VERSATILE AND RESILIENT HOLOGRAPHIC SENSING ON IMAGES

Alfred Marcel Bruckstein*†  Martianus Frederic Ezerman†  Adamas Aqsa Fahreza†  San Ling†

* Department of Computer Science, Technion, Israel Institute of Technology, Haifa, Israel.
† School of Physical and Mathematical Sciences, Nanyang Technological University, Singapore.

ABSTRACT

Holographic representations of data have two main objectives. The sensing process generates and then distributes multiple descriptions of information in packets that enable progressive recovery. The packets are designed to have as equal importance as possible to guarantee smoothness in the quality of the recovered information, independent of the order of the packet’s arrival.

We recently developed a least-squares approach to the design of holographic representation for stochastic data vectors. While it relies on the framework widely used in modeling images, it had not been rigorously tested on actual images. This paper describes the results of such tests on various images under noisy sensing environment. We report that holographic sensing is indeed versatile and resilient in actual deployment.

Index Terms— holographic sensing, mean-squared error estimation, representation of images.

1. INTRODUCTION

Compression is a recurrent theme in image processing. Professional photographers, for example, prefer to keep raw images for their quality. When sent over networks or digitally stored, they are typically compressed in jpeg format [1]. There are known approaches in describing data in reduced dimensions or in sending them over various channels with an assurance of recovery within some threshold. Examples include compressive sensing, whose theoretical foundation was established in [2] and [3], and multiple description coding as explained in [4]. Prior to our recent proposal in [5], there have been methods for holographic representations of signals in [6].

This paper describes our implementation of holographic sensing as developed in [5] on images, including natural images taken with photographic effects, paintings and drawings. We highlight the versatility and resilience of the approach.

In terms of compression rates, it is easy to adjust the system parameters to suit implementation constraints and fluctuation in available resources. Unlike in jpeg where having multiple packets of the same image does not lead to gradual improvement, multiple data packets of an image, holographically sensed, always lead to improvement, regardless of their order of arrival. The approach is resilient. It comfortably handles dynamically changing noise while maintaining excellent denoising ability.

For uniformity, we retain the notations in [5]. Given integers 1 ≤ k < N, [k] denotes {1, 2, . . . , k} and [k, N] denotes {k, k + 1, . . . , N}. Vectors are columns denoted by bold lowercase letters. Bold uppercase letters or upper Greek symbols are for matrices. A diagonal matrix with diagonal entries v j : j ∈ [n] is often written as diag(v1, v2, . . . , vn). I n is the identity matrix. Concatenation of vectors or matrices is signified by | between the components. The transpose and the conjugate transpose of A are, respectively, A⊤ and A†.

The target images used here are in Figure 1. They are either copyright-free or licensed under conditions that allow for sharing and adapting. Our simulation routines are in python 2.7, incorporating image manipulation tools from the PIL 1.1.7 library [7]. The numpy and matplotlib packages assist matrix computations and graph plottings.

The image is partitioned into 8 × 8 blocks, with paddings of 0s as needed. The colour values are encoded in range 0 to 255. Let Ω be some index set. The blocks forming the entire image are in {xω : ω ∈ Ω} with |Ω| being the number of blocks. We break this image representation into N packets of data. Each of the packets stores m-dimensional vectors, also indexed by Ω. Actual sensing is necessarily noisy. The noise nω is assumed to be i.i.d Gaussian with zero mean and variance σ2 n. For example, there are 2 to 3 errors per 256-pixel representation of xω for each ω ∈ Ω when σ2 n = 0.01 and the error count doubles for σ2 n = 0.02. In short, we are in the unaligned case as discussed in [5] Section 4.

2. THE TASK FLOW

The first step is to compute the autocovariance matrix

$$R_{xx} = \frac{1}{|Ω|} \sum_{\omega \in Ω} x_\omega x_\omega^\dagger. \quad (1)$$
We then use singular value decomposition (SVD), that sends $R_{xx} \mapsto \Psi \Lambda \Psi^\dagger$, to obtain the unitary matrix $\Psi$ and the diagonal matrix $\Lambda$ with positive entries along its diagonal. To define the sensing operators it is crucial to have $\Psi$. The diagonal entries in $\Lambda = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{64})$ are needed to determine the subspace arrangements that yield the best MSE value. We collect the $\Lambda$ profiles, in their respective logarithmic values for ease of reading, in Figure 2. The aggregate profile is the average of the lambda profiles of many images, including the six explicitly listed above, and is useful as a control value.

