Sparse & Redundant Signal Representation, and its Role in Image Processing

Michael Elad
The CS Department
The Technion – Israel Institute of technology
Haifa 32000, Israel

Wave 2006 – Wavelet and Applications
Ecole Polytechnique Federale de Lausanne (EPFL)
July 10^{th}–14^{th}, 2006

$\mathbf{D}\underline{\alpha} = \underline{x}$

Today's Talk is About

Sparsity

and

Redundancy

We will try to show today that

- Sparsity & Redundancy can be used to design new/renewed & powerful signal/image processing tools (e.g., transforms, priors, models, compression, ...),
- The obtained machinery works very well – we will show these ideas deployed to image processing **applications**.

Agenda

1. **A Visit to *Sparseland***

   Motivating Sparsity & Overcompleteness

2. Problem 1: Transforms & Regularizations

   How & why should this work?

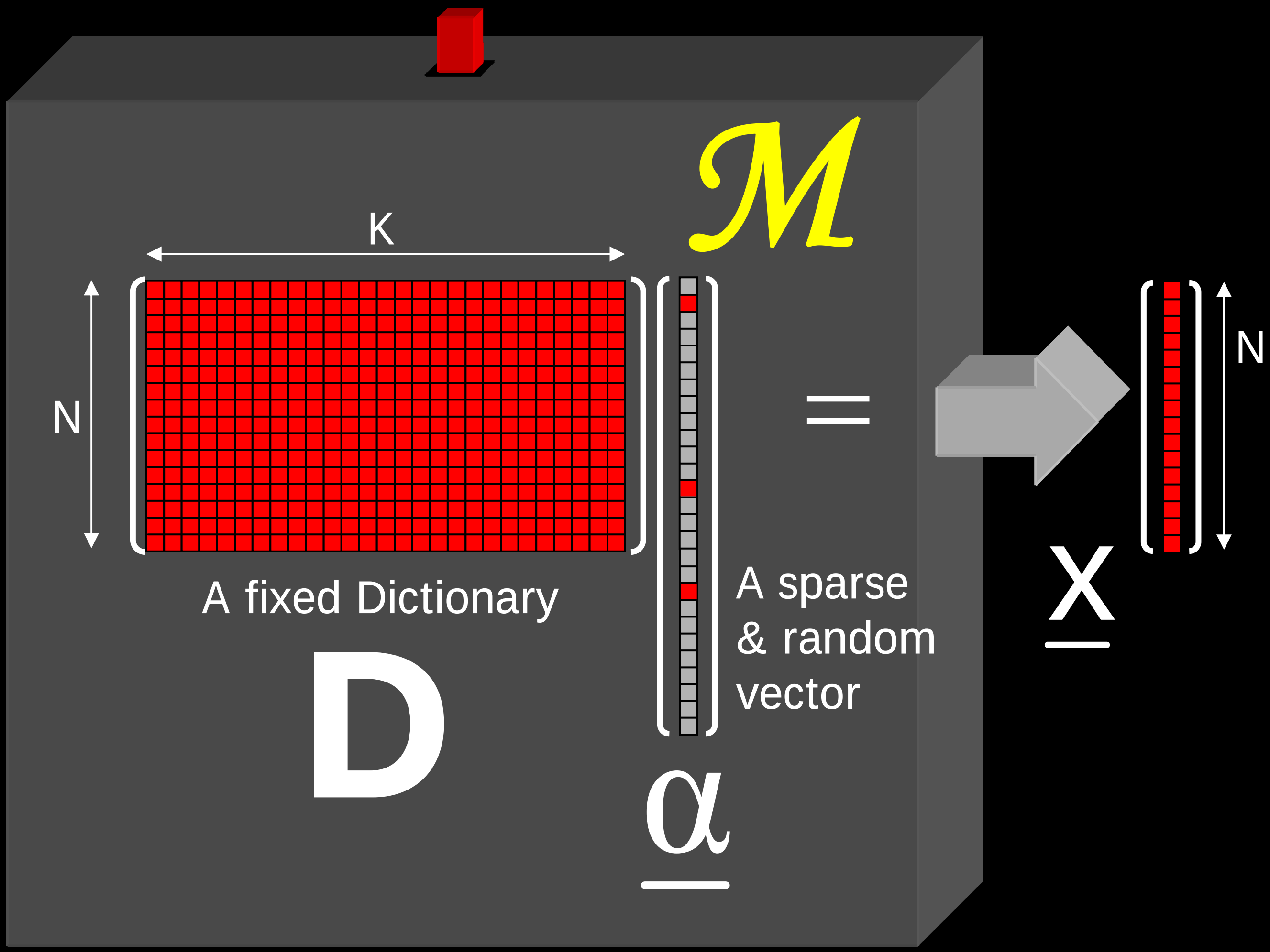
3. Problem 2: What About **D**?

   The quest for the origin of signals

4. Problem 3: Applications

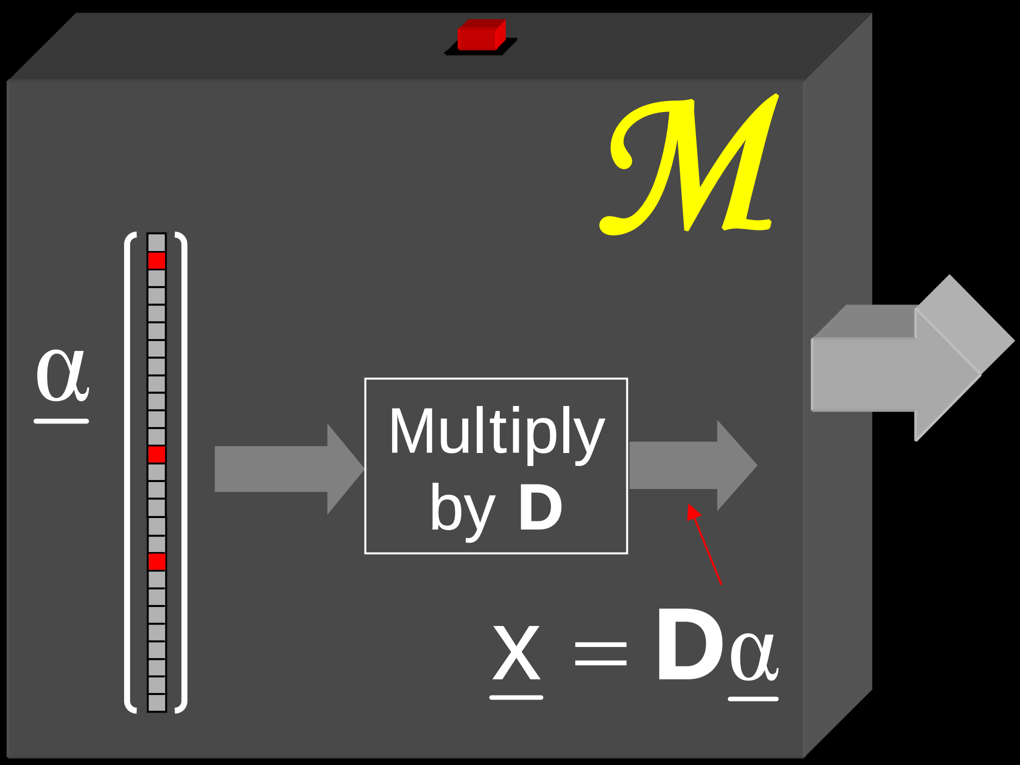
   Image filling in, denoising, separation, compression, ...

