49 natural images. This data set can be easily enlarged into a big database if desired, and similarly with the types of images. The system has been demonstrated to also perform well on randomly selected control images not present in the data set.

In conclusion we also wish to mention two distance-based correlation models that work quite well in estimating $\mathbf{R}_{xx}$ for natural images. We call them the `grid` model, when the distance is the Manhattan length, and the `line` model, when the ℓ_2-norm is used. The respective models can be formulated as follows. Let $\mathbf{R}^{\mathrm{G}}$ and $\mathbf{R}^{\mathrm{L}}$ be the respective $M \times M$ matrices that approximate $\mathbf{R}_{xx}$ in the `grid` and `line` models. Without loss of generality, we can assume M to be a square, e.g., 8^2 as in our typical implementation, and let $\nu := \sqrt{M}$.

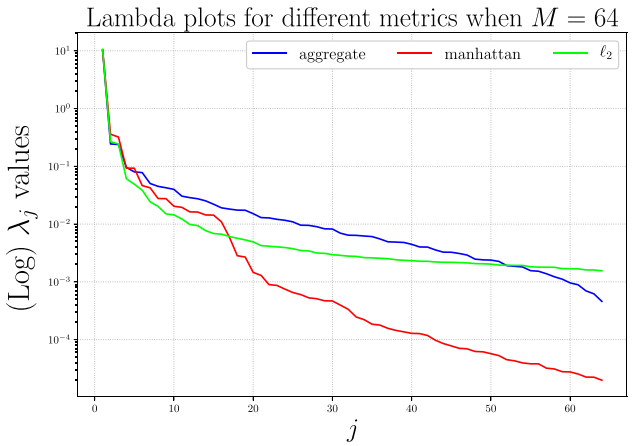
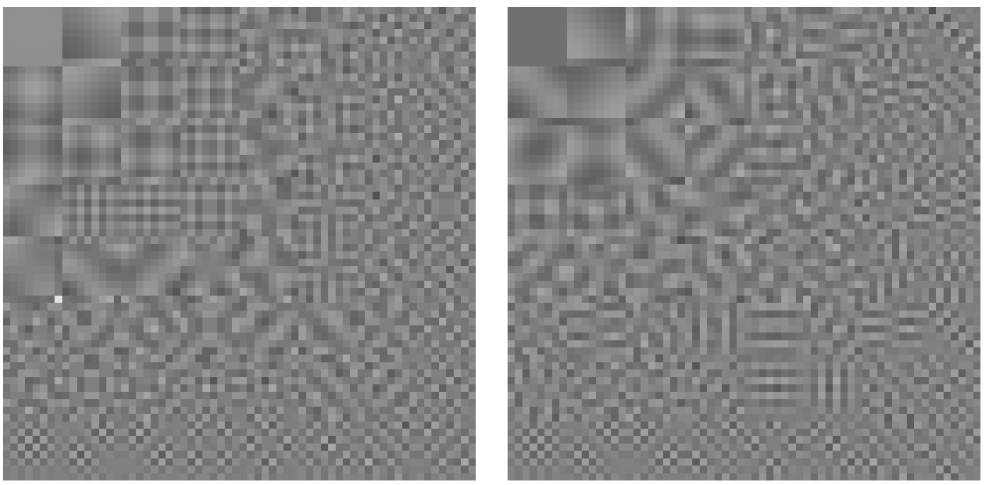


Figure 13. *Comparison of the* Λ *profiles of* `aggregate` *and of the two models for* $M = 64$.



(a) Ψ in `grid` model (b) Ψ in `line` model

Figure 14. *The visualized* Ψ *matrices to be compared with the one for* `aggregate` *in Figure 2.*

