INTRODUCTORY APPLIED MACHINE LEARNING

Yan-Fu Kuo

Dept. of Bio-industrial Mechatronics Engineering

National Taiwan University

Today:

- Sparse coding

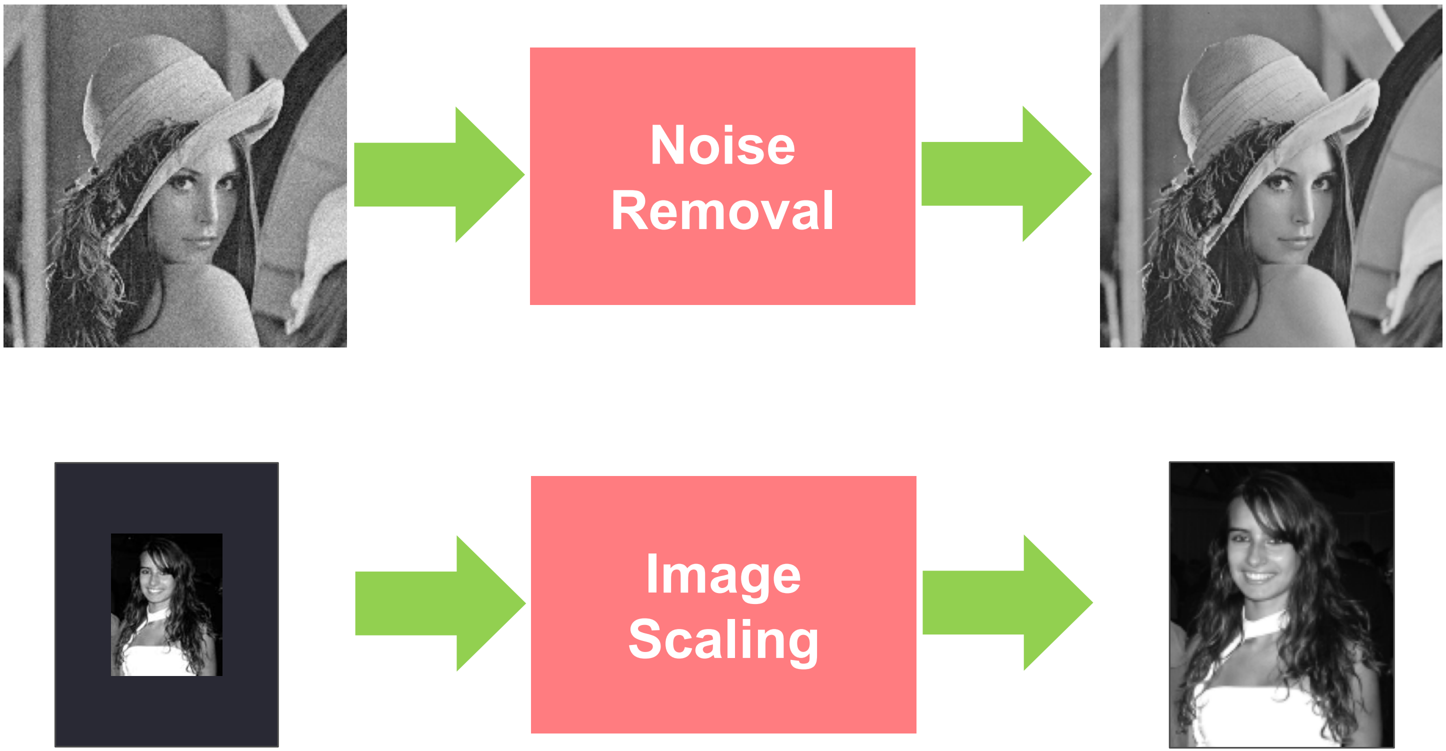
Outline

- Goal of the lecture
- Denoising by sparse representations
- Sparsity and overcompleteness
- Theoretical and numerical foundations
- Dictionary learning and K-SVD algorithm
- Putting it all together

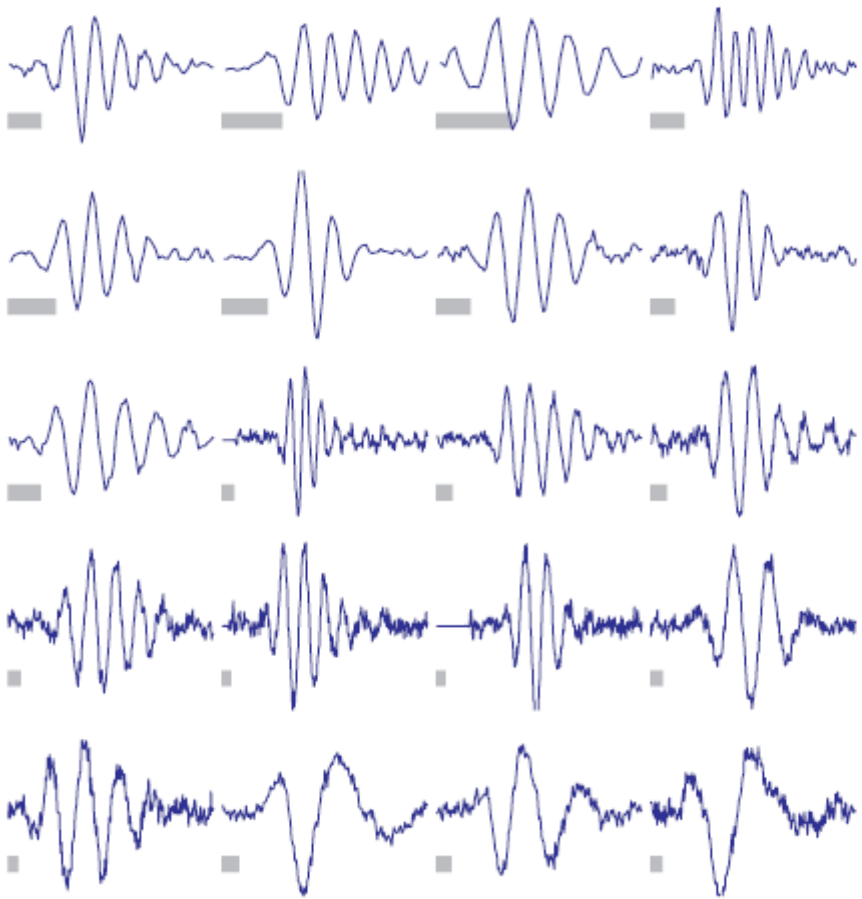
Goals

- After this, you should be able to:
  - Understand the principles of sparse coding
  - Apply sparse coding methods