Generating Signals in *Sparseland*



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

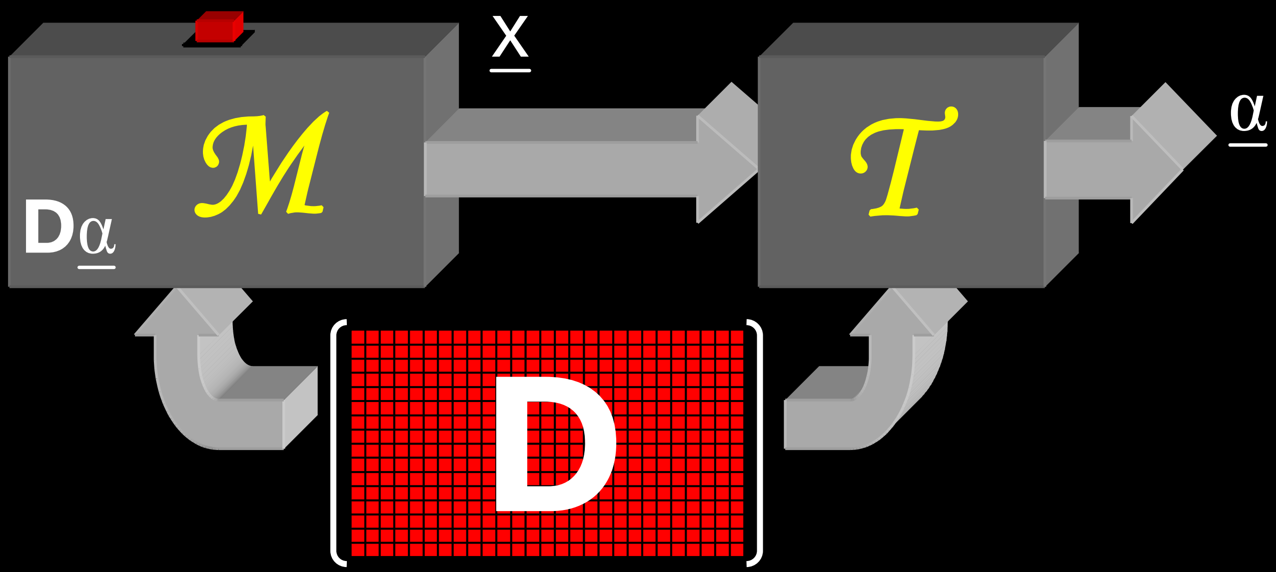
Sparseland Signals Are Special



- **Simple:** Every generated signal is built as a linear combination of **few** atoms from our dictionary **D**
- **Rich:** A general model: the obtained signals are a special type mixture-of-Gaussians (or Laplacians).

Transforms in *Sparseland* ?

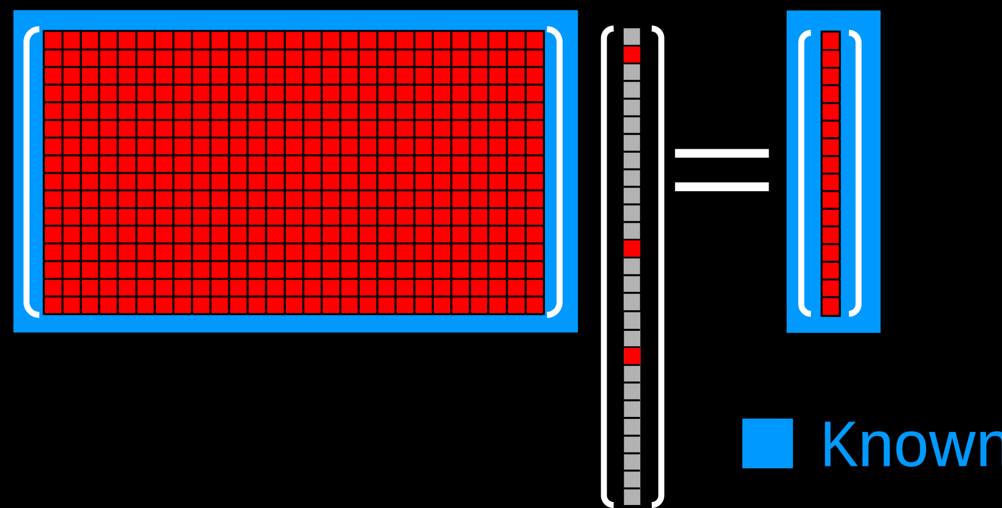
- Assume that $\underline{x}$ is known to emerge from $\mathcal{M}$.
- We desire simplicity, independence, and expressiveness.
- How about "Given $\underline{x}$, find the $\underline{\alpha}$ that generated it in $\mathcal{M}$" ?



