Denoising problem

until now we've work with the model:

$s = Dx + n$

then the approximated signal is recovered as:

$\hat{y} = D\hat{x} = \sum d_i \hat{x}_i$

where we assume $\hat{x}$ to be sparse ($\|\hat{x}\|_0$ small)

Notation: $y \in \mathbb{R}^M$

since we assume that $\|x_0\| = L$ which is smaller than M
then we can say that y lives in a subspace of M in dimension L.

so, the idea to learn a basis from data is the following:
1. extract patches from image (in a sliding DCT style)
2. unroll them and save them in $S \in \mathbb{R}^{M \times w}$ where $m = p^2$ and w is the number of pixels of the image

---

assumption:
- all patches live in a subspace of dimension L and this subspace is the same for all patches (let's call the orthogonal basis of this subspace D_L), then

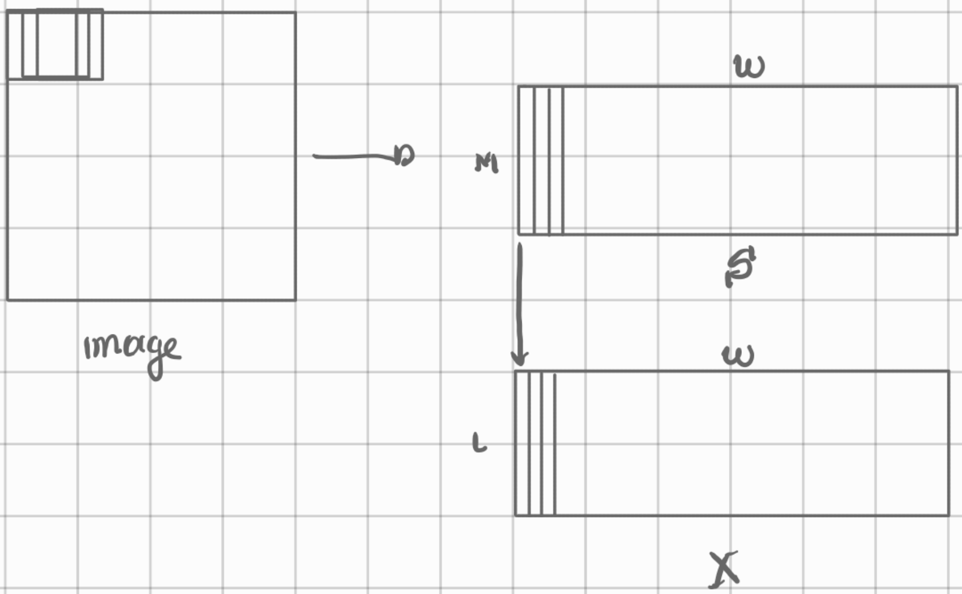
$s = D_L x \quad \forall$ patches where $D_L \in \mathbb{R}^{M \times L}$

since D_L is not square:

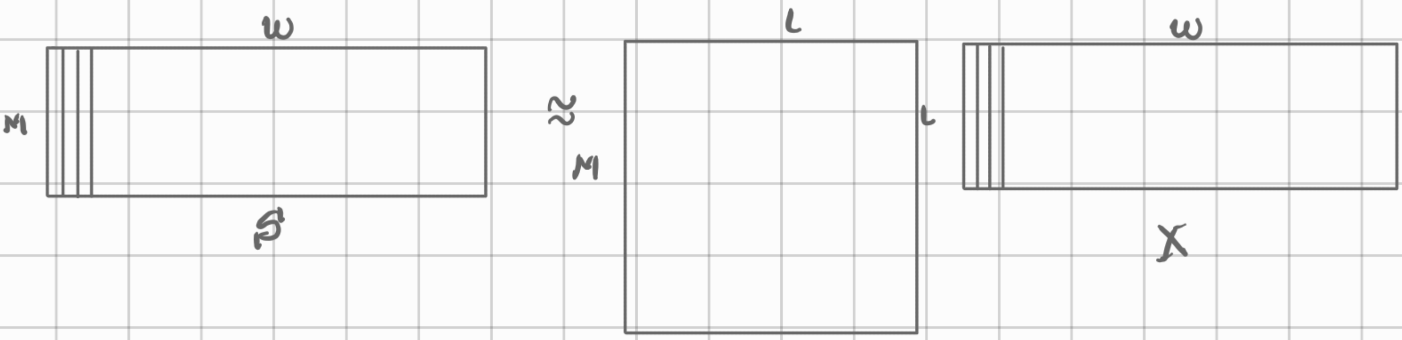
$$\boxed{D_L^T D_L = \mathbb{I}_L \qquad \text{rank}(D_L) = L \qquad D_L D_L^T \neq \mathbb{I}_L}$$

3. calculate X so, $S \approx DX$

Visualization:



then,



so compute D_L we need to

$$\min_{D_L, X} \| S - D_L X \|_F^2$$

$$\| A \|_F^2 = \sum_{i,j} a_{ij}^2$$

$$= \sum_i \sigma_i^2$$

↳ singular values

so instead of minimizing that problem we minimize

$$\min_{Z} \|S - Z\|_F^2 \quad \text{s.t} \quad \text{rank}(Z) \leq L$$

by Eckart-Young-Mirsky theorem we know that:

$$S = U \Sigma V^T$$

then, our solution to Z is:

$$Z = \underbrace{U[:, 0:L]}_{D_L} \underbrace{\Sigma[0:L, 0:L]\ V[:, 0:L]^T}_{X}$$

so, our cost function would be:

$$\|S - Z\|_F^2 = \sum_{i=L+1} \sigma_i^2$$

now, we can not only consider L dimensions, so taking L first colums but taking the entire U.

$$D = U$$

and with this D we will perform the denoising.

Denoising

Steps:

1. estimate D
2. perform hard thresholding to get $\hat{y}$

Algorithm

1. Build S collecting all patches
2. Build S_{avg}, column-wise average | preprocessing
3. $\tilde{S} = S - S_{avg}$
4. $SVD(\tilde{S}) = U\Sigma V^T \rightsquigarrow U = D$

---

optimization:

so supposse $\tilde{S} \in \mathbb{R}^{M \times W}$ where $W \sim 10k$, then taking the SVD can be very time consuming, so we will develop some math to:

$$S = U\Sigma V^T \quad \text{where} \quad U \in \mathbb{R}^{M\times M}, \ \Sigma \in \mathbb{R}^{M\times W}, \ V \in \mathbb{R}^{W\times W}$$

so if we consider:

$$\tilde{S}S^T = U\Sigma V^T V \Sigma^T U^T = \underbrace{U\Sigma\Sigma^T}_{\tilde{\Sigma}} \underset{\tilde{V}^T}{U^T}$$

so we can compute U by:

$$\boxed{SVD(\tilde{S}\tilde{S}^T) = U\tilde{\Sigma}\tilde{V}^T} \quad \boxed{U = D}$$

---

5. perform hard thresholding with aggregation using uniform weight (sliding DCT style)

footnote (PCA bonus):

$\tilde{S}\tilde{S}^T = (S - S_{avg})(S - S_{avg})^T = cov(S)$

$SVD(\tilde{S}\tilde{S}^T) = PCA(S)$

approximation via low dimensional subspace $\longleftrightarrow$ explained variance