Image Processing

Rao
Politecnico di Milano

Originally written in: **2025-02-17**

Last updated at: **2025-06-04**

Contents

1 Sparsity and Parsimony

The principle of sparsity or "parsimony" consists in representing some phenomenon with as few variable as possible. Stretch back to philosopher William Ockham in the 14th century, Wrinch and Jeffreys relate simplicity to parsimony:

The **existence of simple laws** is, then, apparently, to be regarded as **a quality of nature**; and accordingly we may infer that it is justifiable to **prefer a simple law to a more complex one that fits our observations slightly better**.

1.1 Sparsity in Statistics

Sparsity is used to **prevent overfitting and improve interpretability of learned models**. In model fitting, the number of parameters is typically used as a criterion to perform model selection. See Bayes Information Criterion (BIC), Akaike Information Criterion (AIC), ..., Lasso.

1.2 Sparsity in Signal Processing

Signal Processing: similar concepts but different terminology. **Vectors corresponds to signals** and data modeling is crucial for performing various operations such as **restoration, compression, solving inverse problems**.

Signals are approximated by sparse linear combinations of **prototypes**(basis elements / atoms of a dictionary), resulting in simpler and compact model.

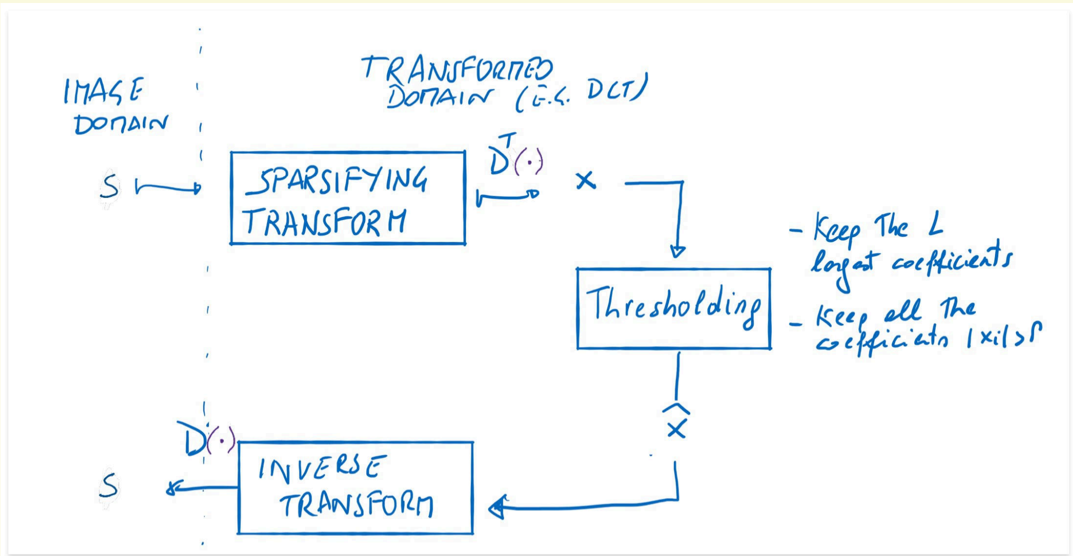


Figure 1.1: Enforce sparsity in signal processing

2 Signal Processing

2.1 Discrete Cosine Transform (DCT)

2.1.1 1D DCT

Generate the DCT basis according to the following formula, the k-th atom of the DCT basis in dimension M is defined as:

$$\mathrm{DCT}_{k(n)} = c_k \cos\left(k\pi \frac{2n+1}{2M}\right) \quad n, k = 0, 1, ..., M-1 \tag{2.1}$$

where $c_0 = \sqrt{\frac{1}{M}}$ and $c_k = \sqrt{\frac{2}{M}}$ for $k \neq 0$.

For each $k = 0, ..., M-1$, just sample each function

$$\mathrm{DCT}_{k(n)} = \cos\left(k\pi \frac{2n+1}{2M}\right) \tag{2.2}$$

at $n = 0, ..., M-1$, obtain a vector. Ignore the normalization coefficient. Divide each vector by its ℓ_2 norm.