So, In Order to Transform …

We need to solve an under-determined linear system of equations:

$$\mathbf{D}\underline{\alpha} = \underline{x}$$

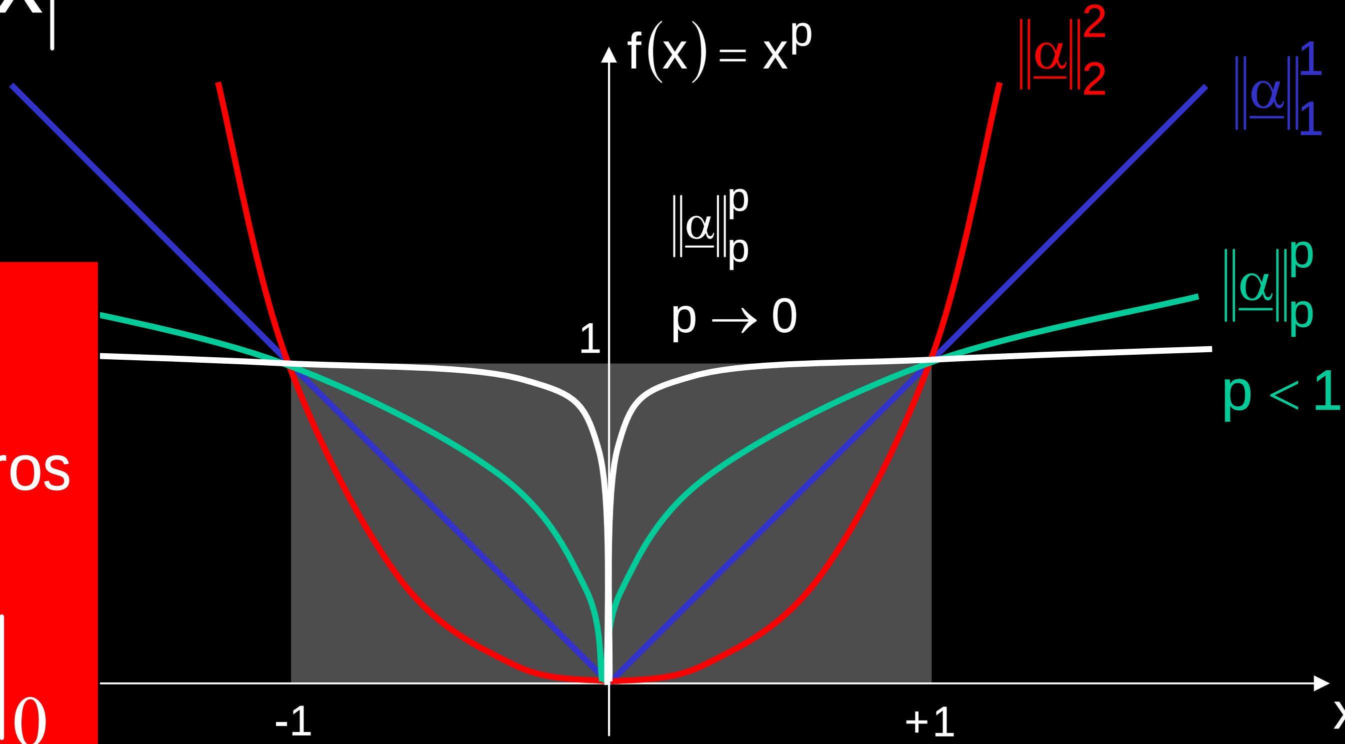


- Among all (infinitely many) possible solutions we want the **sparsest** !!
- We will measure sparsity using the L_0 norm:

$$\rightarrow \quad \left\|\underline{\alpha}\right\|_0$$

Measure of Sparsity?

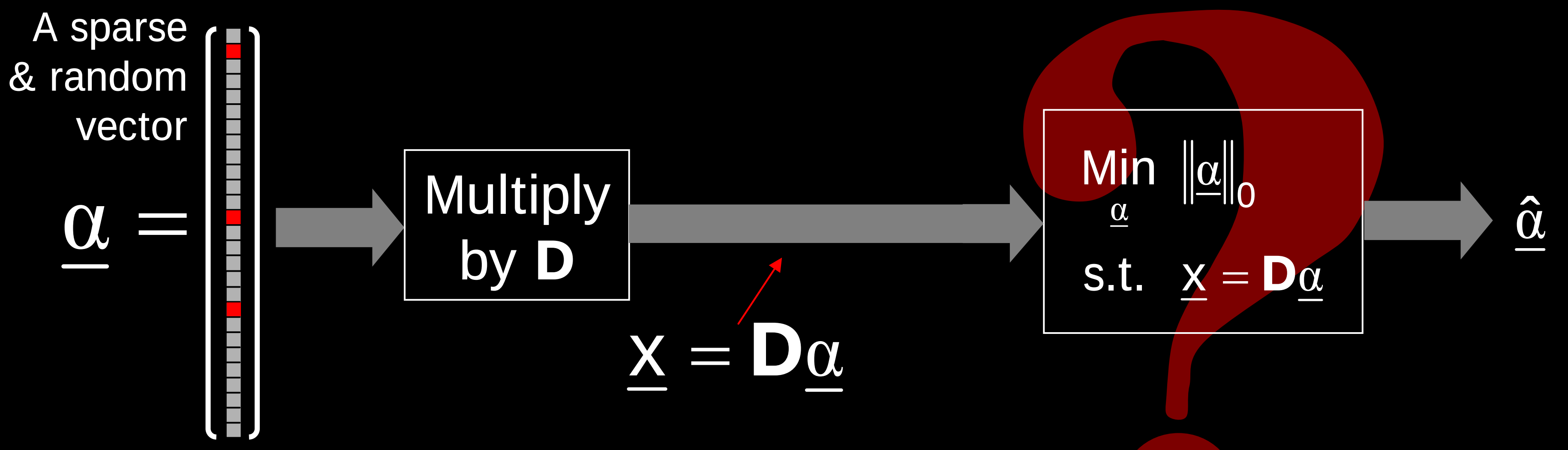
$$\|\underline{x}\|_p^p = \sum_{j=1}^{k} |x|^p$$



