Image Denoising via Learned Dictionaries and Sparse Representations *

* Joint work with Michal Aharon

Michael Elad

The Computer Science Department

The Technion – Israel Institute of technology

Haifa 32000, Israel

IEEE Computer Society Conference on
Computer Vision and Pattern Recognition (CVPR)

New-York, June 18-22, 2006

Noise Removal ?

Our story focuses on image denoising …

Remove Additive Noise ?

- ❑ Important: (i) Practical application; (ii) A convenient platform (being the simplest inverse problem) for testing basic ideas in image processing.
- ❑ Many Considered Directions: Partial differential equations, Statistical estimators, Adaptive filters, Inverse problems & regularization, Example-based techniques, Sparse representations, …

Part I:

Sparse and Redundant Representations?

Denoising By Energy Minimization

Many of the proposed denoising algorithms are related to the minimization of an energy function of the form

$$f(\underline{x}) = \boxed{\frac{1}{2}\left\|\underline{x}-\underline{y}\right\|_2^2} + \boxed{\Pr(\underline{x})}$$

$\underline{y}$: Given measurements

$\underline{x}$: Unknown to be recovered

Relation to measurements

Prior or regularization

- This is in-fact a Bayesian point of view, adopting the Maximum-Aposteriori Probability (MAP) estimation.
- Clearly, the wisdom in such an approach is within the choice of the prior – **modeling the images** of interest.

Thomas Bayes
1702 - 1761

The Evolution Of Pr($\underline{x}$)

During the past several decades we have made all sort of guesses about the prior Pr($\underline{x}$) for images:

$$\Pr(\underline{x}) = \lambda\|\underline{x}\|_2^2$$

Energy

$$\Pr(\underline{x}) = \lambda\|\mathbf{L}\underline{x}\|_2^2$$

Smoothness

$$\Pr(\underline{x}) = \lambda\|\mathbf{L}\underline{x}\|_{\mathbf{W}}^2$$

Adapt+ Smooth

$$\Pr(\underline{x}) = \lambda\rho\{\mathbf{L}\underline{x}\}$$

Robust Statistics

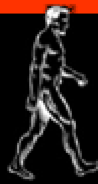
$$\Pr(\underline{x}) = \lambda\|\nabla\underline{x}\|_1$$