Mathematically, suppose the image signal is $s \in \mathbb{R}^M$.

$$x = \mathrm{dct2}(s) = D^T s \tag{2.3}$$

where D^T represents the **DCT basis matrix**. x contains the **DCT coefficients**, which are a **sparse representation** of s.

The **inverse DCT transformation** reconstructs s from x:

$$s = \mathrm{idct2}(x) = Dx \tag{2.4}$$

2.1.2 2D DCT

2D Discrete Cosine Transform (DCT) can be used as a dictionary for representing image patches. A small patch of an image is extracted, represented as s, with dimension $p \times p$. This patch can be **flattened** into a vector of length $M = p^2$, meaning each patch is reshaped into a vector of length M. The **2D-DCT** is used to transform the patch s into DCT coefficients x.

Suppose the image signal is $S \in \mathbb{R}^{M \times N}$.The 2D DCT can be decomposed into two **1D DCT** operations:

1. **Column-wise DCT**: apply **1D DCT** to each column of the image patch: $\mathrm{Z} = D^T S$
2. **Row-wise DCT**: apply **1D DCT** to each column of the image patch: $X^T = D^T \mathrm{Z}^T \rightarrow X = D^T S D$

e.g. JPEG Compression **example 2.1.1**

The image is divided into non-overlapping 8×8 blocks. Each block is treated separately during the compression process.

For each 8×8 block, the **DCT** is applied, transforming pixel values into frequency-domain coefficients. Each 8×8 block's coefficients are checked against a compression threshold τ, coefficients with absolute values below τ are **discarded**(set to zero). The larger the threshold τ, the more coefficients are discarded, leading to **higher compression**.

The compression ratio is defined as:

$$\text{Comp Ratio} = 1 - \frac{\#\text{Non-zero coefficients}}{\#\text{Pixels in the image}} \tag{2.5}$$

To measure how much the image quality is degraded after compression, **Peak Signal-to-Noise Ratio (PSNR)** is used:

$$\text{PSNR} = 10 \log_{10} \left(\frac{1}{\text{MSE}\left(Y, \hat{Y}\right)} \right) \tag{2.6}$$

3 Image Denoising

Image denoising provides a simple and clear problem formulation example. The ***observation model*** is:

$$z(x) = y(x) + \eta(x) \tag{3.1}$$

where:

- $z(x)$ is noisy observation at pixel coordinate x
- $y(x)$ is ideal (noise-free) image
- $\eta(x)$is the noise component
- x is pixel coordinate

We assume that the noise is:

- **Additive Gaussian noise**: $\eta(x) \sim N(0, \sigma^2)$.
- **Independent and identically distributed (i.i.d.)**: Noise realizations at different pixels are independent.

Our goal is to estimate $\hat{y}$ that is close to the true image y.

The observation model provides a **prior on noise** but we also need a **prior on images**.

- **Noise Prior:** Given the Gaussian assumption, the true image y is likely to be a in a circular neighborhood around the observation $z(x)$.
- **Image Prior:** Additional assumptions about the structure of natural images are needed for effective denoising.

3.1 Local Constancy Prior

Assumption: Images are locally constant within small patches. For a constant signal corrupted by Gaussian noise:

$$\hat{y} = \frac{1}{M} \sum_{i=1}^{M} z(x_i) \tag{3.2}$$

Properties of the this estimator:

- **Unbiased:** $\mathbb{E}[\hat{y}] = y$ (true signal)
- **Reduced Variance:** $\mathrm{Var}[\hat{y}] = \frac{\sigma^2}{M}$

Limitations: Local averaging introduces *bias* at edges

3.2 Sparsity-Based Image Prior

3.2.1 Motivation for Sparsity

Natural images have **sparse representations** in certain transform domains (e.g., DCT), as evidenced by the success of JPEG compression.

Key Insight: If images can be sparsely represented for compression, this same property can be leveraged for denoising.