As $p \to 0$ we get a count of the non-zeros in the vector

$$\Rightarrow \|\underline{\alpha}\|_0$$

Signal's Transform in *Sparseland*

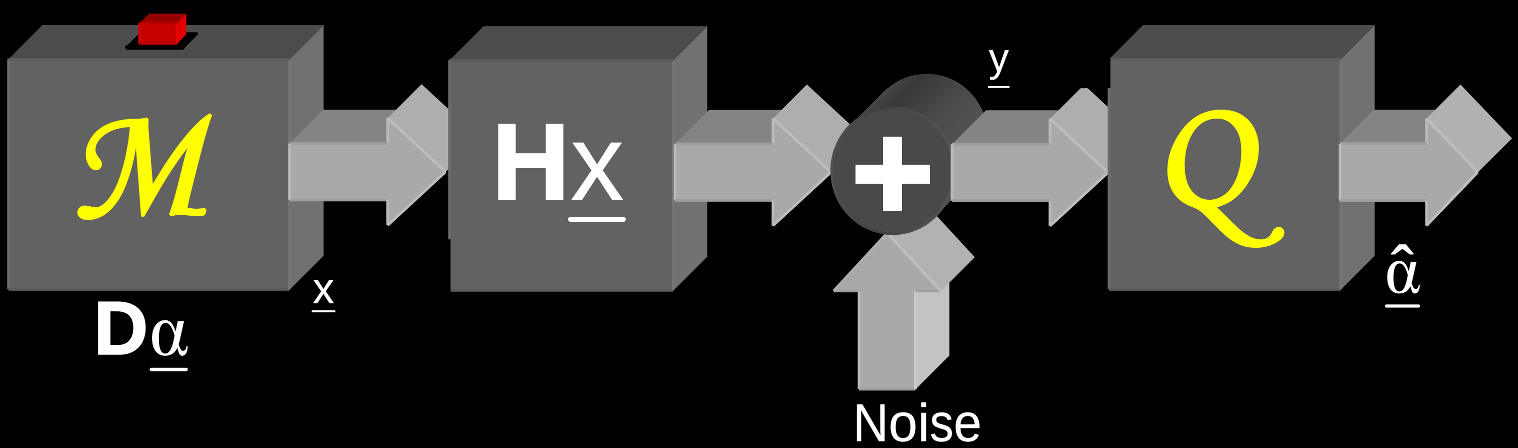


4 Major Questions

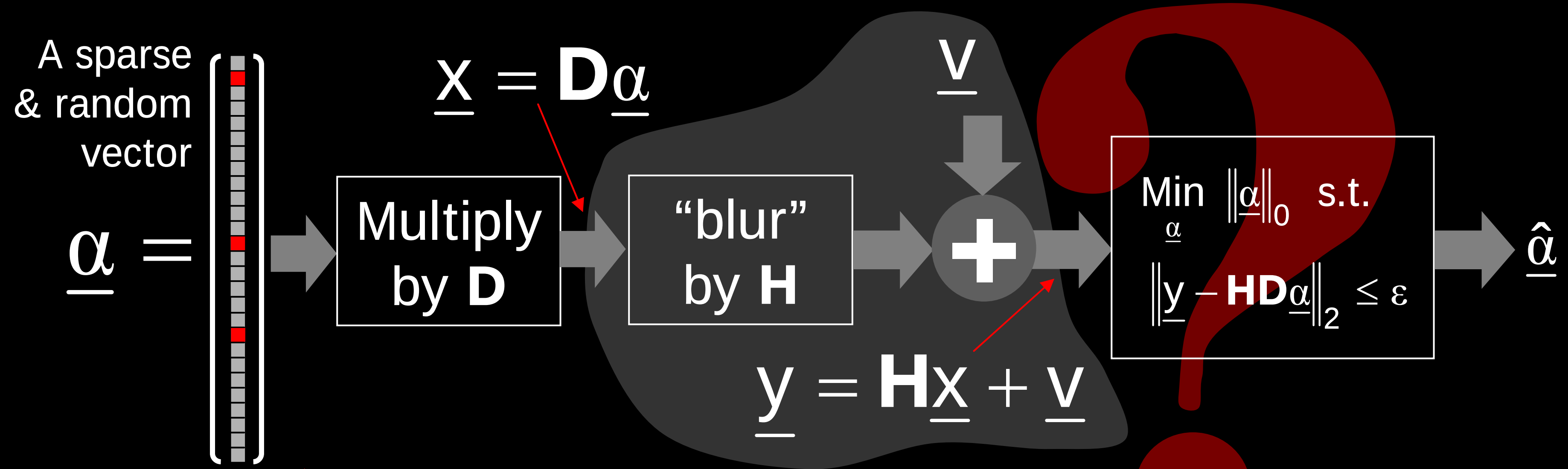
- Is $\underline{\hat{\alpha}} = \underline{\alpha}$? Under which conditions?
- Are there practical ways to get $\underline{\hat{\alpha}}$?
- How effective are those ways?
- How would we get **D**?

Inverse Problems in *Sparseland* ?

- Assume that $\underline{x}$ is known to emerge from $\mathcal{M}$.
- Suppose we observe $\underline{y} = \mathbf{H}\underline{x} + \underline{v}$, a “blurred” and noisy version of $\underline{x}$ with $\|\underline{v}\|_2 \leq \varepsilon$. How will we recover $\underline{x}$?
- How about “find the $\underline{\alpha}$ that generated the $\underline{x}$ ...” again?



Inverse Problems in *Sparseland* ?



4 Major Questions (again!)

- Is $\hat{\underline{\alpha}} = \underline{\alpha}$? How far can it go?
- Are there practical ways to get $\hat{\underline{\alpha}}$?
- How effective are those ways?
- How would we get **D**?

Back Home … Any Lessons?

Several recent trends worth looking at:

- JPEG to JPEG2000 - From (L_2-norm) KLT … t and non-linear approximation → S…
- From W… Fourier) to L_1. (… Sparsity.
- From un… -invariance, steerable… pleteness.
- Approxim… ximation → Spars… pleteness.
- ICA and related models → Independence and Sparsity.

Sparseland **is HERE !**

To Summarize so far ...

The *Sparseland* model $\mathcal{M}$ for signals is interesting since it is rich and yet every signal has a simple description

Who cares?

We do! this model is relevant to us just as well.

So, what are the implications?

We need to solve (or approximate the solution of) a linear system with more unknowns than equations, while finding the sparsest possible solution

Sparsity? looks complicated

(a) Practical solvers?
(b) How will we get **D**?
(c) Applications?

Agenda

1. A Visit to *Sparseland*

   Motivating Sparsity & Overcompleteness

2. **Problem 1: Transforms & Regularizations**

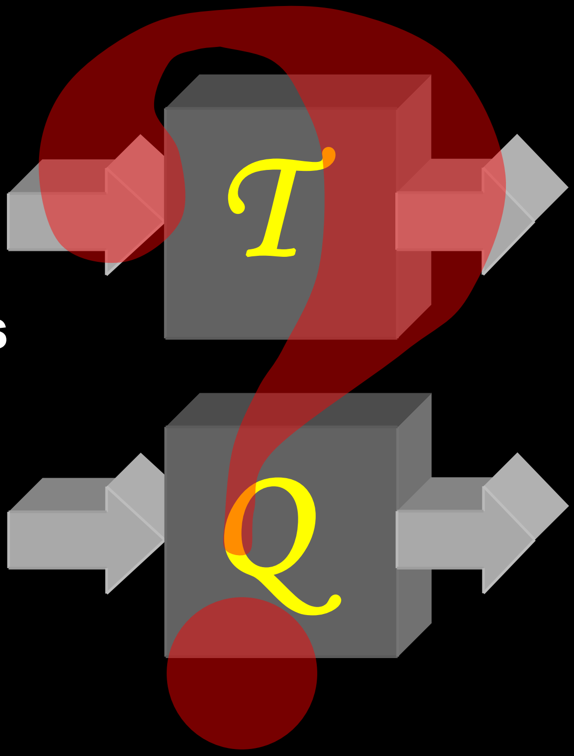
   How & why should this work?

