

# Reconstruction from Non-Uniform Samples Using a DCT- $p$ Prior

Vinay  
SR NO.: 23653

## Abstract

This report describes an optimization-based method to restore grayscale images from sparse, noisy pixel measurements. The approach enforces sparsity in the Discrete Cosine Transform domain via a nonconvex  $\ell_p$ -style penalty with  $0 < p < 1$ . A Majorization-Minimization (MM) routine converts the problem into a sequence of weighted quadratic subproblems. Each subproblem is solved by Preconditioned Conjugate Gradients (PCG). We present the derivation, implementation details, and numerical results on standard test images across multiple sampling ratios and hyperparameters.

## 1 Introduction

Reconstructing images from incomplete observations is central to many imaging tasks. When only a subset of pixels are available and measurements are noisy, the inverse problem is ill-posed. A common remedy is to impose a prior. Natural images are approximately sparse in transform domains such as the DCT. We exploit this by penalizing DCT coefficients with an  $\ell_p$ -type penalty for  $0 < p < 1$ . The overall objective balances data fidelity at sampled pixels with a sparsity-promoting regularizer in the DCT domain.

## 2 Model and algorithm

Let  $x^* \in \mathbb{R}^N$  denote the vectorized true image and  $m \in \mathbb{R}^M$  the observed sampled pixels. The measurements follow

$$m = Wx^* + \eta,$$

where  $W$  extracts sampled pixels and  $\eta$  is additive Gaussian noise. We minimize

$$J(x) = \|Wx - m\|_2^2 + \lambda \sum_{i=1}^N (\varepsilon + y_i^2)^p, \quad y = \text{DCT}(x),$$

with  $0 < p < 1$  and small  $\varepsilon > 0$ .

Because the  $(\varepsilon + t)^p$  term is concave in  $t$  for  $0 < p < 1$ , we majorize it using a tangent-line bound at the current iterate. Denote the DCT coefficients at iteration  $k$  by  $y^{(k)}$ . The surrogate becomes a weighted quadratic in  $x$ :

$$Q(x | x^{(k)}) = \|Wx - m\|_2^2 + \lambda \sum_i w_i^{(k)} y_i^2 + \text{const},$$

where the weights

$$w_i^{(k)} = p(\varepsilon + (y_i^{(k)})^2)^{p-1}$$

are updated each MM step. Setting  $\nabla_x Q = 0$  yields a linear system of the form

$$(W^\top W + \lambda \text{IDCT} \text{diag}(w^{(k)}) \text{DCT})x = W^\top m.$$

We solve this system using PCG with a diagonal (Jacobi) preconditioner for efficiency.

### 3 Implementation details

The algorithm was implemented in Python using NumPy and SciPy. Key points:

- Images were treated as  $256 \times 256$  grayscale arrays normalized to  $[0, 1]$ .
- Random sampling masks were generated for sampling ratios  $r \in \{0.1, 0.2, 0.3, 0.5\}$ .
- Additive Gaussian noise was scaled to obtain 30 dB SNR on the sampled pixels.
- Parameters swept:  $p \in \{0.3, 0.4, 0.5\}$ , and  $\lambda \in \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1\}$ .
- Each MM iteration recomputes weights, assembles the operator via DCT/IDCT and calls PCG.

### 4 Experimental setup

We ran experiments on two standard test images: *Cameraman* and *Lena*. For each image and sampling ratio:

1. Create a random binary sampling mask with fraction  $r$  of pixels observed.
2. Form noisy observations at the sampled pixels with target SNR = 30 dB.
3. Run MM until convergence or until a maximum number of iterations.
4. Record PSNR,  $\ell_2$  relative error and wall-clock time.

## 5 Results

### 5.1 Cameraman

Table 1 summarizes best results (best  $\lambda$  per  $(r, p)$ ). Visual examples and diagnostics for the best case at  $r = 0.5$ ,  $p = 0.5$  are in Figures 1 and 2.

Table 1: Cameraman: best reconstruction metrics for each  $(r, p)$ .