Total-Variation

$$\Pr(\underline{x}) = \lambda\|\mathbf{W}\underline{x}\|_1$$

Wavelet Sparsity

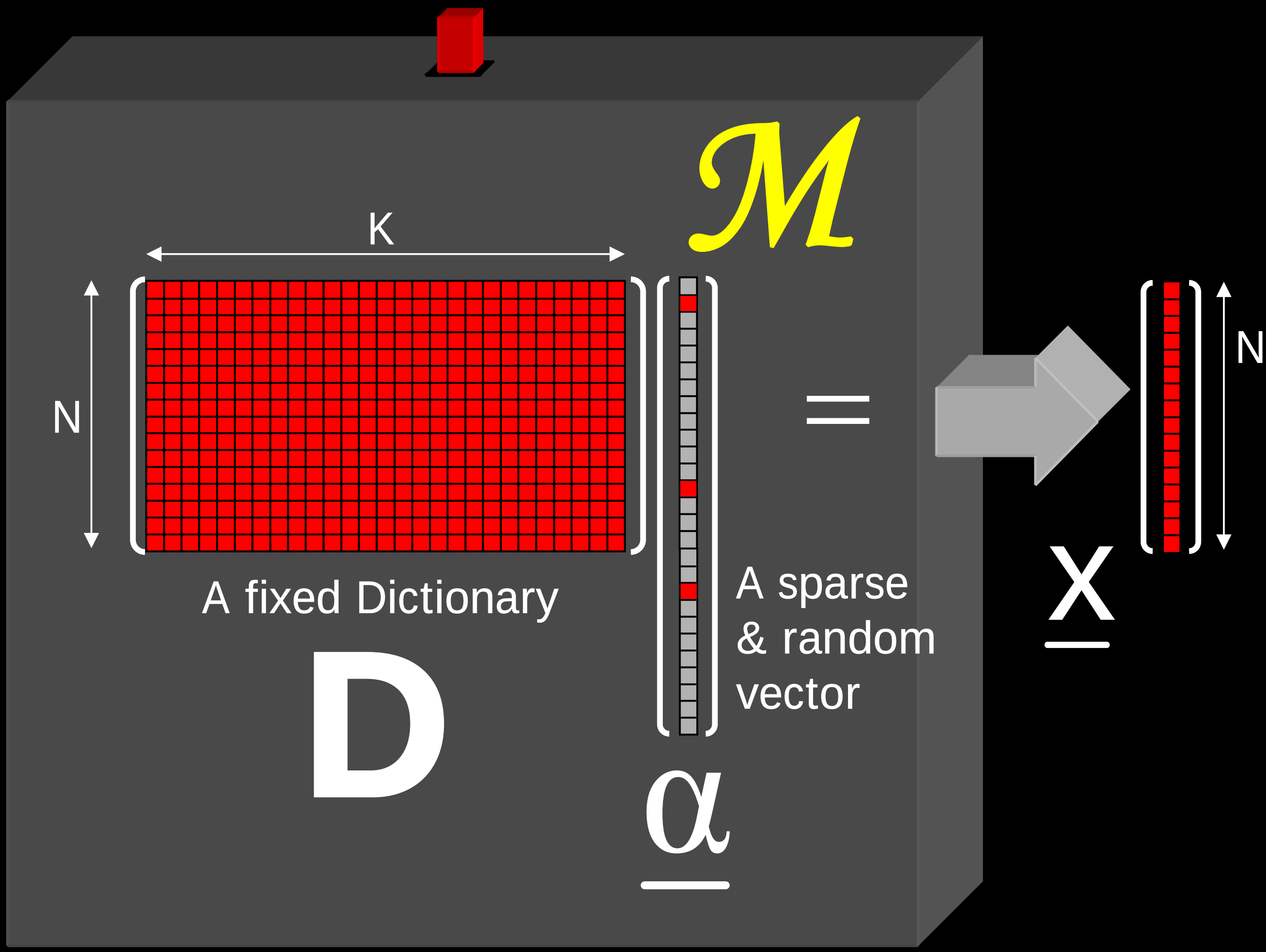
$$\Pr(\underline{x}) = \lambda\|\underline{\alpha}\|_0^0 \quad \text{for} \quad \underline{x} = \mathbf{D}\underline{\alpha}$$



Sparse & Redundant

- Mumford & Shah formulation,
- Compression algorithms as priors,
- …

The *Sparseland* Model for Images



- Every column in **D** (dictionary) is a prototype signal (Atom).
- The vector $\underline{\alpha}$ is generated randomly with few (say L) non-zeros at random locations and with random values.

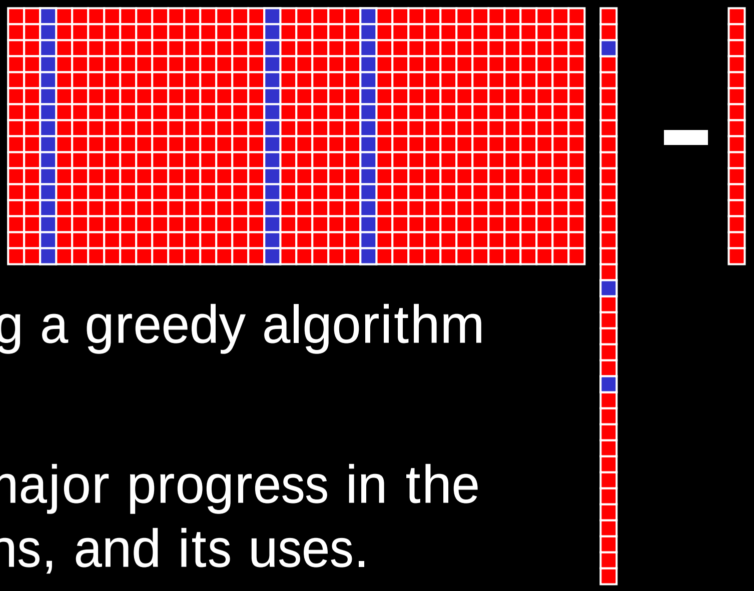
Our MAP Energy Function

- We L_o norm is effectively counting the number of non-zeros in $\underline{\alpha}$.