3. Problem 2: What About **D**?

   The quest for the origin of signals

4. Problem 3: Applications

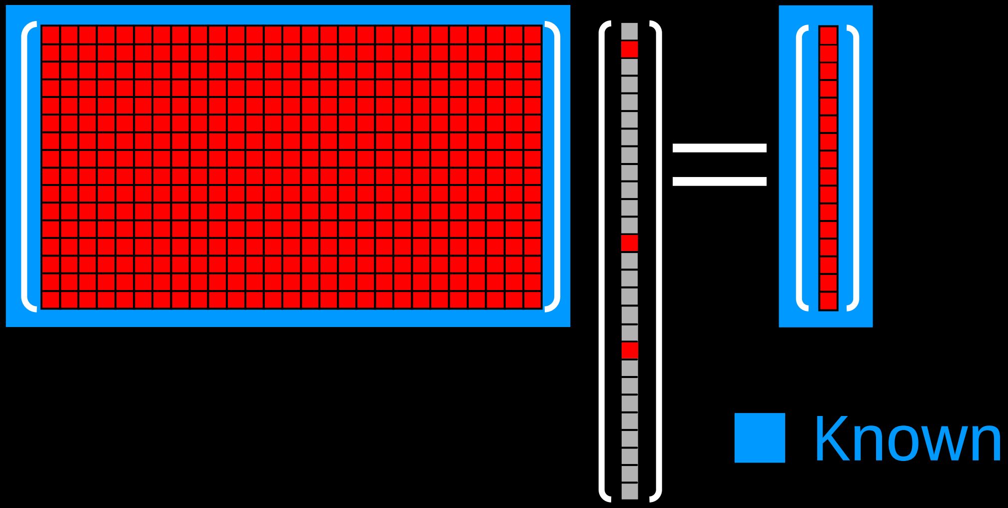
   Image filling in, denoising, separation, compression, ...



Lets Start with the Transform …

Our dream for Now: Find the sparsest solution of

$$\mathbf{D}\underline{\alpha} = \underline{x}$$

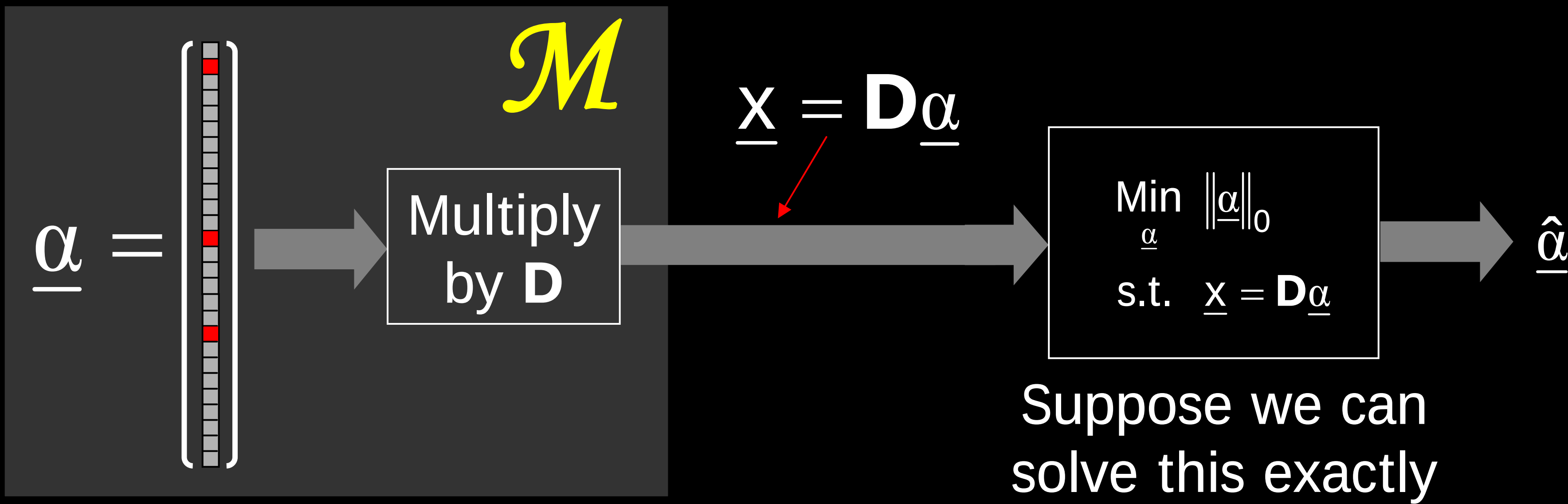


Put formally,

$$\min_{\underline{\alpha}} \left\|\underline{\alpha}\right\|_0 \quad \text{s.t.} \quad \underline{x} = \mathbf{D}\underline{\alpha}$$

known

Question 1 – Uniqueness?



Why should we necessarily get $\hat{\underline{\alpha}} = \underline{\alpha}$?

It might happen that eventually $\|\hat{\underline{\alpha}}\|_0 < \|\underline{\alpha}\|_0$.

Matrix "Spark"

Definition: Given a matrix **D**, σ=Spark{**D**} is the smallest number of columns that are linearly dependent. *

Donoho & E. ('02)

Example:

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

Rank = 4

Spark = 3

* In tensor decomposition, Kruskal defined something similar already in 1989.