Now that we have the $\Lambda$ profile, the next step is to find the best arrangements of subspaces. We want to generate packets of sensed data that minimize the expected mean-squared error when all $N$ packets are present for recovery. More concretely, we define the sensing operators.

The orthonormal operator $P_k$ projects 64-dimensional vectors onto $m$-dimensional subspaces. Let $U_k$ be a $64 \times m$ matrix whose columns form an orthonormal basis for $P_k$ for $k \in [N]$. Note that $P_k = U_k U_k^\dagger$ and $U_k \Psi \omega$ is a vector of $m$ entries. It displays the coefficients of the representation of $P_k \Psi \omega$ in the basis represented in $U_k$, making $P_k \Psi \omega = U_k (U_k^\dagger \Psi \omega)$. Since we are in the unaligned case, instead of $\Psi \omega$, we sense $y_\omega := \Psi \omega$. For any $\omega \in \Omega$ and any $k \in [N]$ the data packets form the set $\{ z_k := U_k \Psi \omega + n_\omega \}$. For a nonempty $K := \{ k_1, k_2, \ldots, k_t \} \subseteq [N]$ the combined measurement from the $\ell$ packets is an $(\ell \cdot m) \times 1$ vector

$$z_{\text{combi.}K} = \begin{pmatrix} z_{k_1} \\ \vdots \\ z_{k_t} \end{pmatrix}. $$

The combined projection matrix $P_K = \sum_{j=1}^t P_{k_j}$ is diagonal with nonnegative integer entries $s_j$ along the diagonal. Let $P_j$ denote the set of positions in the diagonal of $P_K$ with common entry $s$. The number of times that the $j$-th coordinate of $y_\omega$ is measured is $s_j$. Hence, $\sum_{j=1}^{64} s_j = m \times N$. We know from [5] Equation (10) that

$$\text{MSE}(\Lambda, \sigma_n^2, N) = \sum_{s=0}^N \sum_{j \in P_j} \lambda_s \sigma_n^2 + 8 \lambda_j = \sum_{j=1}^M \lambda_j \sigma_n^2 + s_j \lambda_j. $$

Now we follow the steps given in [5] Case 2, Section 3] to compute the values of $s_j$s that minimize $\text{MSE}(\Lambda, \sigma_n^2, N)$ based on each image’s $\Lambda$ profile. One must ensure that the rounding off is done properly and $s_j \leq N$ for all $j$. Table 1 gives some computed values that will be used in the sequel.

Table 1. Computed values of $s_j$ for illustrative purposes.

<table>
<thead>
<tr>
<th>No.</th>
<th>Image</th>
<th>$(m, N, \sigma_n^2)$</th>
<th>$t$</th>
<th>${ s_j : j \in [t] }$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ant</td>
<td>(4.8, 0.02)</td>
<td>9</td>
<td>$[6 \quad 5^2 \quad 4^2 \quad 3^2 \quad 2^4]_1^t$</td>
</tr>
<tr>
<td>2</td>
<td>fly</td>
<td>(4.8, 0.02)</td>
<td>6</td>
<td>$[7^3 \quad 4^2 \quad 3]_1^t$</td>
</tr>
<tr>
<td>3</td>
<td>abstract</td>
<td>(4.4, 0.01)</td>
<td>13</td>
<td>$[2^6 \quad 3^2 \quad 110]_1^t$</td>
</tr>
<tr>
<td>4</td>
<td>(4, 0, 01)</td>
<td>19</td>
<td>$[2^3 \quad 13 \quad 1]_1^t$</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>face</td>
<td>(4, 0, 01)</td>
<td>32</td>
<td>$[1^32]_1^t$</td>
</tr>
<tr>
<td>6</td>
<td>rain</td>
<td>(4, 0, 03)</td>
<td>26</td>
<td>$[2^6 \quad 1^20]_1^t$</td>
</tr>
<tr>
<td>7</td>
<td>(4, 0, 01)</td>
<td>16</td>
<td>$[1^16]_1^t$</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>(4, 10, 02)</td>
<td>40</td>
<td>$[1^40]_1^t$</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>(4, 0, 01)</td>
<td>9</td>
<td>$[5^4 \quad 4^2 \quad 3 \quad 2^2 \quad 1^1]_1^t$</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>aggregate</td>
<td>(4, 0, 01)</td>
<td>12</td>
<td>$[4^2 \quad 3^3 \quad 2^1 \quad 1^3]_1^t$</td>
</tr>
</tbody>
</table>

Fig. 2. The $\Lambda$ Profiles.

