Supplementary Material: Generative Patch Priors for Practical Compressive Image Recovery

Rushil Anirudh
Lawrence Livermore National Laboratory
Livermore, CA
anirudh1@llnl.gov

Suhas Lohit
Misubishi Electric Research Laboratories
Cambridge, MA
slohit@merl.com

Pavan Turaga
Arizona State University
Tempe, AZ
pturaga@asu.edu

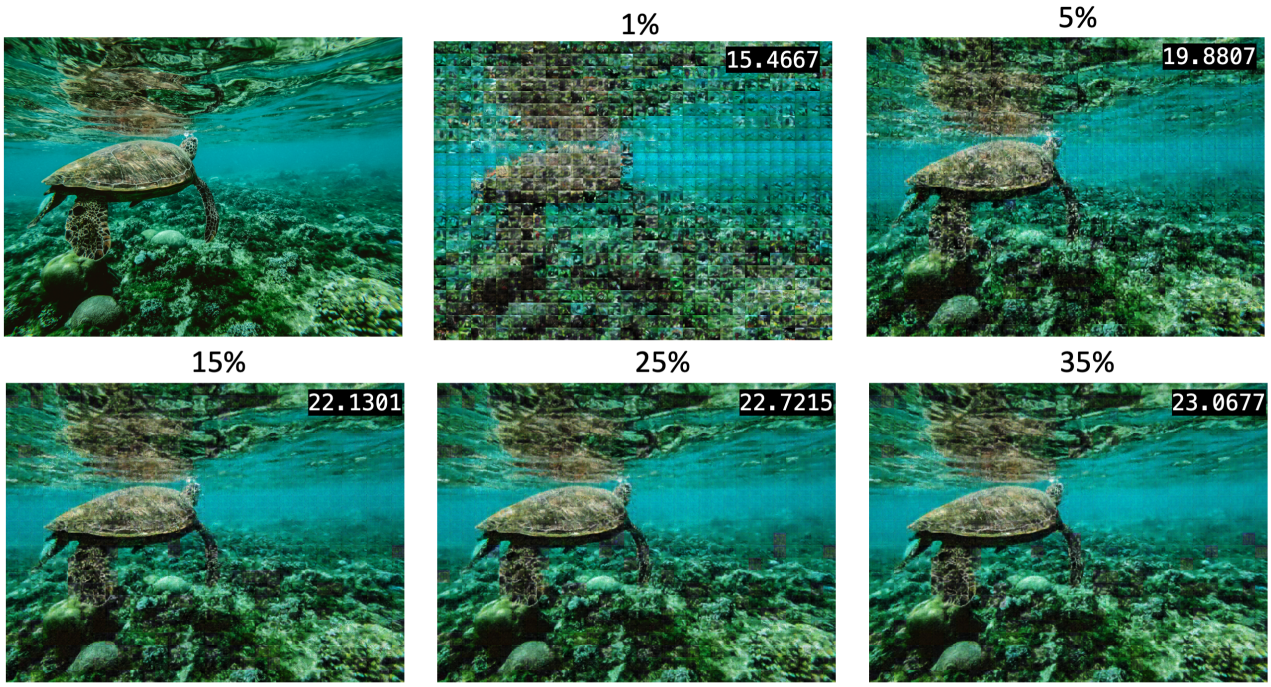


Figure 1: GPP degrades gracefully as the number of available observations are reduced. Results shown here are for an image of size 1024×768, recovered using patches of size 32×32. We do not use BM3D here to illustrate the patch artifacts under very few observations (1%). The PSNR (dB) is also shown along with the reconstruction, compared to the ground truth which is shown in the top left.

1. Additional results

In figure 2, we show sample reconstructions for the phase retrieval task at a measurement rate of 10%.

Figure 2: Compressive phase retrieval sensing at a measurement rate of 10%.

1.1. Self calibration under unknown sensor shift

In figure 3 we illustrate how reconstruction methods can easily fail to recover the solution when there is even a small shift in the operator. We simulate this using $b = -0.25$ and compare the proposed self calibration approach against no calibration and the untrained network prior (DIP) [2]. We observe that the self calibration is able to successfully correct for the unknown shift, compared to the models that do not account for it.