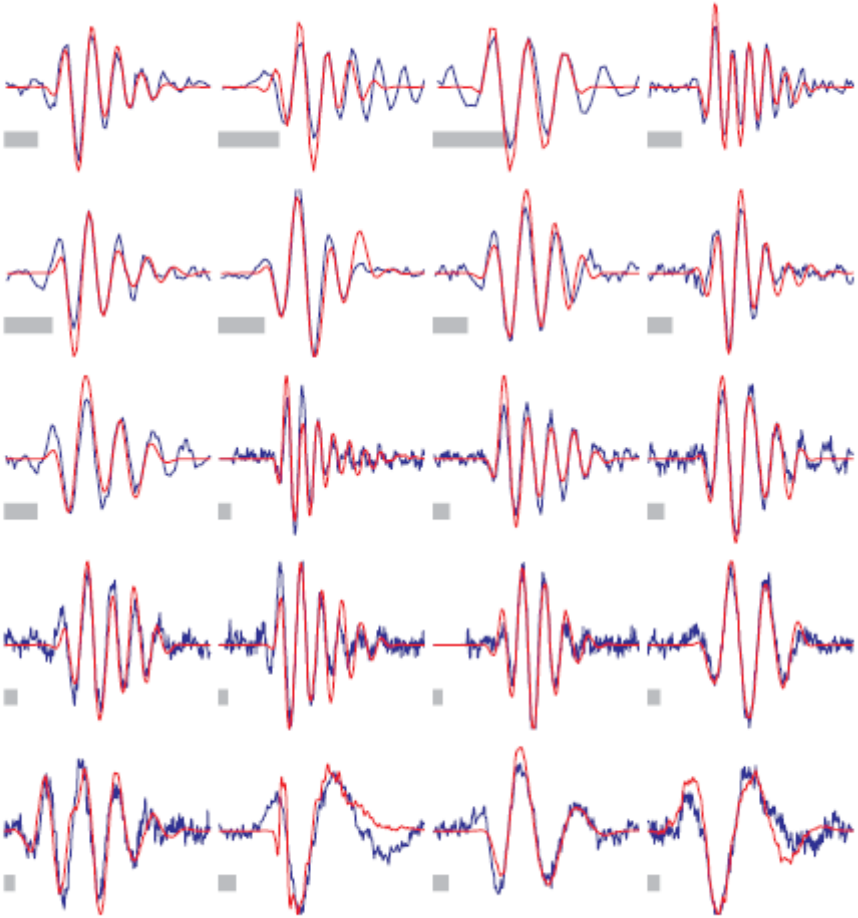
Noise Removal and Image Scaling Problems



Audio Signal with Noise



Noise Reduction



Denoising by Energy Minimization

- Let $\boldsymbol{y} \in \Re^N$ be measurements with noise, and let $\boldsymbol{x} \in \Re^N$ be true signal (which is unknown) to be recovered
- Assume that $\boldsymbol{x}$ can be calculated from a dictionary $\boldsymbol{D}$, i.e., $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha}$
- Denoising is to minimize an energy function:

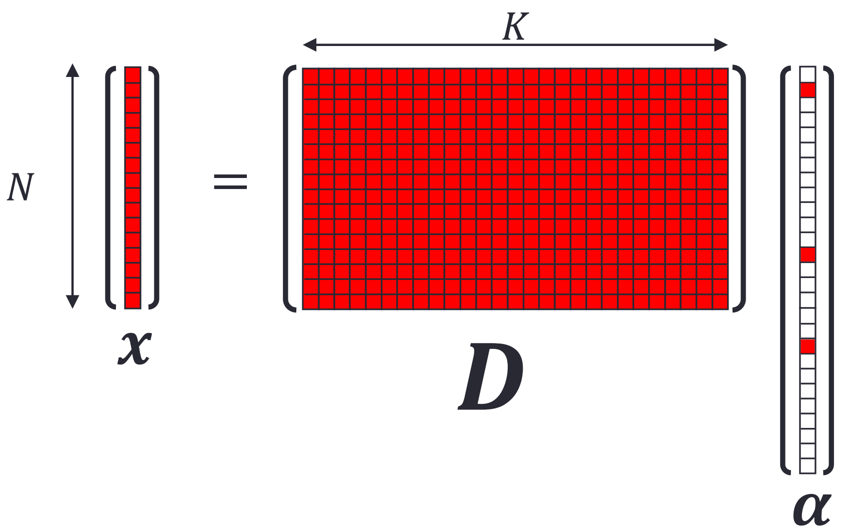
$$f(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \Pr(\boldsymbol{x}) \quad \lambda\|\boldsymbol{\alpha}\|_0^0$$

Relation to measurements

Prior or regularization

- For "sparse" representation, $\Pr(\boldsymbol{x}) = \lambda\|\boldsymbol{\alpha}\|_0^0$ for $\boldsymbol{x} = \boldsymbol{D}\boldsymbol{\alpha}$

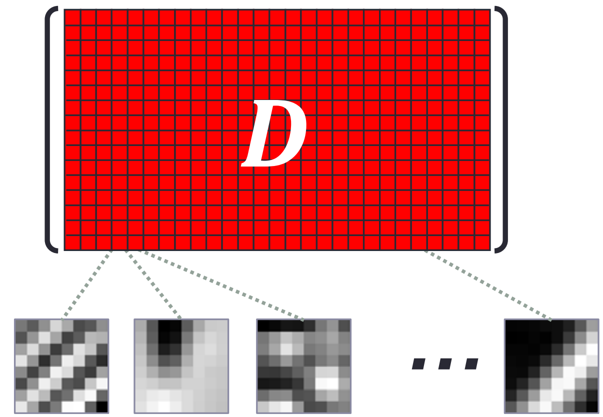
True Signal in Sparse Representation



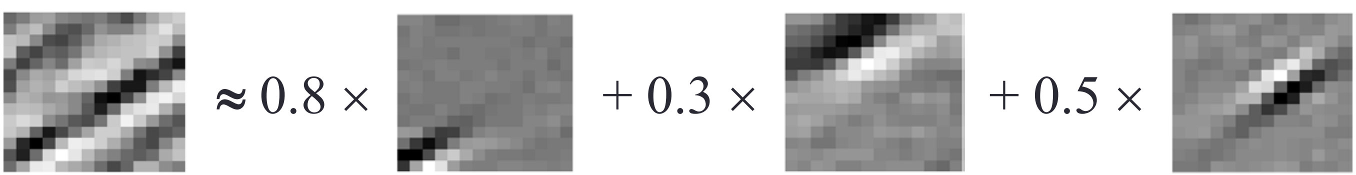
- The true signal $\boldsymbol{x}$ is assumed to be a linear combination of some prototype signal (or atoms) from a dictionary $\boldsymbol{D} \in \Re^{N\times K}$
- The coefficient vector $\boldsymbol{\alpha} \in \Re^{K}$ is a vector with few (say L) non-zeros entries

What is A Dictionary *D*?