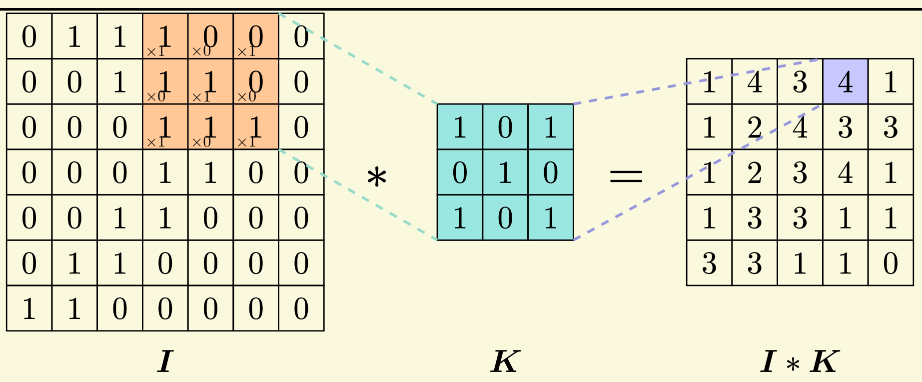
3.2.2 DCT-Based Denoising Pipeline

Step 1 : Analysis

$$X = D^T S \tag{3.3}$$

where:

- S is the vectorized image patch
- D is the DCT basis matrix
- X is the DCT coefficients vector



Step 2: Enforce Sparsity (Thresholding)

$$\hat{X}_i = \begin{cases} X_i & \text{if } |X_i| \geq \gamma \\ 0 & \text{if } |X_i| < \gamma \end{cases} \quad (3.4)$$

Important: Apply thresholding only to the coefficients $i \geq 1$ (preserve the DC component).

Step 3: Synthesis

$$\hat{S} = D\hat{X} \quad (3.5)$$

3.2.3 Universal Denoising (Donoho)

Extreme Value Theory

$$\gamma = \sigma\sqrt{2\log(n)} \quad (3.6)$$

where:

- σ is the noise standard deviation
- n is the dimension of coefficients vector

For 8×8 patches: $\gamma \approx 3\sigma$

3.3 Noise Standard Deviation Estimation

To use the universal thresholding, we need to estimate σ from the noisy image itself.

3.3.1 Robust Estimation Method

Step 1: Compute Image Differences

$$D = Z * [-1, 1] \quad (3.7)$$

This computes differences between adjacent pixels.

Step 2: Robust Standard Deviation Estimation

$$\sigma = \frac{\text{MAD}(D)}{0.6765} \quad (3.8)$$

where MAD = Median Absolute Deviation, defined as:

$$\text{MAD}(D) = \text{median}(|D - \text{median}(D)|) \quad (3.9)$$

Rationale:

- In flat regions: $D \approx \eta_i - \eta_j$
- Robust estimator ignores outliers from edges

3.4 Sliding DCT Algorithm

Processing patches independently creates artifacts at patch boundaries, especially when patches contain edges.

The solution is overlapping patches, where each pixel is processed multiple times.

1. For each pixel (r, c) in the image:
   1. Extract patch S centered at (r, c) with size 8×8.
   2. Apply DCT denoising pipeline: $S \to X \to \hat{X} \to \hat{S}$
   3. Store estimate for center pixel
2. Aggregate all estimates for each pixel

Instead of simple averaging, use weights based on sparsity of coefficients:

$$w = \frac{1}{\text{Number of non-zero coefficients}} \quad (3.10)$$

Rationale: Sparser representations are more reliable under our prior assumption. So the final estimate is:

$$\hat{y}(r, c) = \frac{\sum_{\text{patches}} w\hat{S}}{\sum_{\text{patches}} w} \quad (3.11)$$

3.4.1 Advantages

- Translation invariant (due to sliding window)
- Adaptive filtering (different processing for different patches)
- Handles edges better than simple smoothing
- Leverages natural image statistics

3.4.2 Limitations

- Computational cost (processing every pixel)
- DCT may not be optimal basis for all image patches
- Border effects (fewer estimates at image boundaries)