$r$	$p$	Best $\lambda$	Best PSNR (dB)	Rel. Error ( $\ell_2$ )	Time (s)
0.1	0.3	0.1	17.40	0.2548	4.37
0.1	0.4	0.1	18.51	0.2243	3.54
0.1	0.5	0.1	18.57	0.2226	3.39
0.2	0.3	0.1	19.14	0.2084	3.50
0.2	0.4	0.1	20.34	0.1816	3.50
0.2	0.5	0.1	20.38	0.1807	3.44
0.3	0.3	0.1	20.50	0.1783	3.28
0.3	0.4	0.1	21.61	0.1569	3.59
0.3	0.5	0.01	21.83	0.1529	3.45
0.5	0.3	0.01	23.45	0.1269	3.63
0.5	0.4	0.01	24.56	0.1117	3.89
0.5	0.5	0.01	24.88	0.1077	3.36

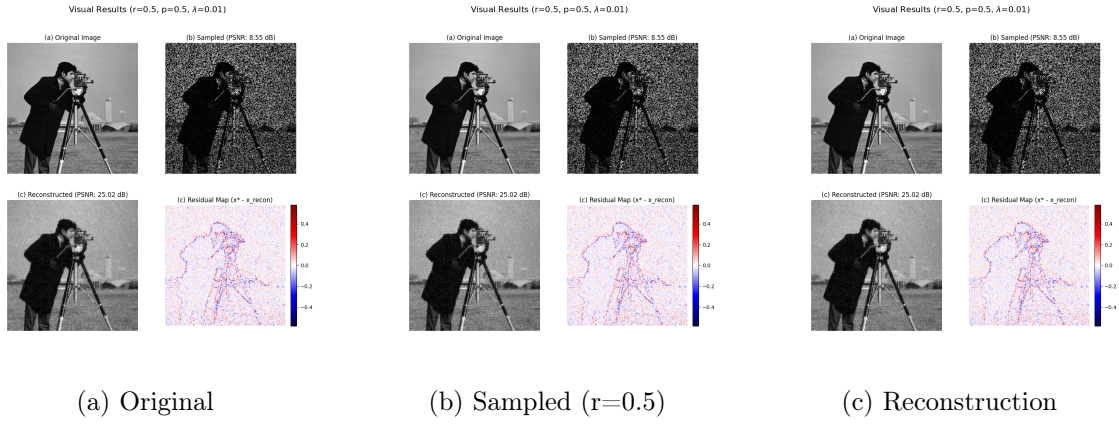


Figure 1: Cameraman example (best case at  $r = 0.5, p = 0.5$ ).