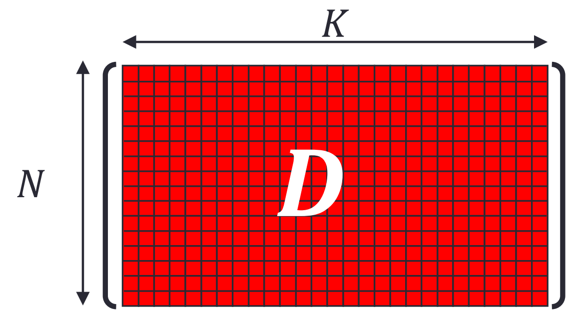
- A dictionary is a signal model that contains many basis (atoms)



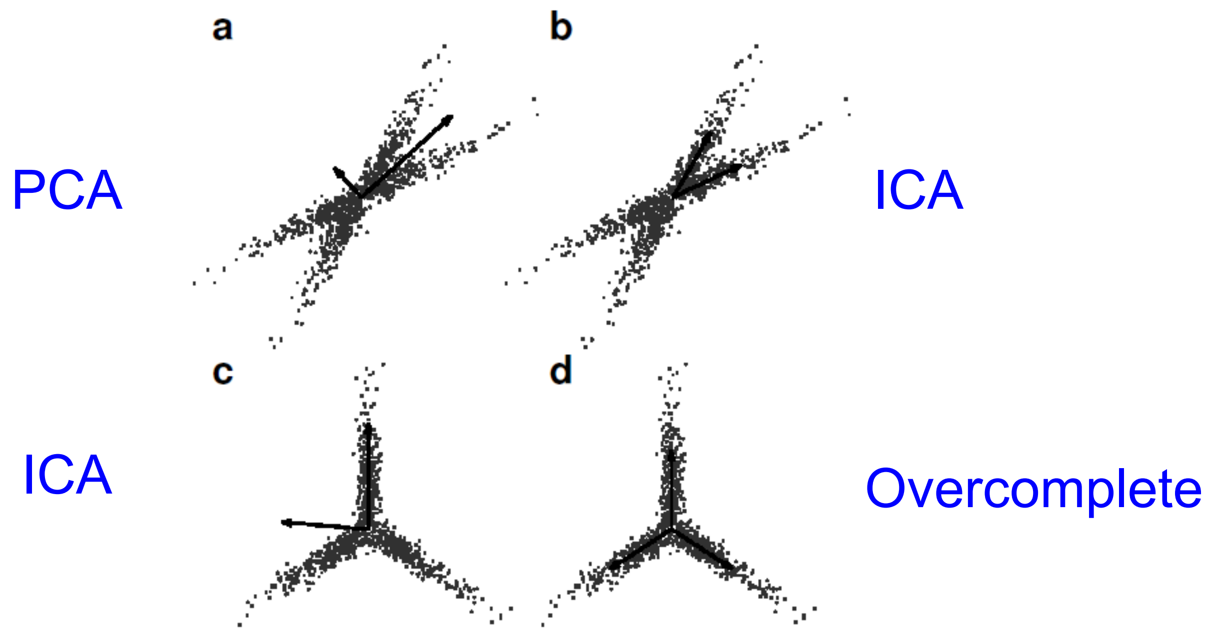
- An image patch is a combination of these atoms



Why Overcomplete?



- The dictionary $\boldsymbol{D}$ is usually overcomplete, i.e., $K > N$
- Why overcomplete?

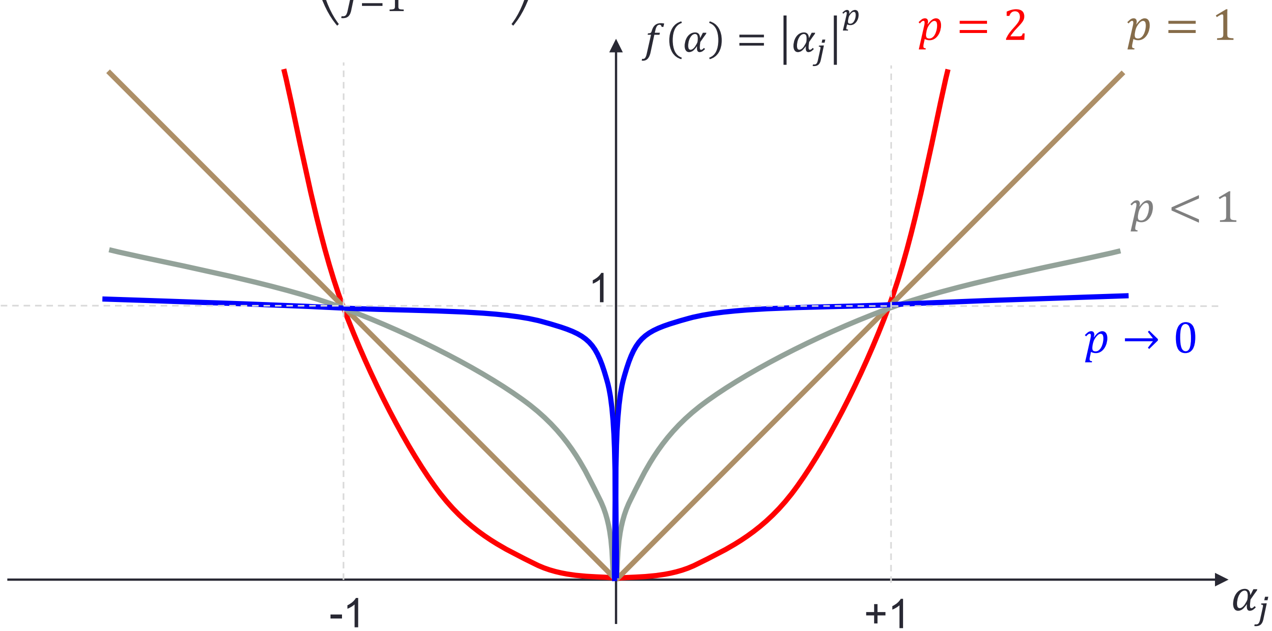


Why Sparse Representation?

- Simple: every signal $\boldsymbol{x}$ is built as a linear combination of a few atoms from the dictionary $\boldsymbol{D}$
- Rich: the obtained signals are a union of many low-dimensional spaces
- Effective: recent works adopt this model and successfully deploy it to applications
- Empirically established: neurological studies show similarity between this model and early vision processes

How to Measure Sparsity? (Why $\|\boldsymbol{\alpha}\|_0^0$?)