- The vector $\underline{\alpha}$ is the representation (**sparse**/**redundant**).

$$\frac{1}{2}\left\| \underline{x} - \underline{y} \right\|_2^2$$

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



- The above is solved (approximated!) using a greedy algorithm - the Matching Pursuit [Mallat & Zhang (`93)].

- In the past 5-10 years there has been a major progress in the field of sparse & redundant representations, and its uses.

What Should D Be?

$$\hat{\underline{\alpha}} = \underset{\underline{\alpha}}{\arg\min} \frac{1}{2} \left\| \mathbf{D}\underline{\alpha} - \underline{y} \right\|_2^2 \text{ s.t. } \left\| \underline{\alpha} \right\|_0^0 \leq L \quad \Rightarrow \quad \hat{\underline{x}} = \mathbf{D}\hat{\underline{\alpha}}$$

Our Assumption: Good-behaved Images have a sparse representation

D should be chosen such that it sparsifies the representations

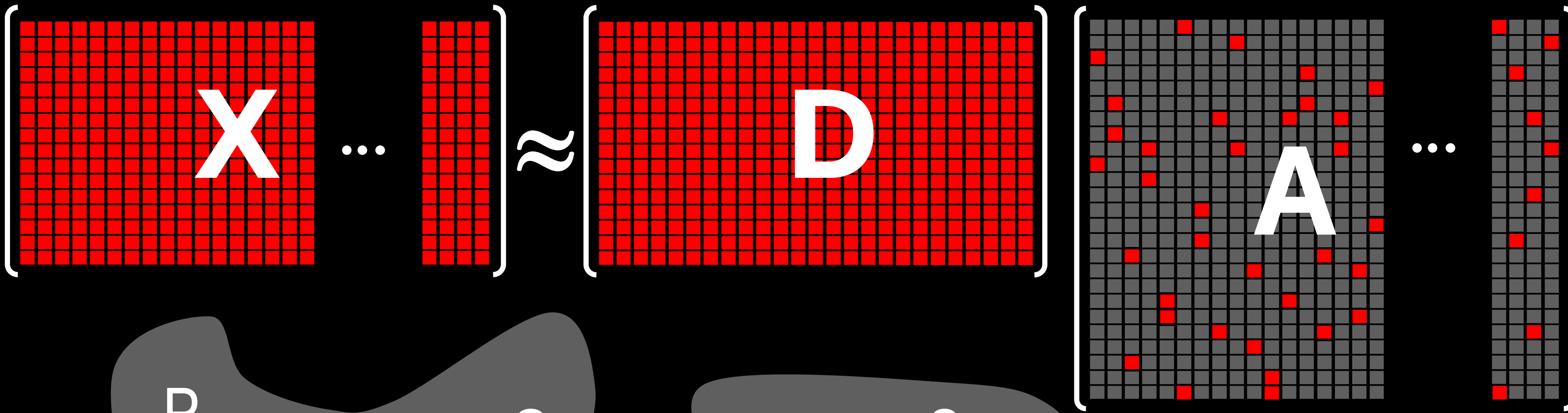
One approach to choose **D** is from a known set of transforms (Steerable wavelet, Curvelet, Contourlets, Bandlets, ...)

The approach we will take for building **D** is training it, based on **Learning** from **Image Examples**

Part II:

Dictionary Learning: The K-SVD Algorithm

Measure of Quality for D



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

Each example is a linear combination of atoms from **D**

Each example has a sparse representation with no more than L atoms

Field & Olshausen ('96)
Engan et. al. ('99)
Lewicki & Sejnowski ('00)
Cotter et. al. ('03)
Gribonval et. al. ('04)
Aharon, Elad, & Bruckstein ('04)
Aharon, Elad, & Bruckstein ('05)