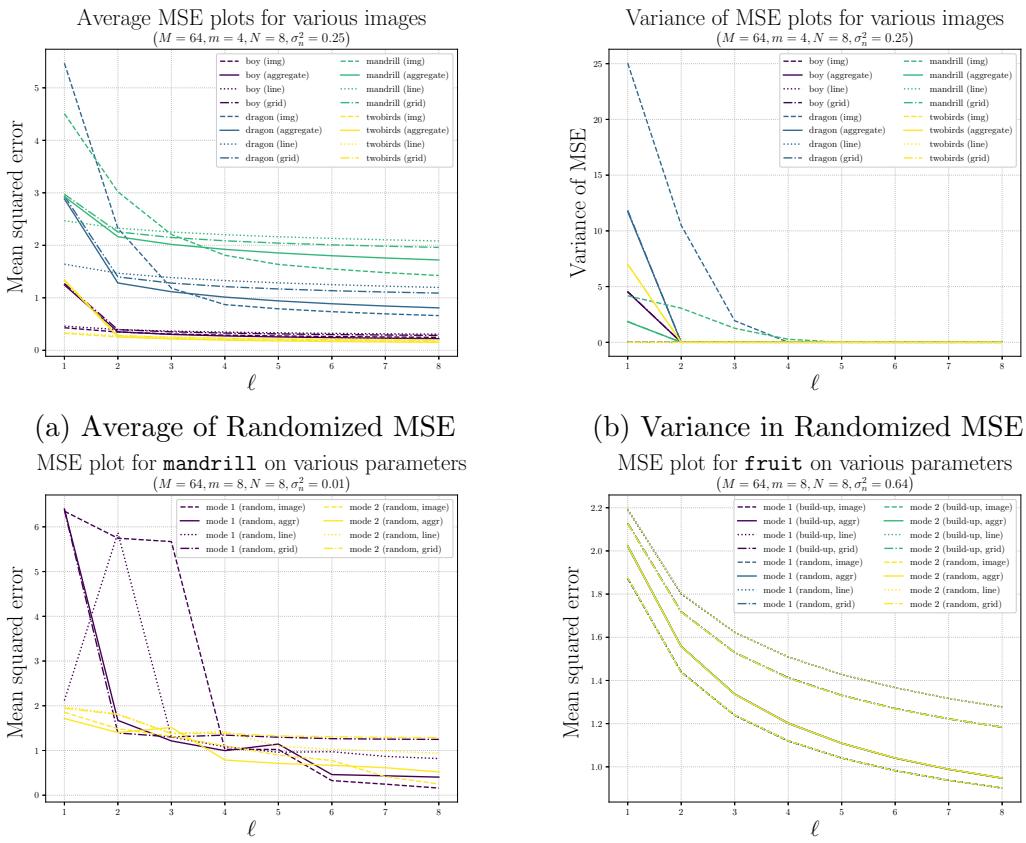
Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

Since correlation decays as a function of distance, the entries in the model matrices must faithfully reflect this fact. Let $i := \nu \cdot a + b$ and $j := \nu \cdot p + q$. For $i, j \in [\![M]\!]$, let

$$\mathbf{R}^{\mathrm{G}}_{i,j} = A \cdot \gamma^{|a-p|+|b-q|} \text{ and } \mathbf{R}^{\mathrm{L}}_{i,j} = A \cdot \gamma^{\sqrt{(a-p)^2+(b-q)^2}}. \tag{7.1}$$

One can then proceed to determine the values of the constants A and γ that make the respective matrices closely resemble some experimentally obtained entries of $\mathbf{R}_{xx}$ for a class of images. For $M = 64$, based on our input data set of images, we obtain $A = 0.18$ and $\gamma = 0.98$. The Λ profiles as well as the corresponding Ψ matrices can then be computed as above. Figures 13 and 14 provide, respectively, a useful Λ comparison plot and a visualization of each of the Ψ matrices for inspection. They show that the `line` model fits the values from our image data set better than the `grid` model.



(a) Average of Randomized MSE

(b) Variance in Randomized MSE

(c) Performance on `mandrill`

(d) Performance on `fruit`

Figure 15. *The first row displays the* average MSE *plots for the* randomized *recovery simulations on* `boy`, `dragon`, `mandrill`, *and* `twobirds` *on Mode* 1 *with* $(M, m, N, \sigma_n^2) = (64, 4, 8, 0.25)$. *The second row displays the performance on a single recovery instance for the indicated settings on* `mandrill` *and* `fruit`. *Notice that the captured randomized recovery instance on Mode* 1 *based on the* Λ *of* `mandrill` *is particularly bad. While such an instance is unlikely to occur, it highlights the potential drawback of using the randomized procedure in Mode* 1 *on some images. Taken together, the plots allow for a quick comparison on the recovery performance across the statistical properties of a specific image, of* `aggregate`, *and in the* `grid` *and* `line` *models, for various settings.*

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.

Using a reasonably fitting model may remove the need to compute for the second order statistics from some database prior to deployment. Figure 15 shows that there is indeed not much of a difference in the recovery performance, either on a single recovery run or on average.