- Need a measure of sparsity of $\boldsymbol{\alpha}$, i.e., $\|\boldsymbol{\alpha}\|_p^p = \#\{j: \alpha_j \neq 0\}$
- Note that $\|\boldsymbol{\alpha}\|_p^p \equiv \left(\sum_{j=1}^{K} |\alpha_j|^p\right)^{1/p}$

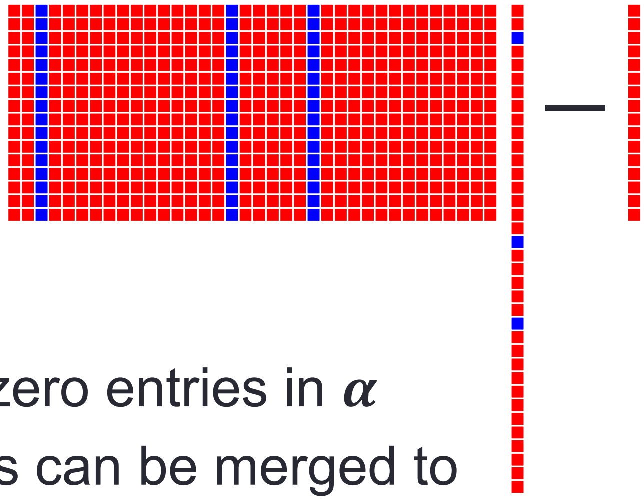


The Sparse Coding Problem

- Assume $\boldsymbol{D}$ and $\boldsymbol{y}$ are known
- What should $\boldsymbol{\alpha}$ be?

$$\hat{\boldsymbol{\alpha}} = \underset{\boldsymbol{\alpha}}{\operatorname{argmin}} \ \frac{1}{2} \| \underbrace{\boldsymbol{x} - \boldsymbol{y}}_{\text{known}} \|_2^2 \quad s.t. \ \ \|\boldsymbol{\alpha}\|_0^0 \leq L$$

$$\boldsymbol{D\alpha} - \boldsymbol{y} =$$



- Need to constrain number of non-zero entries in $\boldsymbol{\alpha}$
- Since only a few (L out of K) atoms can be merged to form the true signal, the noise cannot be fitted well

Issues with the Formulation

- Numerical problem: how should we solve or approximate the solution of the problem?

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{D\alpha} - \boldsymbol{y}\|_2^2 \quad s.t. \quad \|\boldsymbol{\alpha}\|_0^0 \le L$$

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0^0 \quad s.t. \quad \|\boldsymbol{D\alpha} - \boldsymbol{y}\|_2^2 \le \varepsilon^2$$

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{D\alpha} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{\alpha}\|_0^0$$

- Theoretical problem: is there a unique sparse representation?
- Practical problem: what dictionary $\boldsymbol{D}$ should we use, such that all this leads to effective denoising?

Solving the Problem

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0^0 \quad s.t. \quad \|\boldsymbol{D}\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2 \leq \varepsilon^2$$

This is a combinatorial problem, proven to be NP-Hard!

- Recipe for solving this problem:

Set $L = 1$ → Gather all the supports $\{S_i\}_j$ of cardinality L → Solve the LS problem $\min_{\boldsymbol{\alpha}} \|\boldsymbol{D}\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2 \;\; s.t. \;\; \sup p(\boldsymbol{\alpha}) = S_i$ for each support → LS error $> \varepsilon^2$?

No → Set $L = L + 1$ → Gather all the supports

Yes → Done

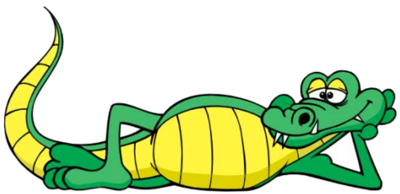
There are $\binom{K}{L}$ such supports

Assume: K=2000, L=10, 1 nano-sec per each LS

We shall need $\sim 8 \times 10^9$ years to solve this problem !!!!!

Approximation

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_0^0 \quad s.t. \quad \|\boldsymbol{D\alpha} - \boldsymbol{y}\|_2^2 \leq \varepsilon^2$$



Relaxation methods

Smooth the L_0 and use continuous optimization techniques

Greedy methods

Build the solution one non-zero element at a time

Relaxation Approach

Solving this instead

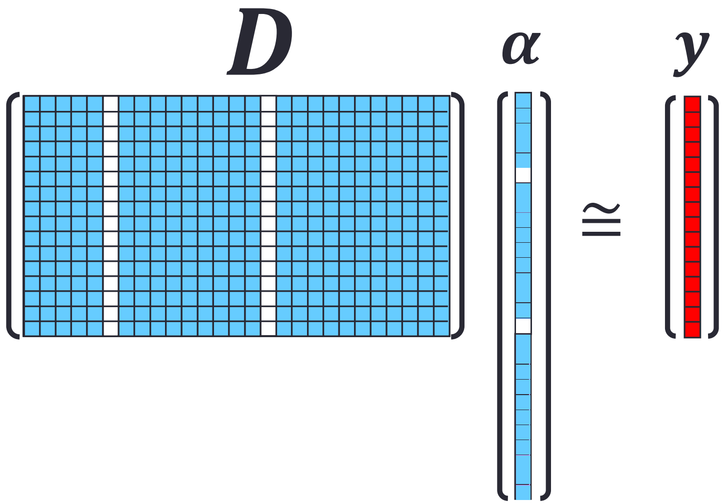
$$\min_{\boldsymbol{\alpha}} ||\boldsymbol{\alpha}||_0^0 \quad s.t. \quad ||\boldsymbol{D\alpha} - \boldsymbol{y}||_2^2 \leq \varepsilon^2$$

$$\min_{\boldsymbol{\alpha}} ||\boldsymbol{\alpha}||_1^1 \quad s.t. \quad ||\boldsymbol{D\alpha} - \boldsymbol{y}||_2^2 \leq \varepsilon^2$$

- Also known as basis pursuit
- The newly defined problem is convex and can be solved using quadratic programming techniques
- Very efficient solvers can be deployed

Greedy Approach

- Also known as matching pursuit (MP)
- Finds one atom at a time
  - First step: find the one atom that best matches the signal
  - Next steps: given the previously found atoms, find the next one to <u>best fit</u> the residual



- The algorithm stops when $\|\boldsymbol{D\alpha} - \boldsymbol{y}\|_2^2 \leq \varepsilon^2$ is satisfied

What Should the Dictionary $\boldsymbol{D}$ Be?

$$\min_{\boldsymbol{\alpha}} \|\boldsymbol{\alpha}\|_1^1 \quad s.t. \quad \|\boldsymbol{D}\boldsymbol{\alpha} - \boldsymbol{y}\|_2^2 \leq \varepsilon^2$$

Assumption: good-behaved images have a sparse representation

$\boldsymbol{D}$ should be chosen such that it sparsifies the representations

Choosing $\boldsymbol{D}$ from a known set of transforms (Fourier, wavelet, consine, etc.)