2. Self-Calibrated Compressive Image Recovery

We evaluate the robustness of GPP using the proposed self-calibration (SC) step. In this experiment, we perturb the measurement operator using the perturbation model described in section 4 of the main paper, $\tilde{\Phi} = a * \Phi + b$ using different values for a and b. The measurements are then obtained by $\mathbf{y} = \tilde{\Phi}\mathbf{x}$, but all the reconstruction algorithms, including ours, are given access to only Φ. We study the average PSNR for the seven test images used earlier, for different values of a, and b. In figure 4a, we vary the gain coefficient a, while keeping b fixed at 0.0. We observe that GPP +SC remains robust to a wide variation of a, while the un-calibrated setup completely fails. We repeat these experiments for the sensor shift b coefficient in figure 4b where $a = 1.0$, and we vary b. We observe similarly that GPP +SC is significantly more robust than GPP , or DIP alone. Finally, in figure 4c, we study the convergence of the calibration algorithm for a mixed case with $a = 0.85, b = 0.5$. We see that the self-calibration step converges quickly to a value very close to the true values, and correspondingly improving the PSNR of the reconstruction.

3. Derivation for a^* and b^*

Consider a vectorized square block of an image $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^n$ which we want to sense, and denote by $\mathbf{y} \in \mathbb{R}^m$ the compressive measurements obtained by the sensor. Given a measurement matrix $\Phi \in \mathbb{R}^{m \times n}$, with $m < n$ and $\Phi_{i,j} \sim \mathcal{N}(0, 1)$, the compressive recovery problem is to estimate $\mathbf{x}$ accurately from $\mathbf{y}$. In the ideal setting, i.e., compressive sensing with known calibration the sensing model is given by $\mathbf{y} = \Phi\mathbf{x}$. Instead we consider a simple calibration model— $\mathbf{y} = (a\Phi + b\mathbf{1})\mathbf{x}$, where $a, b \in \mathbb{R}^1$ are unknown calibration parameters and have to be estimated, and $\mathbf{1} \in \mathbb{R}^{m \times n}$ is a matrix of the same size as Φ with 1s.

In order to derive a and b, we assume we have a current estimate of the solution $\mathbf{x}$ from a pre-trained generator $\mathcal{G}(\mathbf{z})$ for a latent vector $\mathbf{z}$. This is be randomly initialized at the beginning. Under this calibration model, let us define mean squared error

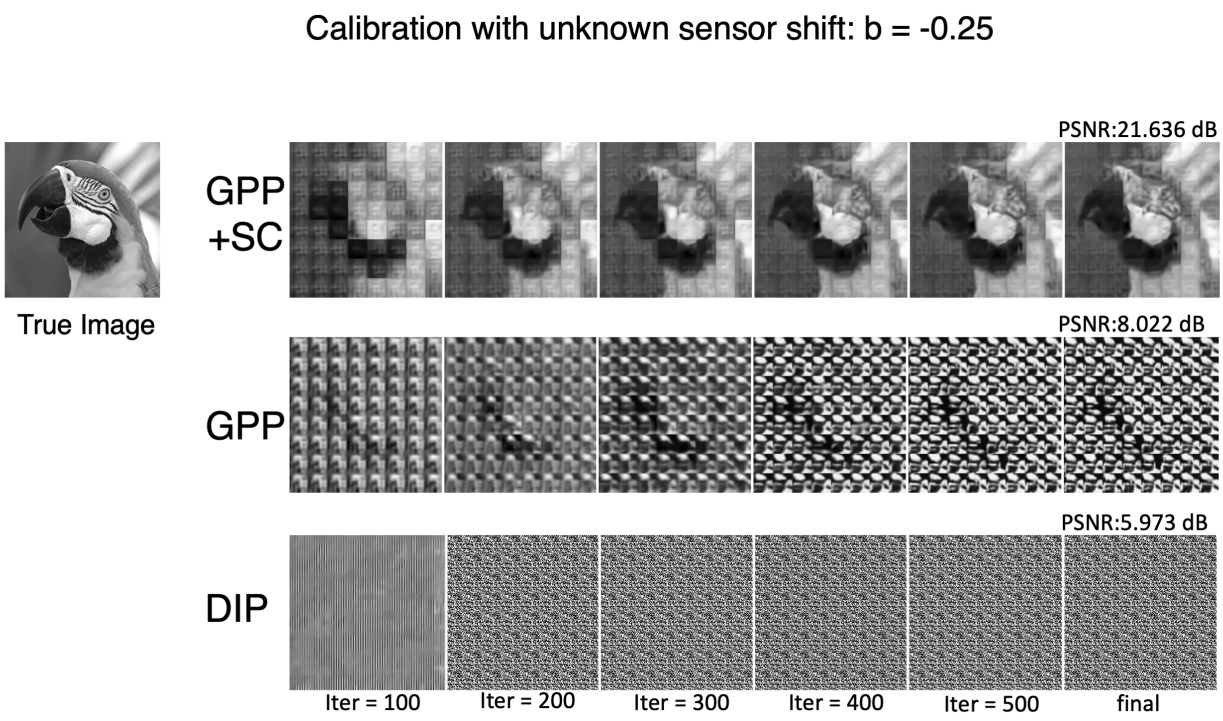
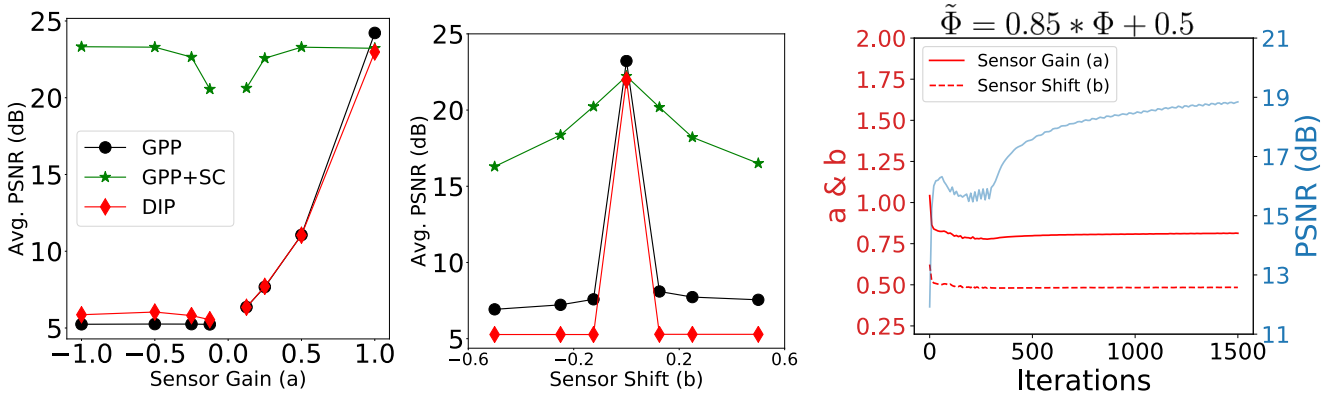