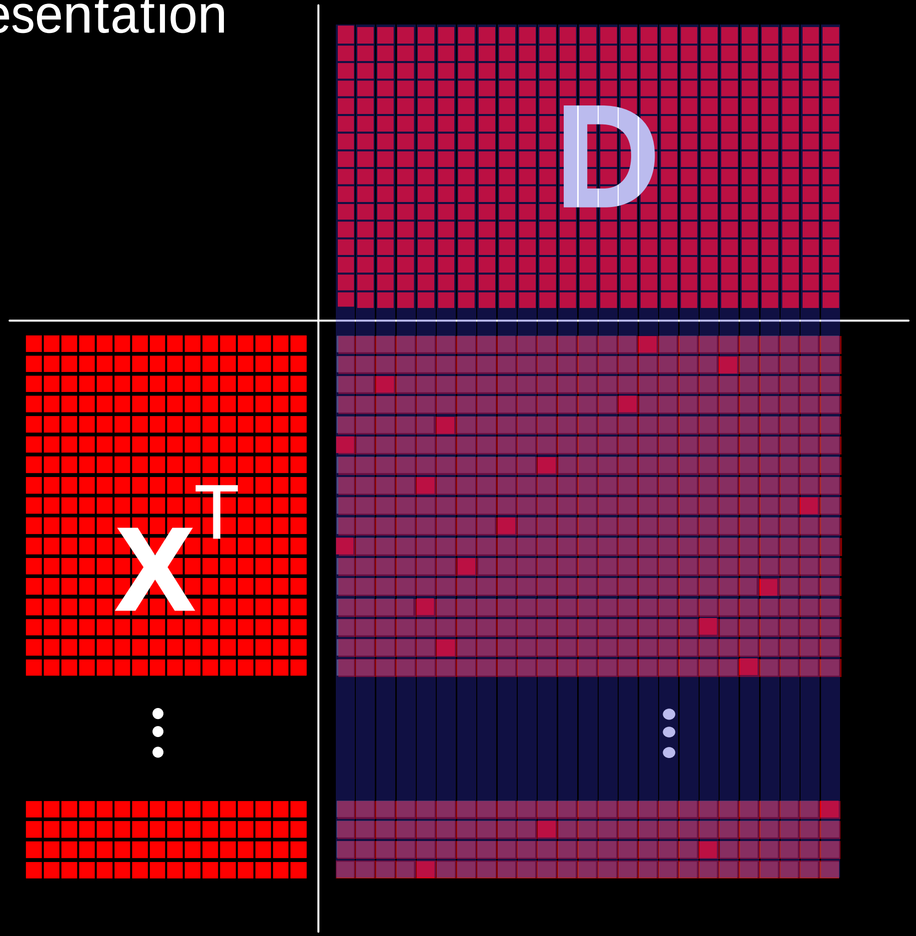
K–Means For Clustering

Clustering: An extreme sparse representation

Initialize
D

Sparse Coding
Nearest Neighbor

Dictionary
Update
Column-by-Column by
Mean computation over the relevant examples



The K–SVD Algorithm – General

Aharon, Elad, & Bruckstein (`04)