Table 1 gives some computed values that will be used in the sequel. Here is how to read Table 1 given $m, N$ and $\sigma_n^2$. We use $t$ to mark the threshold where $s_j = 0$ for all $j > t$. In Entry 2, for example, $[7^3 \quad 4^2 \quad 3]$ means that to obtain the best MSE upon recovery of fly from all 8 packets combined, the first three coordinates of $y_\omega$ must be measured 7 times, the fourth
and fifth coordinates 4 times each, the sixth 3 times while the rest of the coordinates can be safely ignored. Using the $\mathcal{P}$ notation, we have $\mathcal{P}_1 = [7, 64]$, $\mathcal{P}_2 = \{1, 2, 3\}$, $\mathcal{P}_3 = \{4, 5\}$, and $\mathcal{P}_3 = \{6\}$. The other entries can be similarly interpreted.

Notice how the $\Lambda$ profiles affect the set $\{s_j : j \in [64]\}$. As illustrated for abstract, face and ark, the system adjusts the arrangement of the subspaces as the parameters change. Once the arrangement of the subspaces leading to the best MSE when all $N$ packets are available has been determined, we proceed, for each $K$ with $|K| = \ell$, to compute for $\text{MSE}(\Lambda, \sigma_n^2, \ell)$ by using \[\text{MSE}(\Lambda, \sigma_n^2, \ell) = \sum_{j=1}^{M} \lambda_j - \sum_{s=1}^{\ell} \sum_{j \in \mathcal{P}_s} \frac{s\lambda_j^2}{\sigma_n^2 + s\lambda_j}.\] (2)

Now we have all the ingredients to recover the image and analyse the recovery performance. Remember that we holographically represent $\mathbf{y}_\omega$ instead of $\mathbf{x}_\omega$. Given $K$, let $\mathbf{U}_{\text{combi}, K} := (\mathbf{U}_{k_1}|\mathbf{U}_{k_2}| \ldots |\mathbf{U}_{k_s})$. The recovered estimate $\hat{\mathbf{y}}_\omega$ of $\mathbf{y}_\omega$ is $\hat{\mathbf{y}}_{\omega,K} = \Lambda \mathbf{U}_{\text{combi}, K} M^{-1} \mathbf{z}_{\text{combi}, K}$ where $M := \mathbf{U}_{\text{combi}, K}^T \Lambda \mathbf{U}_{\text{combi}, K} + \sigma_n^2 I(\ell.m)$. The original $\mathbf{x}_\omega$ is thus estimated by $\hat{\mathbf{x}}_{\omega,K} := \Psi \hat{\mathbf{y}}_{\omega,K}$. To measure “smoothness in recovery” we determine the expected value

\[E(\text{MSE}, \ell) := \frac{1}{\binom{N}{\ell}} \sum_{K=1}^{\binom{N}{\ell}} \left( \frac{1}{|\Omega|} \sum_{\omega=1}^{|\Omega|} (\mathbf{x}_\omega - \hat{\mathbf{x}}_{\omega,K})^2 \right)\] (3)

with $(\mathbf{x}_\omega - \hat{\mathbf{x}}_{\omega,K})^2 = (\mathbf{x}_\omega - \hat{\mathbf{x}}_{\omega,K})^T (\mathbf{x}_\omega - \hat{\mathbf{x}}_{\omega,K})$.

The constraints on the set $\{s_j : j \in [64]\}$ above can be satisfied by any one of the many possible subspace arrangements. We randomly generate a number, say 3000, of such arrangements per run. For an explicit image reconstruction, we first choose one arrangement $\mathcal{A}$ uniformly at random and then perform the following steps.

Starting from $\ell = 1$, choose $k_1$ uniformly at random from $[N]$, recover using $\mathbf{z}_{k_1}$ only, and define $R_1 \leftarrow [N] \setminus \{k_1\}$. For $\ell = 2$, choose $k_2$ uniformly at random from $R_1$, recover using the combined packets $\mathbf{z}_{k_1}$ and $\mathbf{z}_{k_2}$, and define $R_2 \leftarrow R_1 \setminus \{k_2\}$. Continue the process for all remaining $\ell \in [3, N]$. Output the reconstructed images in sequence of $\ell$ from 1 to $N$. The desired progressive refinement property can be immediately confirmed. Figure 3 shows a typical gradual recovery using ant as the target image. Finally, we present the $\text{MSE}(\ell)$ value in the analysis plot for all $\ell$ in the range for this specific choice of $K$ in the arrangement $\mathcal{A}$. The logarithmic values $\log(\text{MSE}(\ell))$ as $\ell$ ranges from 1 to $N$ are connected by the red dashed line in the analysis plot.