Figure 3: Compressive sensing at a measurement rate of 10% under unknown sensor shift (b). We see that methods that do not account for this shift can easily break. Note the iterations in DIP are scaled since we run it for 10000 iterations compared to 1000 iterations on GPP and GPP + SC.



(a) unknown sensor gain (a)

(b) unknown sensor shift (b)

(c) Convergence of the calibration algorithm

Figure 4: Calibration experiments

loss function as follows:

$$\mathcal{L} = (\mathbf{y} - (a\Phi + b\mathbf{1})\mathbf{x})^\intercal(\mathbf{y} - (a\Phi + b\mathbf{1})\mathbf{x})$$

$$\implies \mathcal{L} = \mathbf{y}^\intercal\mathbf{y} - \mathbf{y}^\intercal(a\Phi + b\mathbf{1})\mathbf{x} - \mathbf{x}^\intercal(a\Phi + b\mathbf{1})^\intercal\mathbf{y} + \mathbf{x}^\intercal(a\Phi + b\mathbf{1})^\intercal(a\Phi + b\mathbf{1})\mathbf{x} \quad (1)$$

As a result, the derivatives with respect to each unknown a, b are:

$$\frac{\partial \mathcal{L}}{\partial a} = -\mathbf{y}^\intercal \Phi \mathbf{x} - \mathbf{x}^T \Phi^\intercal \mathbf{y} + \mathbf{x}^\intercal \left[2a\Phi^\intercal \Phi + b\Phi^\intercal \mathbf{1} + b\mathbf{1}^\intercal \Phi\right] \mathbf{x}$$
$$\text{Similarly, } \frac{\partial \mathcal{L}}{\partial b} = -\mathbf{y}^\intercal \mathbf{1}\mathbf{x} - \mathbf{x}^\intercal \mathbf{1}^\intercal \mathbf{y} + \mathbf{x}^\intercal \left[a\Phi^\intercal \mathbf{1} + \mathbf{1}^\intercal \Phi + 2b\mathbf{1}^\intercal \mathbf{1}\right] \mathbf{x} \tag{2}$$