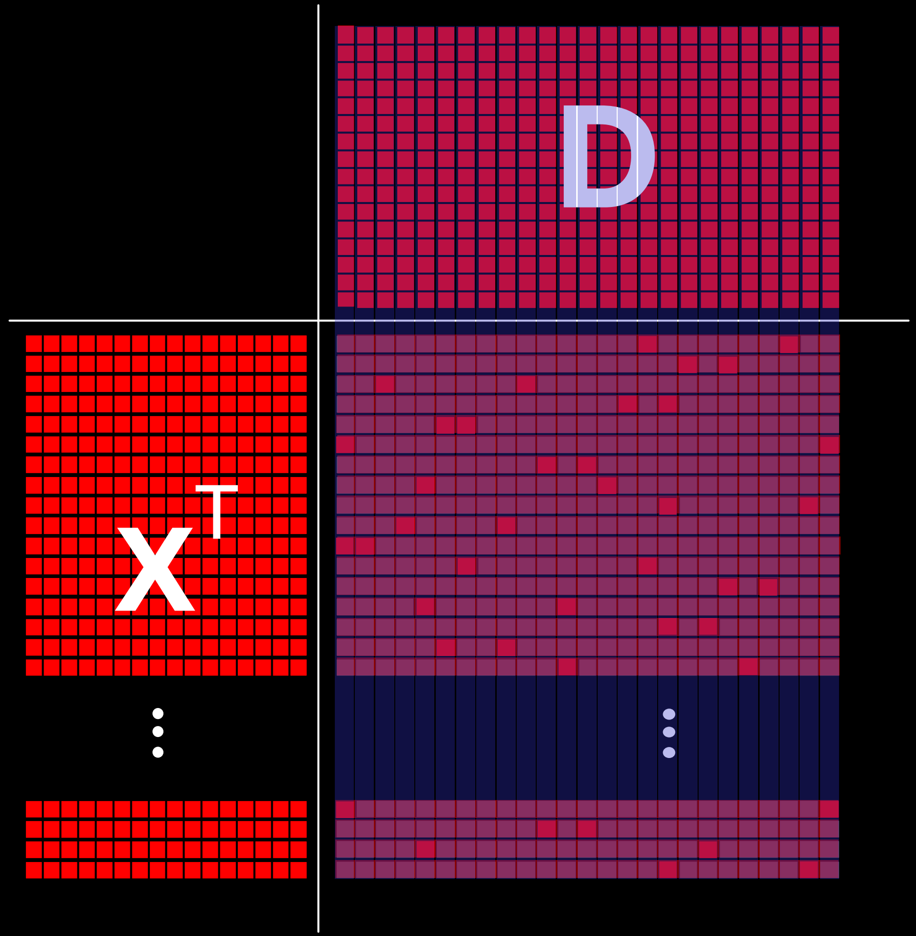
Initialize
D

Sparse Coding

Use Matching Pursuit

Dictionary Update

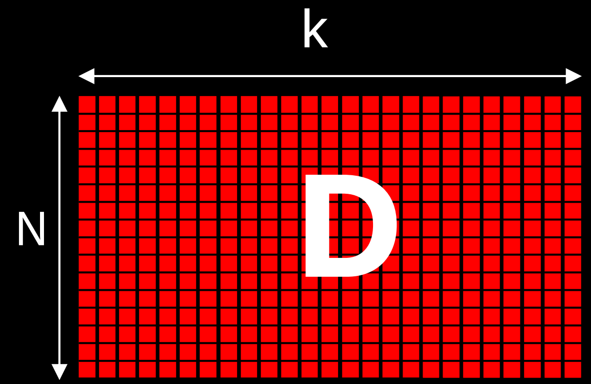
Column-by-Column by
SVD computation over the relevant examples



Part III:
Combining It All

From Local to Global Treatment

- The K-SVD algorithm is reasonable for low-dimension signals (N in the range 10-400). As N grows, the complexity and the memory requirements of the K-SVD become prohibitive.



- So, how should large images be handled?
- The solution: Force shift-invariant sparsity - on each patch of size N-by-N (N=8) in the image, including overlaps [Roth & Black (`05)].

$$\hat{\underline{x}} = \underset{\underline{x},\{\underline{\alpha}_{ij}\}_{ij}}{\text{ArgMin}} \; \frac{1}{2}\left\|\underline{x}-\underline{y}\right\|_2^2 + \mu\sum_{ij}\left\|\mathbf{R}_{ij}\underline{x} - \mathbf{D}\underline{\alpha}_{ij}\right\|_2^2 \quad \text{s.t.} \quad \left\|\underline{\alpha}_{ij}\right\|_0^0 \le L$$

Extracts a patch in the ij location

Our prior

What Data to Train On?

Option 1:

- Use a database of images,
- We tried that, and it works fine (~0.5-1dB below the state-of-the-art).

Option 2:

- Use the corrupted image itself !!
- Simply sweep through all patches of size N-by-N (overlapping blocks),
- Image of size 1000^2 pixels ➡ ~10^6 examples to use – more than enough.
- This works much better!