Uniqueness Rule

Suppose this problem has been solved somehow

$$\underset{\underline{\alpha}}{\text{Min}} \|\underline{\alpha}\|_0 \quad \text{s.t.} \quad \underline{x} = \mathbf{D}\underline{\alpha}$$

Uniqueness

If we found a representation that satisfy

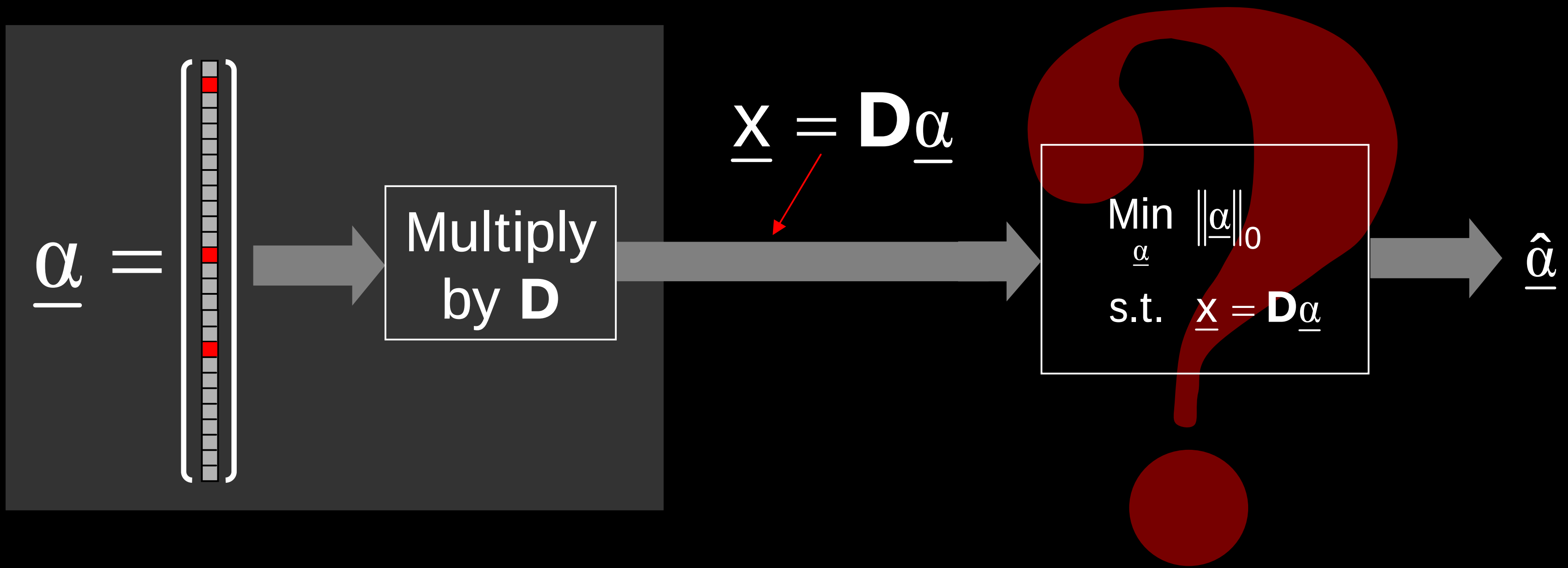
$$\frac{\sigma}{2} > \|\underline{\alpha}\|_0$$

Then necessarily it is unique (the sparsest).

Donoho & E. ('02)

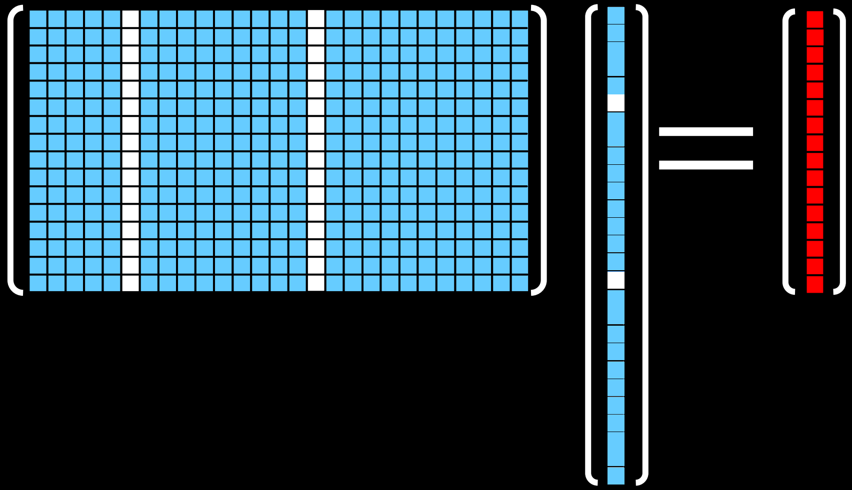
This result implies that if $\mathcal{M}$ generates signals using "sparse enough" $\underline{\alpha}$, the solution of the above will find it exactly.

Question 2 – Practical P_0 Solver?



Are there reasonable ways to find $\hat{\underline{\alpha}}$?

Matching Pursuit (MP) Mallat & Zhang (1993)



- The MP is a greedy algorithm that finds one atom at a time.
- Step 1: find the one atom that best matches the signal.
- Next steps: given the previously found atoms, find the next **<u>one</u>** to best fit ...
- The Orthogonal MP (OMP) is an improved version that re-evaluates the coefficients after each round.

Basis Pursuit (BP) Chen, Donoho, & Saunders (1995)