By setting these derivatives to zero, we get:

$$\frac{\partial \mathcal{L}}{\partial a} = 0 \implies -\mathbf{y}^\intercal \Phi \mathbf{x} - \mathbf{x}^\intercal \Phi^\intercal \mathbf{y} + 2a\mathbf{x}^\intercal \Phi^\intercal \Phi \mathbf{x} + b\mathbf{x}^\intercal \Phi^\intercal \mathbf{1}\mathbf{x} + b\mathbf{x}^\intercal \mathbf{1}^\intercal \Phi \mathbf{x} = 0. \tag{3}$$
$$\implies -2\mathbf{y}^\intercal \Phi \mathbf{x} + 2a\mathbf{x}^\intercal \Phi^\intercal \Phi \mathbf{x} + 2b\mathbf{x}^\intercal \Phi^\intercal \mathbf{1}\mathbf{x} = 0. \tag{4}$$
$$\implies b = \frac{\mathbf{y}^\intercal \Phi \mathbf{x} - a\mathbf{x}^\intercal \Phi^\intercal \Phi \mathbf{x}}{\mathbf{x}^\intercal \Phi^\intercal \mathbf{1}\mathbf{x}} \tag{5}$$

Note, in (4) all the terms are scalars and therefore $\mathbf{y}^\intercal \Phi \mathbf{x} = \mathbf{x}^\intercal \Phi^\intercal \mathbf{y}$ etc. Next, we take the partial derivative with respect to b.

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \implies \mathbf{y}^\intercal \mathbf{1}\mathbf{x} - \mathbf{x}^\intercal \mathbf{1}^\intercal \mathbf{y} + a\mathbf{x}^\intercal \Phi^\intercal \mathbf{1}\mathbf{x} + a\mathbf{x}^\intercal \mathbf{1}^\intercal \Phi \mathbf{x} + 2b\mathbf{x}^\intercal \mathbf{1}^\intercal \mathbf{1}\mathbf{x} = 0. \tag{6}$$
$$\implies -2\mathbf{y}^\intercal \mathbf{1}\mathbf{x} + 2a\mathbf{x}^\intercal \Phi^\intercal \mathbf{1}\mathbf{x} + 2b\mathbf{x}^\intercal \mathbf{1}^\intercal \mathbf{1}\mathbf{x} = 0. \tag{7}$$
$$\implies b = \frac{\mathbf{y}^T \mathbf{1}\mathbf{x} - a\mathbf{x}^\intercal \Phi^\intercal \mathbf{1}\mathbf{x}}{\mathbf{x}^\intercal \mathbf{1}^\intercal \mathbf{1}\mathbf{x}} \tag{8}$$

Combining equations (8) and (5), we get the following:

$$\left(\mathbf{y}^\intercal \Phi \mathbf{x} - a\mathbf{x}^\intercal \Phi^\intercal \Phi \mathbf{x}\right) \mathbf{x}^\intercal \mathbf{1}^\intercal \mathbf{1}\mathbf{x} = \left(\mathbf{y}^T \mathbf{1}\mathbf{x} - a\mathbf{x}^\intercal \Phi^\intercal \mathbf{1}\mathbf{x}\right) \mathbf{x}^\intercal \Phi^\intercal \mathbf{1}\mathbf{x} \tag{9}$$

As in the paper, let us define scalar quantities for notational convenience: $c_\Phi = \mathbf{y}^T \Phi \mathbf{x}, c_\mathbf{1} = \mathbf{y}^T \mathbf{1}\mathbf{x}, \theta_\Phi = (\Phi \mathbf{x})^T (\Phi \mathbf{x}), \theta_\mathbf{1} = (\mathbf{1}\mathbf{x})^T (\mathbf{1}\mathbf{x}), \lambda = (\Phi \mathbf{x})^T (\mathbf{1}\mathbf{x})$. This implies, (9) is now reformulated as:

$$(c_\Phi - a\theta_\Phi)\theta_\mathbf{1} = (c_\mathbf{1} - a\lambda)\lambda \tag{10}$$
$$\implies a = \frac{c_\mathbf{1}\lambda - c_\Phi \theta_\mathbf{1}}{\lambda^2 - \theta_\Phi \theta_\mathbf{1}} \tag{11}$$
$$\text{and } b = \frac{c_\mathbf{1} - a^*\lambda}{\theta_\mathbf{1}} \tag{12}$$

In each step of the alternating minimization, we use the estimates from (11), and (12) and update the latent vector $\mathbf{z}$, which is repeated until convergence in $a, b, \mathbf{z}$. Since our generative model is defined at a patch level, we estimate a, b for each individual patch i separately and assign the mean values of all the patch-estimates as the single a, b for the entire image: $a = \frac{1}{N}\sum_{i=1}^{N} a_i$; and $b = \frac{1}{N}\sum_{i=1}^{N} b_i$. We continue with this alternating minimization until the loss converges. In practice, we find that the algorithm converges within 1500 iterations, and finding recovery and convergence properties of this algorithm remain part of our future work. We empirically study convergence properties of this self-calibrating mechanism under different settings in the supplement.