To highlight the smoothness property we perform statistical analysis on the quality of recovery. For $\ell \in [N]$, we generate numerous $\binom{N}{\ell}$ possible combinations of $\mathbf{z}_{\text{combi}, K}$ from $\mathbf{z}_{k_1}, \mathbf{z}_{k_2}, \ldots, \mathbf{z}_{k_8}$ with $K := \{k_1, k_2, \ldots, k_8\} \subseteq [N]$. We compute the corresponding $\text{MSE}(\ell)$ for each $\ell$ to compare the quality of recovery from $\mathbf{z}_{\text{combi}, K}$ and present the

\[\log \left( \frac{1}{|\Omega|} \sum_{\omega=1}^{|\Omega|} (\mathbf{x}_\omega - \hat{\mathbf{x}}_{\omega,K})^2 \right).\] (4)

These values correspond to the respective subsets $K$. The means as $\ell$ ranges from 1 to $N$ are connected by the blue line.

3. PERFORMANCE

Let $M$, $N$, and $\sigma_n^2$ be fixed. One may use an aggregate $\Lambda$ profile and its corresponding $\Psi$ and $s_j$s for all images, instead of computing the individual $\Lambda$ profile of each image. The values can even be preprogrammed into the routines to save computational time. On many occasions, this does not incur significant penalty in performance although the difference becomes more prominent for images with richer colours and contrast. While comparable outcomes are practically indistinguishable, a closer look at the respective analysis plots confirms that the $\text{MSE}(N)$ value for the individual $\Lambda$ profile is always less than or equal to the value for the aggregate $\Lambda$.

The recovered images in Figure 4 provide the illustration for rain for $m = 4$, $N = 8$, and $\sigma_n^2 = 0.01$. Entries 9 and 10 in Table 1 show the difference in their subspace arrangements.

\[\text{Fig. 3.} \text{ Recovery of ant with } m = 4, N = 8 \text{ and } \sigma^2 = 0.02.\]

\[\text{Fig. 4.} \text{ Analysis on rain with } m = 4, N = 8 \text{ and } \sigma^2 = 0.01.\]
The recovery tends to be smoother when the dimension \( m \) of the data packet is increased, say doubled, while \( N \) and \( \sigma^2 \) are fixed. Figure 5 gives an example when \( m \) is doubled from 4 to 8 for fly with \( N = 8 \) and \( \sigma^2_n = 0.02 \).

![Fig. 5. Recovery analysis on fly with \( N = 8 \) and \( \sigma^2 = 0.02 \). First row: \( m = 4 \). Second row: \( m = 8 \).](image)

We use abstract in Figure 6 to see what happens when we increase the number \( N \) of the data packet from 4 to 8 while keeping \( m = 8 \) and \( \sigma^2 = 0.01 \) fixed. The analysis plots show that while MSE(\( N \)) improves as \( N \) is doubled, there is a relatively higher variance, \( i.e. \), less smooth recovery for a chosen \( \ell \).

![Fig. 6. Recovery analysis on abstract with \( m = 8 \) and \( \sigma^2 = 0.01 \). First row: \( N = 4 \). Second row: \( N = 8 \).](image)

The approach is resilient to noise. This is illustrated by using face in Figure 7 where the noise level increases from \( \sigma^2_n = 0.01 \) to 0.03 while \( m = 4 \) and \( N = 8 \) are fixed. The routine adjusts \( s_j \)'s significantly as can be seen in Entries 5 and 6 in Table 1.

![Fig. 7. Recovery and analysis on face with \( m = 4 \) and \( N = 8 \). First row: \( \sigma^2_n = 0.01 \). Second row: \( \sigma^2_n = 0.03 \).](image)

Changing several system parameters simultaneously does not pose any handling issue in our implementation. Figure 8 highlights this point when we alter both the total number of packets and the noise level for ark.

![Fig. 8. Recovery and analysis on ark with \( m = 4 \). First row: \( N = 4 \) and \( \sigma^2_n = 0.01 \). Second row: \( N = 10 \) and \( \sigma^2_n = 0.02 \).](image)

Overall, we have shown that images of various kinds are perfect target for our holographic sensing approach.
4. REFERENCES