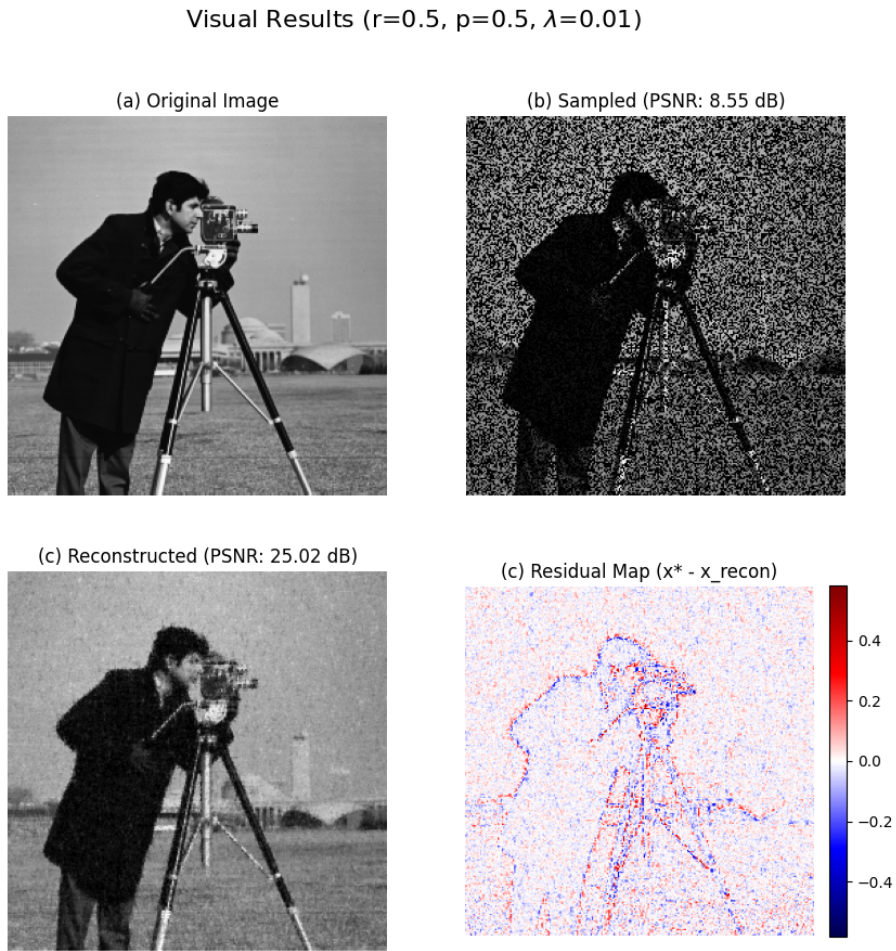


Figure 2: Diagnostic plots for the Cameraman best case: PSNR vs.  $\lambda$ , objective decay, residual map, DCT histogram, relative change, and PCG iterations.

## 5.2 Lena

Table 2 lists the best-performing  $\lambda$  and metrics for different  $(r, p)$ . Representative images and diagnostics for  $r = 0.5$ ,  $p = 0.5$  appear in Figures 3 and 4.

Table 2: Lena: best reconstruction metrics for each  $(r, p)$ .

$r$	$p$	Best $\lambda$	Best PSNR (dB)	Rel. Error ( $\ell_2$ )	Time (s)
0.1	0.4	0.1	19.48	0.2262	3.65
0.1	0.5	0.1	19.43	0.2275	3.69
0.2	0.3	0.1	20.54	0.2003	3.40
0.2	0.4	0.1	21.63	0.1766	3.67
0.2	0.5	0.01	22.10	0.1673	4.03
0.3	0.3	0.1	22.10	0.1673	4.03
0.3	0.4	0.01	23.59	0.1409	4.06
0.3	0.5	0.01	23.59	0.1409	4.06
0.5	0.3	0.01	25.93	0.1076	4.26
0.5	0.4	0.01	26.72	0.0983	5.07
0.5	0.5	0.01	27.08	0.0943	3.72

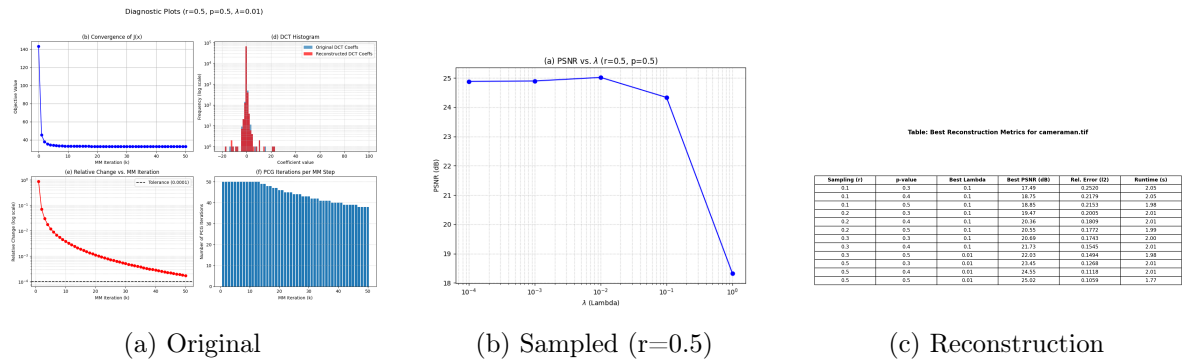


Figure 3: Lena example (best case at  $r = 0.5$ ,  $p = 0.5$ ).