Acknowledgments. We thank Professor Vivek Goyal and the referees for their valuable comments and suggestions that helped us improve the overall presentation of this paper.

REFERENCES

[1] *Matplotlib*, https://github.com/matplotlib/matplotlib/tree/v2.2.4.
[2] *NumPy*, https://numpy.org/.
[3] N. Ahmed, T. Natarajan, and K. Rao, *Discrete cosine transform*, IEEE Trans. Comput., C-23 (1974), pp. 90–93.
[4] A. Bruckstein, M. Ezerman, A. Fahreza, and S. Ling, *Holographic sensing*, Appl. Comput. Harmon. Anal., 49 (2020), pp. 296–315.
[5] A. M. Bruckstein, R. J. Holt, and A. N. Netravali, *Low discrepancy holographic image sampling*, Int. J. Imaging Syst. Technol., 15 (2005), pp. 155–167.
[6] A. M. Bruckstein, R. J. Holt, and A. N. Netravali, *Holographic representations of images*, IEEE Trans. Image Process., 7 (1998), pp. 1583–1597.
[7] A. M. Bruckstein, R. J. Holt, and A. N. Netravali, *On holographic transform compression of images*, Int. J. Imaging Syst. Technol., 11 (2000), pp. 292–314.
[8] Y. Dar and A. M. Bruckstein, *Benefiting from duplicates of compressed data: Shift-based holographic compression of images*, J. Math. Imaging Vis. (2021), https://doi.org/10.1007/s10851-020-01003-1.
[9] S. Dolev and S. Frenkel, *Multiplication free holographic coding*, in proceedings of the IEEE 26th Convention of Electrical and Electronics Engineers in Israel, 2010, pp. 146–150.
[10] S. Dolev, S. L. Frenkel, and A. Cohen, *Holographic coding by Walsh-Hadamard transformation of randomized and permuted data*, Inform. Appl., 6 (2012), pp. 76–83.
[11] R. Dovgard, *Holographic image representation with reduced aliasing and noise effects*, IEEE Trans. Image Process., 13 (2004), pp. 867–872.
[12] V. K. Goyal, J. Kovačević, R. Arean, and M. Vetterli, *Multiple description transform coding of images*, in Proceedings of ICIP98, Vol. 1, 1998, pp. 674–678.
[13] V. K. Goyal, J. Kovačević, and J. A. Kelner, *Quantized frame expansions with erasures*, Appl. Comput. Harmon. Anal., 10 (2001), pp. 203–233.
[14] J. D. Hunter, *Matplotlib: A 2D graphics environment*, Comput. Sci. Eng., 9 (2007), pp. 90–95.
[15] G. Kutyniok, A. Pezeshki, R. Calderbank, and T. Liu, *Robust dimension reduction, fusion frames, and Grassmannian packings*, Appl. Comput. Harmon. Anal., 26 (2009), pp. 64–76.
[16] F. Lundh, A. Clark, and Contributors, *Pillow, the Friendly PIL Fork*, https://python-pillow.org/.
[17] T. E. Oliphant, *A Guide to NumPy*, Vol. 1, Trelgol Publishing USA, 2006.
[18] S. Pradhan, R. Puri, and K. Ramchandran, *n-Channel symmetric multiple descriptions—Part* I*: (n, k) source-channel erasure codes*, IEEE Trans. Inform. Theory, 50 (2004), pp. 47–61.
[19] R. Puri, S. Pradhan, and K. Ramchandran, *n-Channel symmetric multiple descriptions—Part* II*: An achievable rate-distortion region*, IEEE Trans. Inform. Theory, 51 (2005), pp. 1377–1392.
[20] S. D. Servetto, K. Ramchandran, V. A. Vaishampayan, and K. Nahrstedt, *Multiple description wavelet based image coding*, IEEE Trans. Image Process., 9 (2000), pp. 813–826.
[21] V. Vaishampayan, *Design of multiple description scalar quantizers*, IEEE Trans. Inform. Theory, 39 (1993), pp. 821–834.
[22] R. Venkataramani, G. Kramer, and V. Goyal, *Multiple description coding with many channels*, IEEE Trans. Inform. Theory, 49 (2003), pp. 2106–2114.

Downloaded 02/21/21 to 3.1.58.13. Redistribution subject to SIAM license or copyright; see https://epubs.siam.org/page/terms

Copyright © by SIAM. Unauthorized reproduction of this article is prohibited.