Note that the latent space optimization is itself a nonconvex optimization problem which is solved only approximately using a gradient-descent type of optimization, yielding a local minimum at each iteration. Therefore the overall optimization problem is nonconvex even though the calibration parameters can be estimated exactly at each step. Using the result in Theorem 3.1 and Lemma 3.2 in the paper by beck [1], we can see that the alternating procedure converges to a stationary point. A stronger result is deferred for future work.

3.1. GPP for image inpainting

GPP is a generic prior to constrain solutions to the natural image manifold. We show an example here of how it can be used in other challenging inverse problems. In figure 5, we illustrate the efficiency of GPP for a for inpainting, where only a small number random pixels are shown, and the task is to recover the original image. Unlike most existing methods, we see that GPP's solution degrades more gracefully than DIP, even recovering some signal when 99.5% of the pixels are missing.

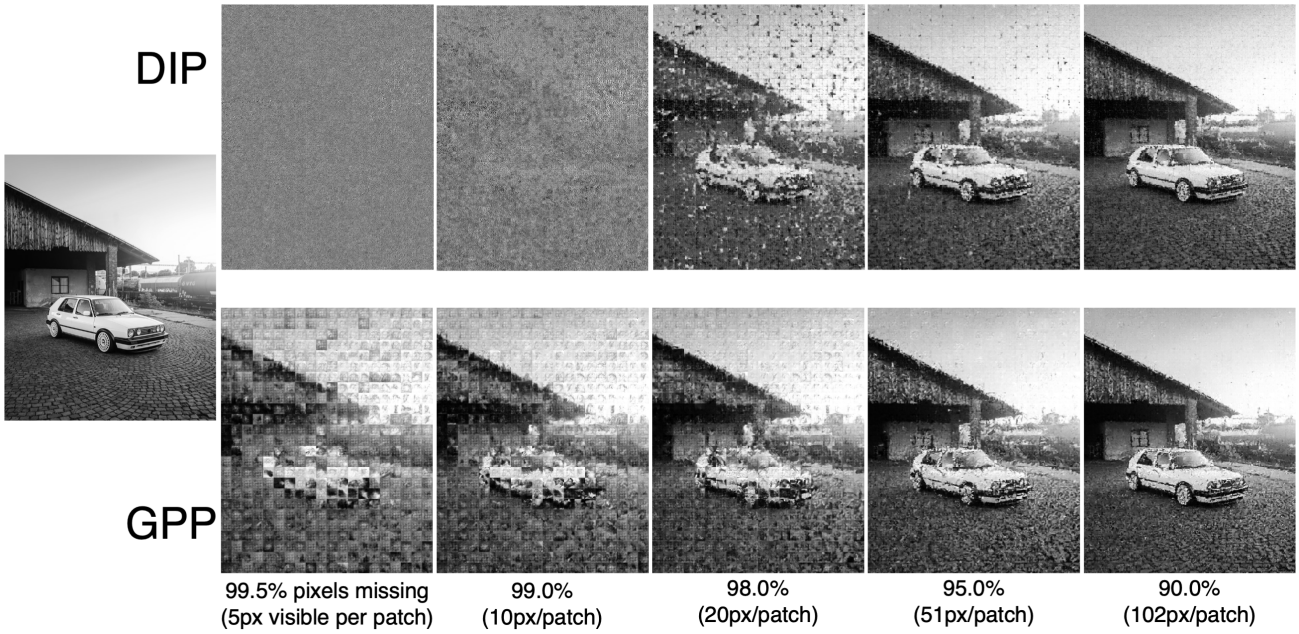


Figure 5: **GPP for image inpainting:** GPP is more efficient than DIP, being able to recover parts of original image even when 99.5 % of the pixels are missing. The original image is of size 800×640.

References

[1] Amir Beck. On the convergence of alternating minimization for convex programming with applications to iteratively reweighted least squares and decomposition schemes. *SIAM Journal on Optimization*, 25(1):185–209, 2015.

[2] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018.