Instead of solving

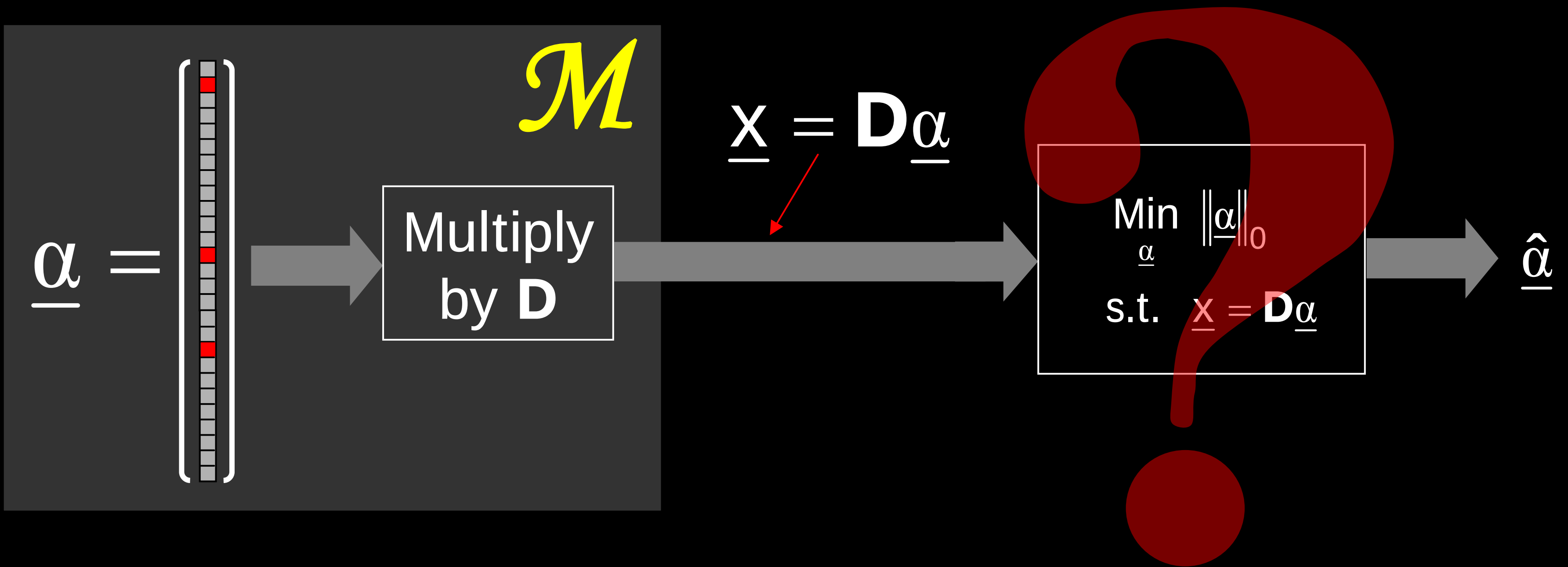
$$\underset{\underline{\alpha}}{\text{Min}} \left\| \underline{\alpha} \right\|_0 \quad \text{s.t.} \quad \underline{x} = \mathbf{D}\underline{\alpha}$$

Solve Instead

$$\underset{\underline{\alpha}}{\text{Min}} \left\| \underline{\alpha} \right\|_1 \quad \text{s.t.} \quad \underline{x} = \mathbf{D}\underline{\alpha}$$

- The newly defined problem is convex (linear programming).
- Very efficient solvers can be deployed:
  - Interior point methods [Chen, Donoho, & Saunders (`95)] ,
  - Sequential shrinkage for union of ortho-bases [Bruce et.al. (`98)],
  - Iterated shrinkage [Figuerido & Nowak (`03), Daubechies, Defrise, & Demole ('04), E. (`05), E., Matalon, & Zibulevsky (`06)].

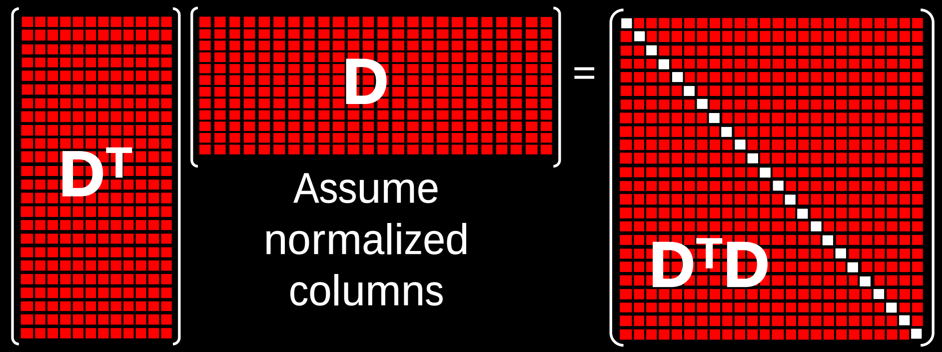
Question 3 – Approx. Quality?



How effective are the MP/BP in finding $\hat{\underline{\alpha}}$?

Evaluating the "Spark"

- Compute



- The Mutual Coherence M is the largest off-diagonal entry in absolute value.

- The Mutual Coherence is a property of the dictionary (just like the "Spark"). In fact, the following relation can be shown:

$$\sigma \geq 1 + \frac{1}{M}$$

BP and MP Equivalence

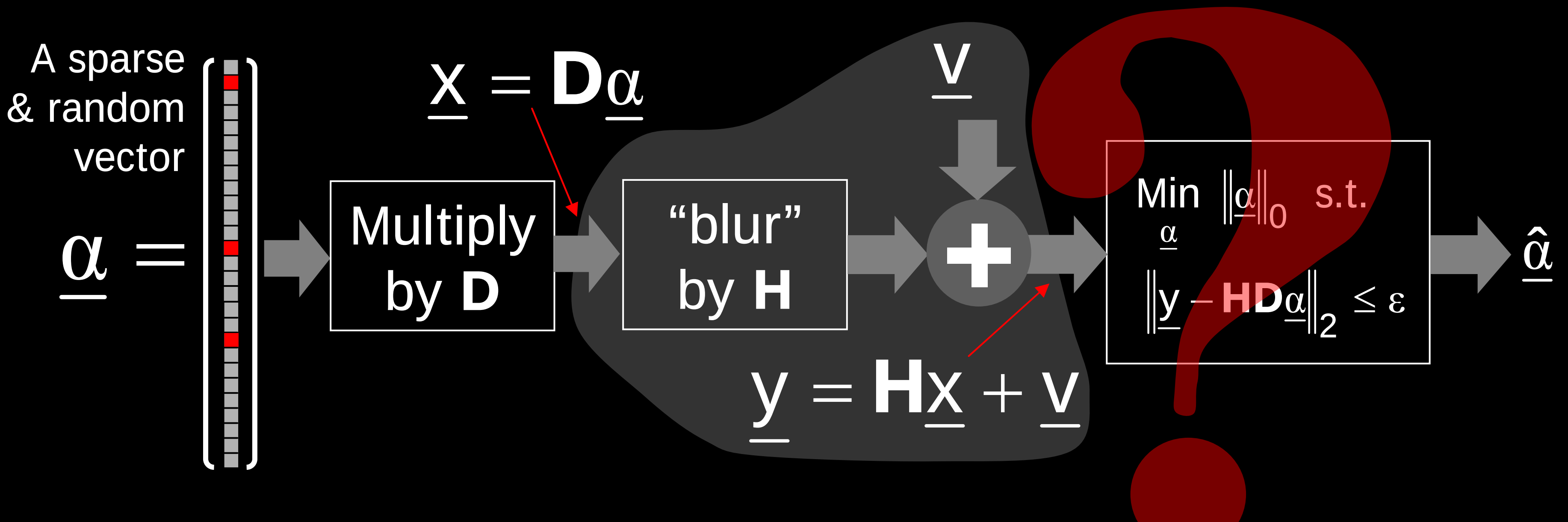
Equivalence

Donoho & E. ('02)
Gribonval & Nielsen ('03)
Tropp ('03)
Temlyakov ('03)

Given a signal $\underline{x}$ with a representation $\underline{x} = \mathbf{D}\underline{\alpha}$,
Assuming that $\|\underline{\alpha}\|_0 < 0.5(1 + 1/M)$, BP and MP are
Guaranteed to find the sparsest solution.