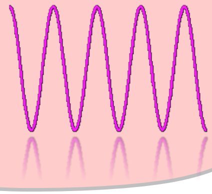
Building $\boldsymbol{D}$ by training it, based on learning from image examples

Some Analytic Dictionaries

Fourier

$$\phi_k(x) = e^{i2\pi kx}$$

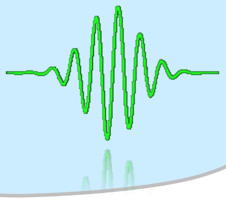
Smooth signals



Gabor

$$\phi_{k,n}(x) = \boldsymbol{\omega}(x - \beta n)\, e^{i2\pi\alpha kx}$$

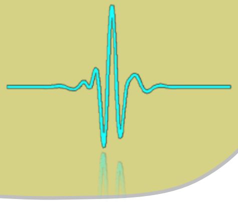
Smooth signals



Wavelets

$$\phi_{m,n}(x) = \alpha^{m/2} f(\alpha^m x - \beta n)$$

Smooth + point singularities

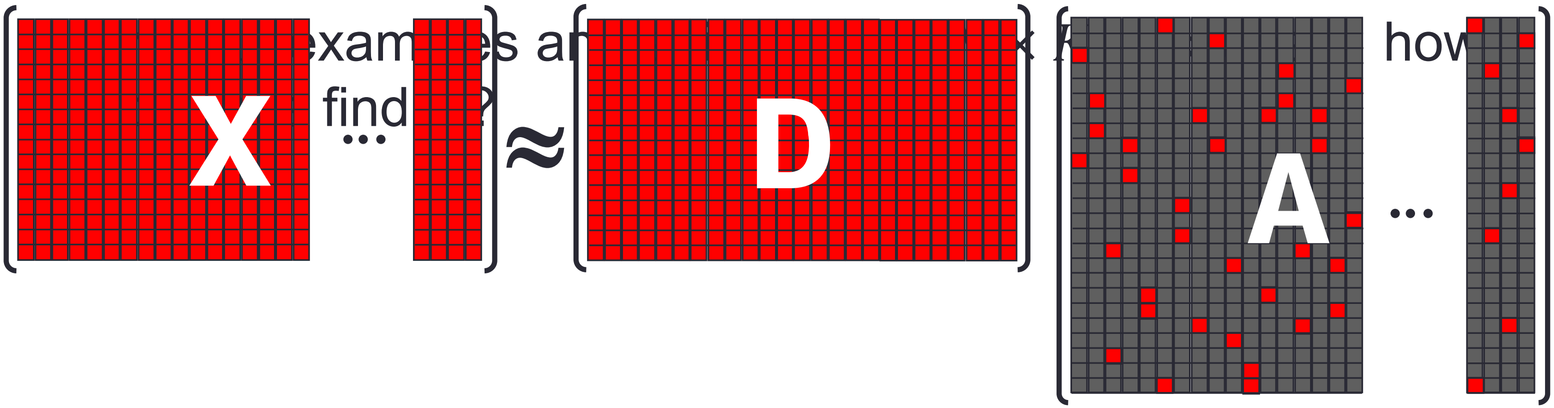


Curvelets

$$\phi_{m,n,\ell}(x) = \phi_m(R_{\Theta_\ell}(x - x_n^{m,\ell}))$$

Smooth + curve singularities

The Dictionary Learning Problem



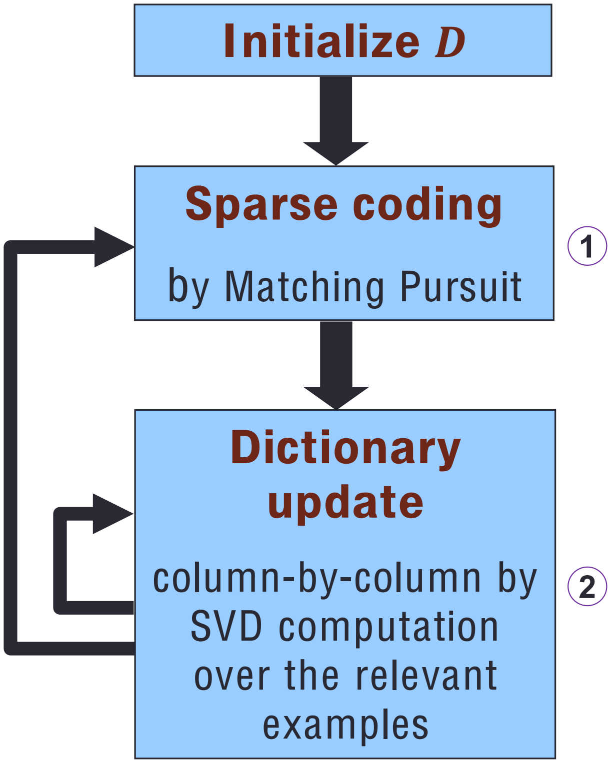
$$\min_{\boldsymbol{D},\boldsymbol{A}} \sum_{j=1}^{P} \|\boldsymbol{D}\boldsymbol{\alpha}_j - \boldsymbol{x}_j\|_2^2 \quad s.t. \quad \|\boldsymbol{\alpha}_j\|_0^0 \le L \ \ \forall j$$

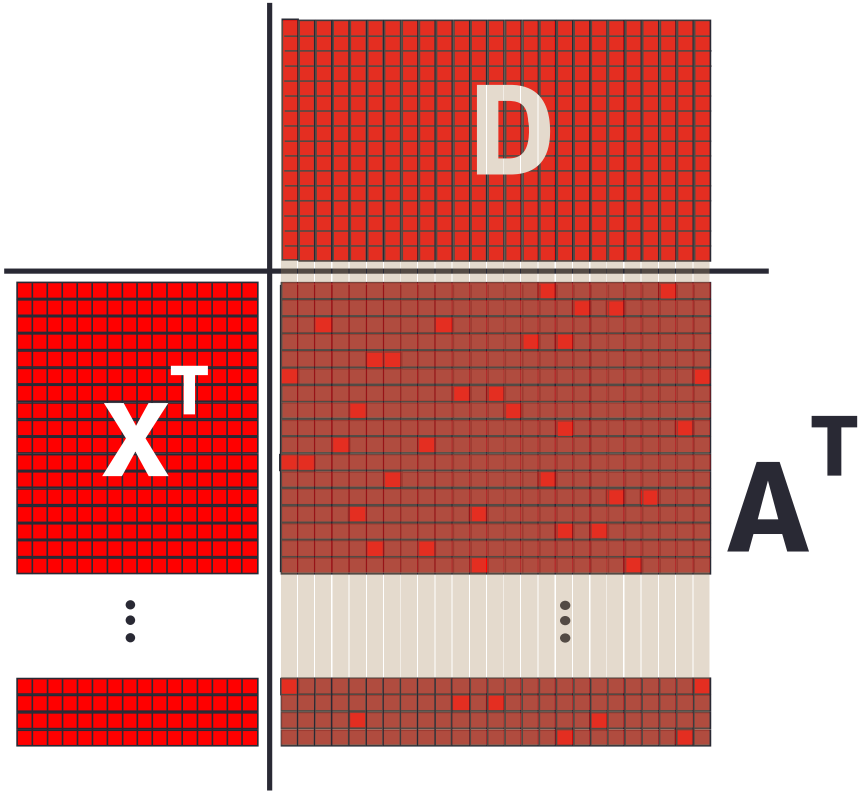
Each example is a linear combination of atoms from $\boldsymbol{D}$