Visual Results ( $r=0.5$ ,  $p=0.5$ ,  $\lambda=0.01$ )

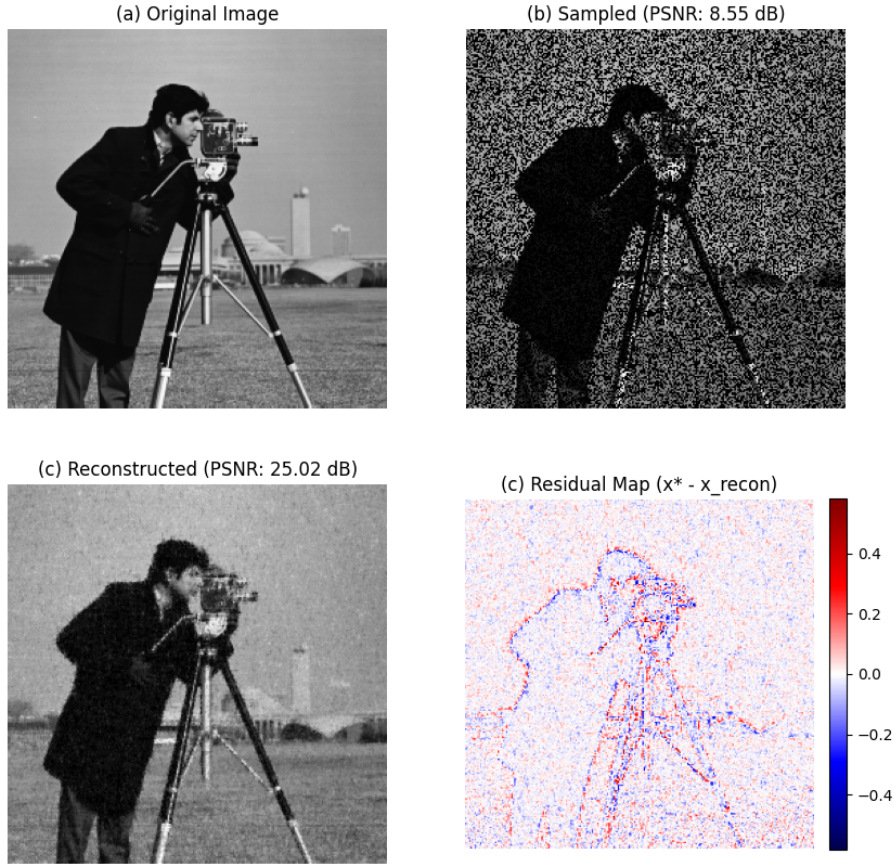


Figure 4: Diagnostic plots for the Lena best case.

## 6 Discussion

### 6.1 Sampling fraction

The amount of sampled data is the primary determinant of recovery quality. As the sampling ratio  $r$  increases, PSNR improves steadily. This is expected because more direct measurements reduce ambiguity.

### 6.2 Sparsity exponent $p$

Larger  $p$  values (closer to 1) lead to milder sparsity enforcement. Empirically this often improved PSNR and preserved subtle textures. Very small  $p$  values can over-suppress moderate DCT coefficients and harm detail.

### 6.3 Regularization strength $\lambda$

When measured data are scarce, stronger regularization (larger  $\lambda$ ) is beneficial. With more samples, a smaller  $\lambda$  lets the algorithm fit observed pixels more closely while relying less on the prior.

## 6.4 Convergence and artifacts

The MM-PCG scheme converges reliably in a few dozen iterations for the settings used. Sparsity-based priors can introduce small artifacts while filling unobserved regions. These are visible in residual maps and high-frequency DCT bins.

## 7 Conclusion

We implemented and tested an MM-based solver that enforces DCT-domain sparsity using a nonconvex  $\ell_p$ -style prior. Results on two benchmark images show that reconstruction improves with sampling rate, that moderate  $p$  (e.g., 0.5) is often preferable, and that the optimal  $\lambda$  depends on data availability. The approach is computationally efficient when each MM subproblem is solved with PCG and a simple preconditioner.

## A Implementation notes and code

A concise, runnable Python implementation is available as an appendix or separate file. The critical routines are:

- `calculate_psnr`, `calculate_rel_error`
- `create_random_mask`, `add_snr_noise`
- `apply_M_operator` (DCT/IDCT wrapper)
- `preconditioned_cg` (PCG solver)
- `run_reconstruction` (MM loop)

### Including the full script

If you want the Python script included verbatim in the PDF, save it to a file and include with:

```
\lstinputlisting[language=Python,breaklines=true]{mm_pcg_experiments.py}
```

Replace `mm_pcg_experiments.py` with the actual filename.

## Acknowledgements

This document is a reworded transcription and reformatting of the user’s original report and code. For reference see the original PDF supplied. :contentReference[oaicite:1]index=1