Block-Coordinate-Relaxation

$$\hat{\underline{x}} = \underset{\underline{x},\{\underline{\alpha}_{ij}\}_{ij},\mathbf{D}}{\text{ArgMin}} \; \frac{1}{2}\left\|\underline{x}-\underline{y}\right\|_2^2 + \mu\sum_{ij}\left\|\mathbf{R}_{ij}\underline{x}-\mathbf{D}\underline{\alpha}_{ij}\right\|_2^2 \;\; \text{s.t.} \;\; \left\|\underline{\alpha}_{ij}\right\|_0^0 \le L$$

$\underline{x}=\underline{y}$ and $\mathbf{D}$ known

Compute α_{ij} per patch

$$\underline{\alpha}_{ij} = \underset{\underline{\alpha}}{\text{Min}}\left\|\mathbf{R}_{ij}\underline{x}-\mathbf{D}\underline{\alpha}\right\|_2^2$$

$$\text{s.t.} \;\; \left\|\underline{\alpha}\right\|_0^0 \le L$$

using the matching pursuit

$\underline{x}$ and α_{ij} known

Compute **D** to minimize

$$\underset{\underline{\alpha}}{\text{Min}}\sum_{ij}\left\|\mathbf{R}_{ij}\underline{x}-\mathbf{D}\underline{\alpha}\right\|_2^2$$

using SVD, updating one column at a time

D and α_{ij} known

Compute $\underline{x}$ by

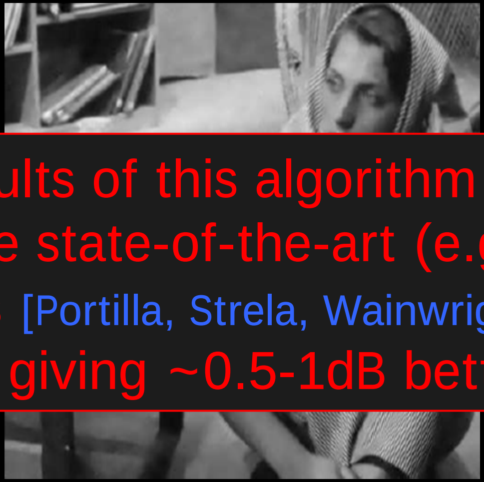
$$\underline{x} = \left[\mathbf{I}+\mu\sum_{ij}\mathbf{R}_{ij}^T\mathbf{R}_{ij}\right]^{-1}\left[\underline{y}+\mu\sum_{ij}\mathbf{R}_{ij}^T\mathbf{D}\underline{\alpha}_{ij}\right]$$

which is a simple averaging of shifted patches

Complexity of this algorithm: $O(N^2 \times L \times \text{Iterations})$ per pixel. For N=8, L=1, and 10 iterations, we need 640 operations per pixel.

Denoising Results

Source

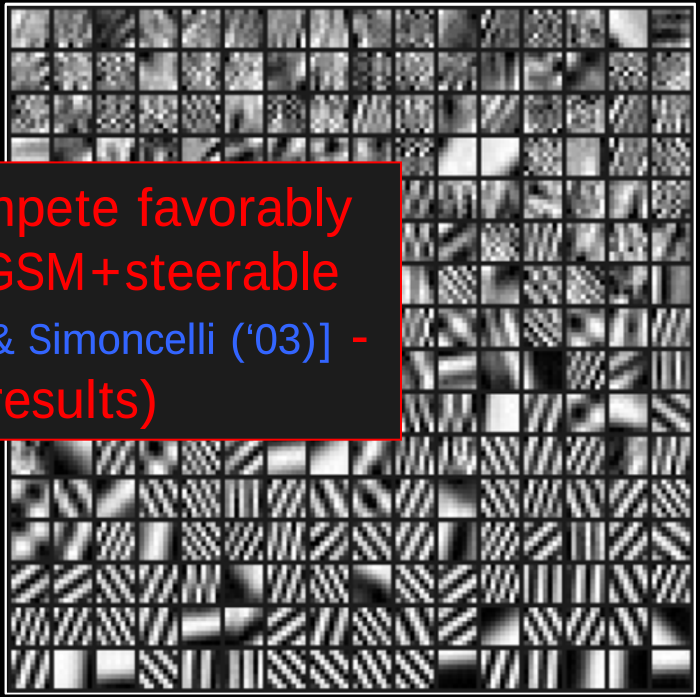


The results of this algorithm compete favorably with the state-of-the-art (e.g., GSM+steerable wavelets [Portilla, Strela, Wainwright, & Simoncelli ('03)] - giving ~0.5-1dB better results)