Each example has a sparse representation with no more than $\boldsymbol{L}$ atoms

K-SVD Algorithm – Overview

- Iterative process





K-SVD Algorithm – Sparse Coding

1

- Assume $\boldsymbol{D}$ is known

$$\min_{\boldsymbol{D},\boldsymbol{A}} \sum_{j=1}^{P} \|\boldsymbol{D}\boldsymbol{\alpha}_j - \boldsymbol{x}_j\|_2^2$$

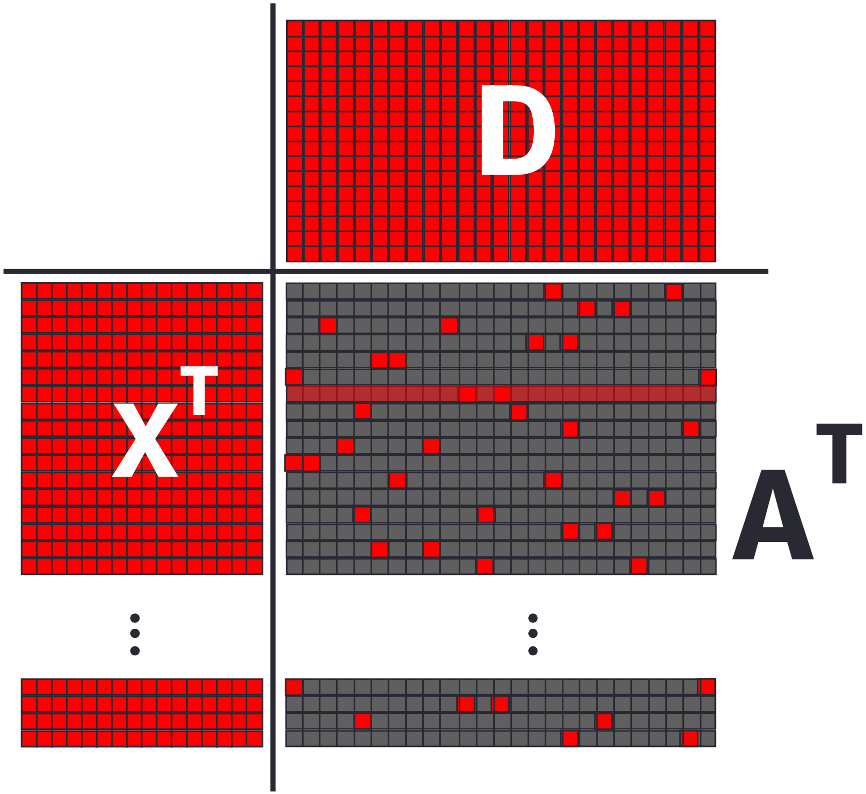
$$s.t. \quad \|\boldsymbol{\alpha}_j\|_0^0 \le L \quad \forall j$$

For the k^{th} raw we solve

$$\min_{\boldsymbol{\alpha}_k} \|\boldsymbol{D}\boldsymbol{\alpha}_k - \boldsymbol{x}_k\|_2^2$$

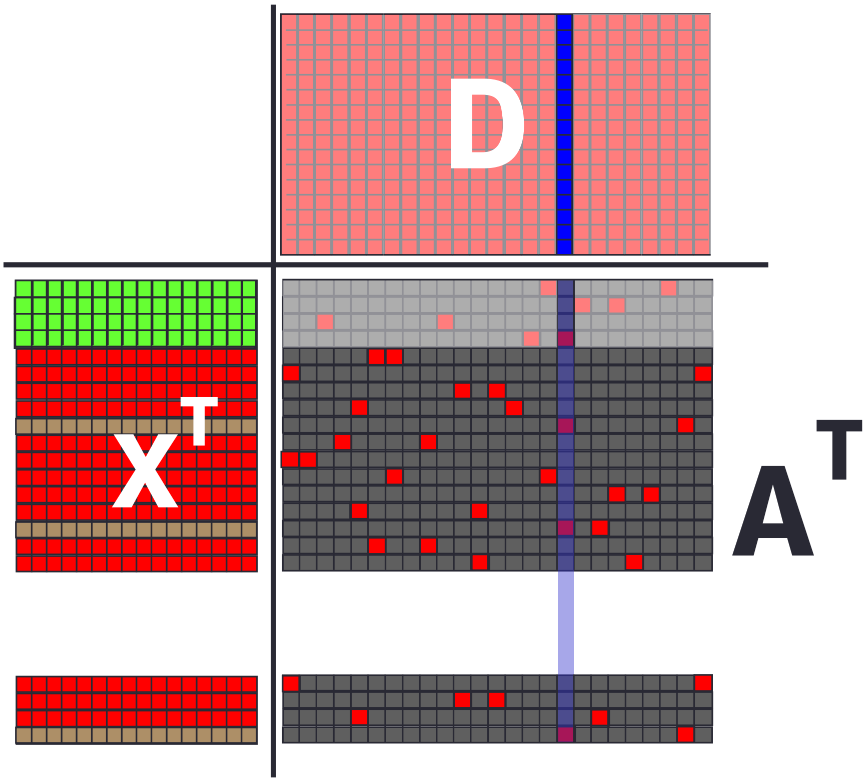
$$s.t. \quad \|\boldsymbol{\alpha}_k\|_p^p \le L$$

Solved by Matching Pursuit



K-SVD Algorithm – Dictionary Learning ②

- Update $\boldsymbol{D}$ and $\boldsymbol{A}$ simutaneously
- Target a column $\boldsymbol{d}_k$
- Identify the examples that use the column $\boldsymbol{d}_k$

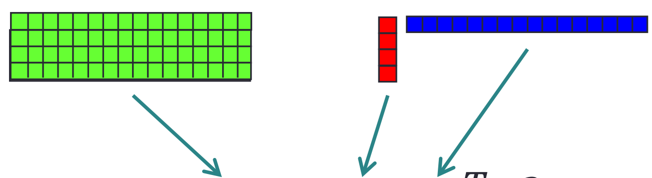


K-SVD Algorithm – Dictionary Update

- A good dictionary column $\boldsymbol{d}_k$ should well describe $\boldsymbol{\Theta}_k$
- Update $\boldsymbol{d}_k$ and $\boldsymbol{\gamma}_k$ by the cost function:

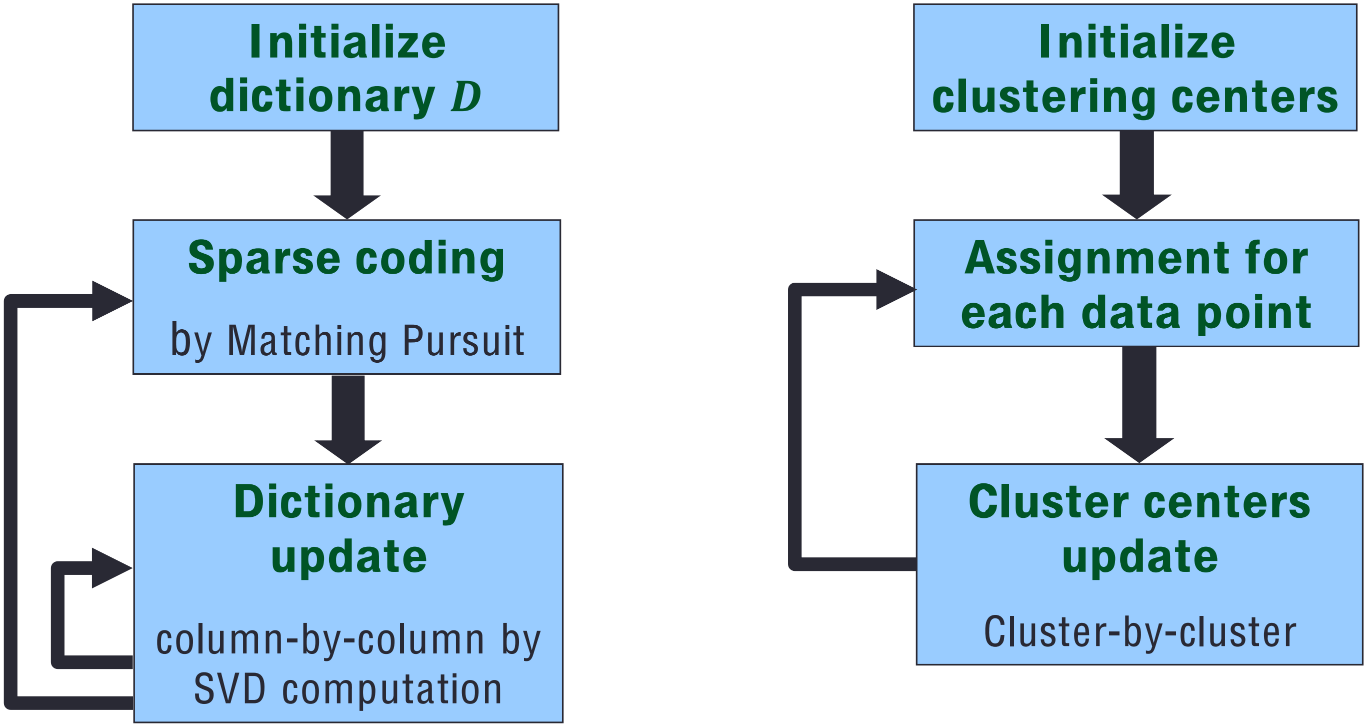
$$\min_{\boldsymbol{d}_k, \boldsymbol{\gamma}_k} \|\boldsymbol{\Theta}_k - \boldsymbol{\gamma}_k \boldsymbol{d}_k{}^{\mathrm{T}}\|_{\mathrm{F}}^2$$

by using SVD



$$\min_{\boldsymbol{d}_k, \boldsymbol{\gamma}_k} \|\boldsymbol{\Theta}_k - \boldsymbol{\gamma}_k \boldsymbol{d}_k{}^{T}\|_{\mathrm{F}}^2$$