Result 30.829dB

Noisy image

$$\sigma = 20$$



The obtained dictionary after 10 iterations

Today We Have Seen …

An energy minimization method

A Bayesian (MAP) point of view

$$\hat{\underline{x}} = \underset{\underline{x},\{\underline{\alpha}_{ij}\}_{ij},\mathbf{D}}{\mathrm{ArgMin}} \left\|\underline{x}-\underline{y}\right\|_2^2 + \mu\sum_{ij}\left\|\mathbf{R}_{ij}\underline{x}-\mathbf{D}\underline{\alpha}_{ij}\right\|_2^2$$

$$\text{Subject to } \left\|\underline{\alpha}_{ij}\right\|_0^0 \le L$$

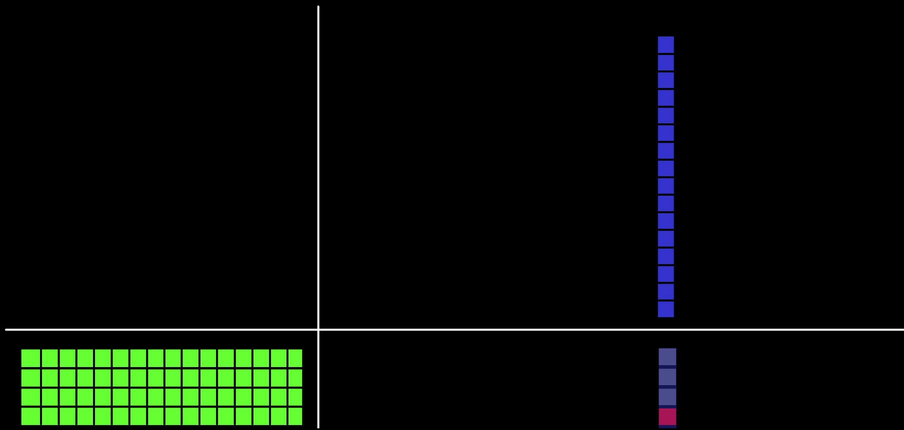
Getting a relatively simple and highly effective denoising algorithm

Using examples from the noisy image itself to learn the image prior

Using an image prior based on sparsity and redundancy

More on this in http://www.cs.technion.ac.il/~elad

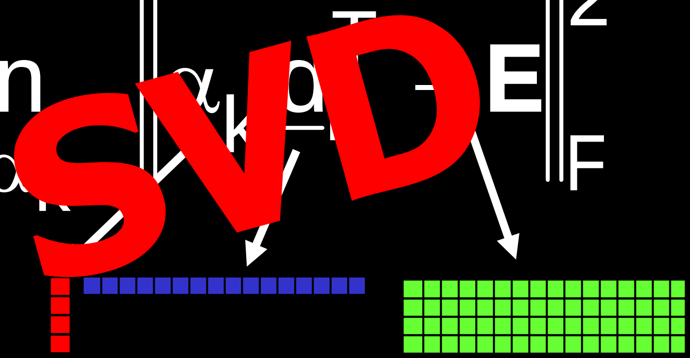
K–SVD: Dictionary Update Stage



We refer only to the examples that use the column $\underline{d}_k$

We should solve:

$$\underset{\underline{d}_k,\alpha_k}{\text{Min}} \left\| \alpha_k \underline{d}_k^T - \mathbf{E} \right\|_F^2$$



Fixing all **A** and **D** apart from the kth column, and seek both $\underline{d}_k$ and the kth column in **A** to better fit the **residual**!

K–SVD: Sparse Coding Stage

$$\underset{\mathbf{A}}{\text{Min}} \sum_{j=1}^{P} \left\| \mathbf{D}\underline{\alpha}_j - \underline{x}_j \right\|_2^2 \quad \text{s.t.} \quad \forall j, \left\| \underline{\alpha}_j \right\|_p^p \le L$$

D is known!
For the j^{th} item
we solve

$$\underset{\underline{\alpha}}{\text{Min}} \left\| \mathbf{D}\underline{\alpha} - \underline{x}_j \right\|_2^2 \quad \text{s.t.} \quad \left\| \underline{\alpha} \right\|_p^p \le L$$

Solved by Matching Pursuit