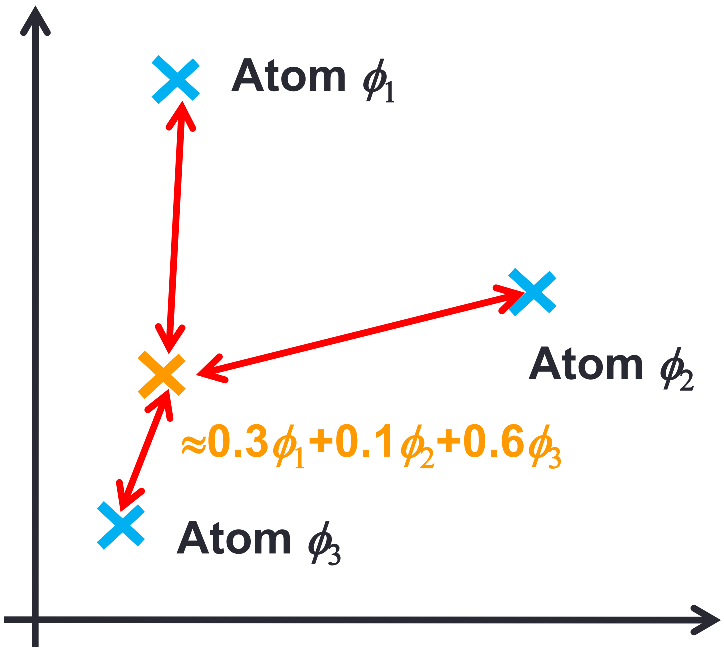
K-SVD vs. K-means



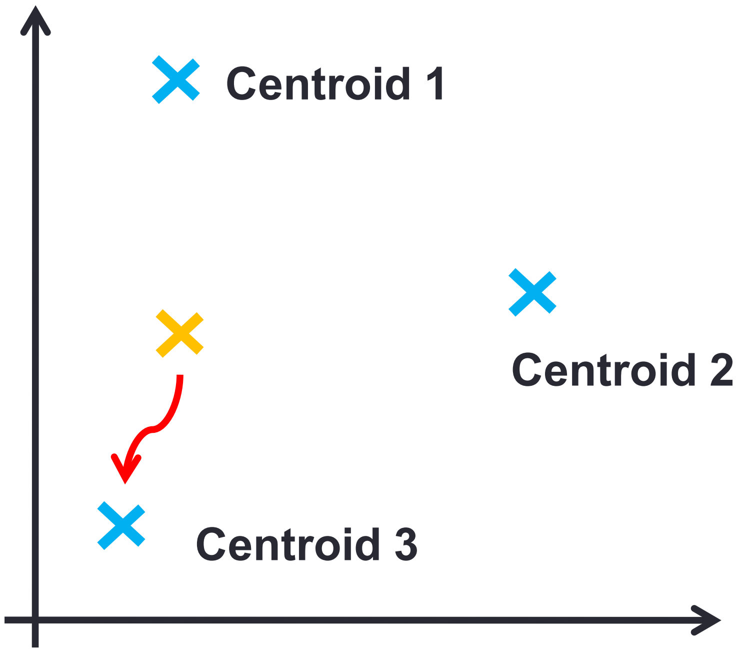
K-SVD vs. K-means

- Atom/centroid training
- Data point clustering

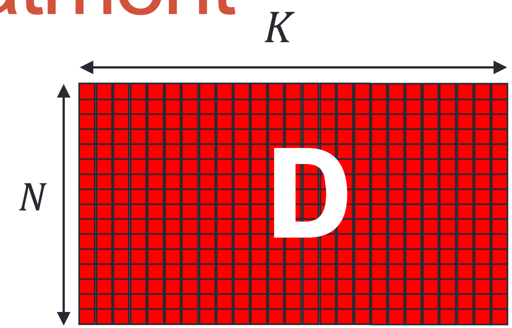
K-SVD



K-means



From Local to Global Treatment



- The K-SVD algorithm is reasonably fast for N in the range of 10 to 400
- As N grows, the complexity and the memory requirements of the K-SVD become prohibitive
- One solution: separate an image into patches of size $\sqrt{N}$-by-$\sqrt{N}$ in the image, including overlaps

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{D},\{\boldsymbol{\alpha}_{ij}\}_{ij}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \mu\sum_{ij}\|\boldsymbol{R}_{ij}\boldsymbol{x} - \boldsymbol{D}\boldsymbol{\alpha}_{ij}\|_2^2 \quad s.t. \quad \|\boldsymbol{\alpha}_{ij}\|_0^0 \le L$$

A binary matrix that extracts a patch in the ij location

What Data to Train On?

- Option 1:
  - Use a database of images
- Option 2:
  - Use the corrupted image itself!!
  - Simply sweep through all patches of size $\sqrt{N}$-by-$\sqrt{N}$ overlapping blocks

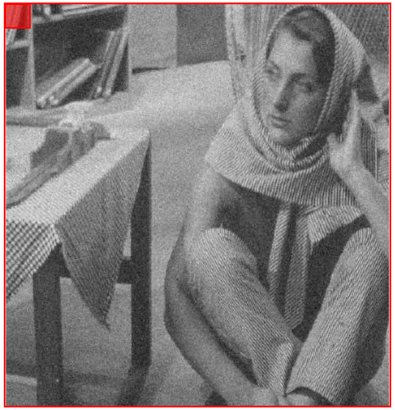


Image Denoising Procedure

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{D},\{\boldsymbol{\alpha}_{ij}\}_{ij}}{\operatorname{argmin}} \frac{1}{2}\|\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \mu\sum_{ij}\|\boldsymbol{R}_{ij}\boldsymbol{x}-\boldsymbol{D}\boldsymbol{\alpha}_{ij}\|_2^2 \quad s.t. \quad \|\boldsymbol{\alpha}_{ij}\|_0^0 \le L$$

$\boldsymbol{x}=\boldsymbol{y}$ and $\boldsymbol{D}$ known

Compute α_{ij} per patch

$$\min_{\boldsymbol{D}} \|\boldsymbol{R}_{ij}\boldsymbol{x}-\boldsymbol{D}\boldsymbol{\alpha}_{ij}\|_2^2 \quad s.t. \quad \|\boldsymbol{\alpha}_{ij}\|_0^0 \le L$$

using the Matching Pursuit

$\boldsymbol{x}$ and $\boldsymbol{\alpha}_{ij}$ known

Compute $\boldsymbol{D}$

$$\min_{\boldsymbol{D}} \sum_{ij}\|\boldsymbol{R}_{ij}\boldsymbol{x}-\boldsymbol{D}\boldsymbol{\alpha}_{ij}\|_2^2$$

using SVD, updating one column at a time

K-SVD

$\boldsymbol{D}$ and $\boldsymbol{\alpha}_{ij}$ known

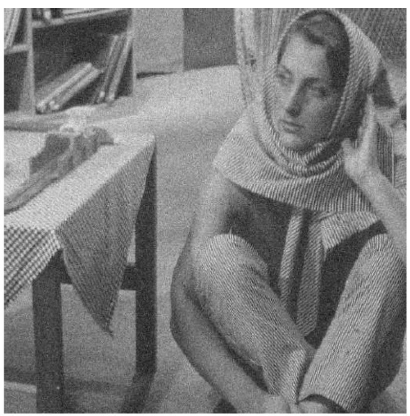
Compute $\boldsymbol{x}$ by

$$\boldsymbol{x} = \left[\boldsymbol{I}+\mu\sum_{ij}\boldsymbol{R}_{ij}^T\boldsymbol{R}_{ij}\right]^{-1}\left[\boldsymbol{y}+\mu\sum_{ij}\boldsymbol{R}_{ij}^T\boldsymbol{D}\boldsymbol{\alpha}_{ij}\right]$$

by averaging of shifted patches

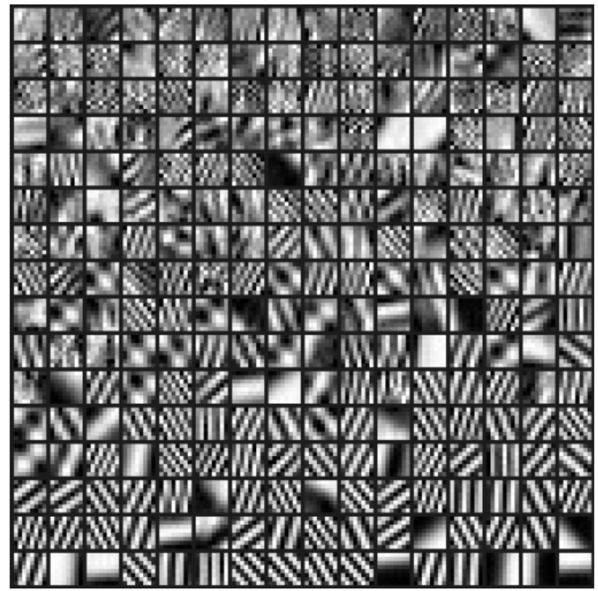
Image Denoising

Source image



Noisy image

Result



The obtained dictionary after 10 iterations

Summary

- Sparsity is an idea that can be used in designing tools to perform deonsing in signal/image processing
- The K-SVD algorithm is an efficient tool that can be applied to perform sparse coding and dictionary learning

References

- R. Rubinstein, *Introduction to Sparse Representation and the K-SVD Algorithm*
- R. Rubinstein, *Sparsity-Based Signal Models and the Sparse K-SVD Algorithm*
- Andrew Ng, Image Classification using Sparse Coding, ECCV10 Tutorial