- MP and BP are different in general (hard to say which is better).
- The above result corresponds to the worst-case.
- Average performance results are available too, showing much better bounds [Donoho (`04), Candes et.al. (`04), Tanner et.al. (`05), Tropp et.al. (`06)].

What About Inverse Problems?



- We had similar questions regarding uniqueness, practical solvers, and their efficiency.
- It turns out that similar answers are applicable here due to several recent works [Donoho, E. and Temlyakov (`04), Tropp (`04), Fuchs (`04), Gribonval et. al. (`05)].

To Summarize so far …

The *Sparseland* model for signals is relevant to us. We can design transforms and priors based on it

Find the sparsest solution?

Use pursuit Algorithms

Why works so well?

A sequence of works during the past 3-4 years gives theoretic justifications for these tools behavior

What next?

(a) How shall we find **D**?

(b) Will this work for applications? Which?

Agenda

1. A Visit to *Sparseland*

   Motivating Sparsity & Overcompleteness

2. Problem 1: Transforms & Regularizations

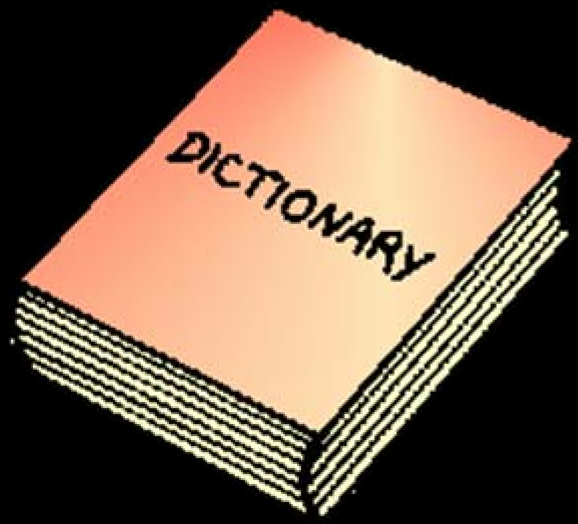
   How & why should this work?

3. **Problem 2: What About D?**

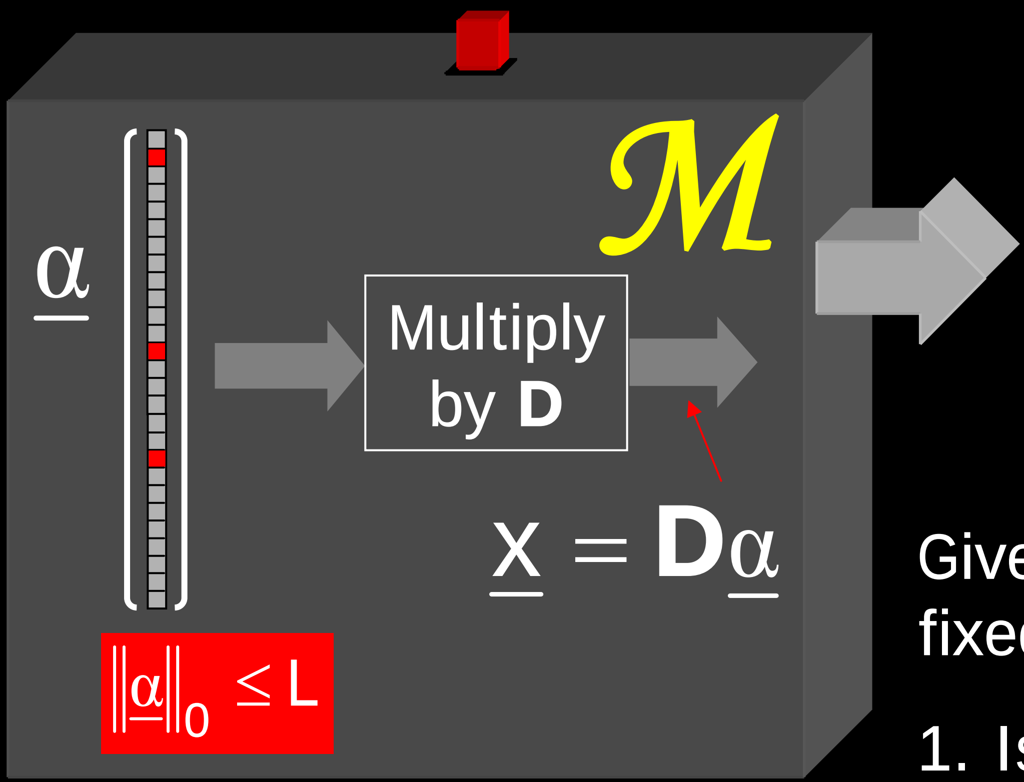
   The quest for the origin of signals

4. Problem 3: Applications

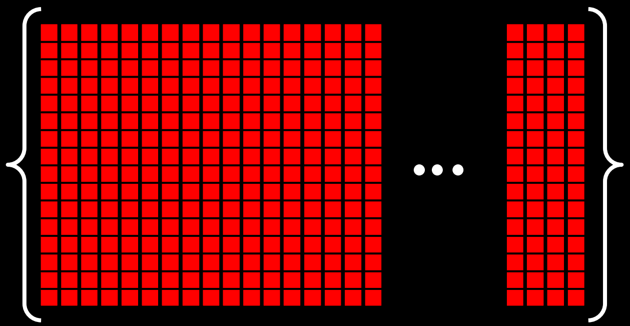
   Image filling in, denoising, separation, compression, ...



Problem Setting



$$\left\{ \underline{x}_j \right\}_{j=1}^{P}$$



Given these P examples and a fixed size [N×K] dictionary **D**:

1. Is **D** unique?
2. How would we find **D**?

Uniqueness?

Uniqueness → If $\left\{\underline{x}_j\right\}_{j=1}^{P}$ is rich enough* and if

$$L < \frac{\mathrm{Spark}\{\mathbf{D}\}}{2}$$

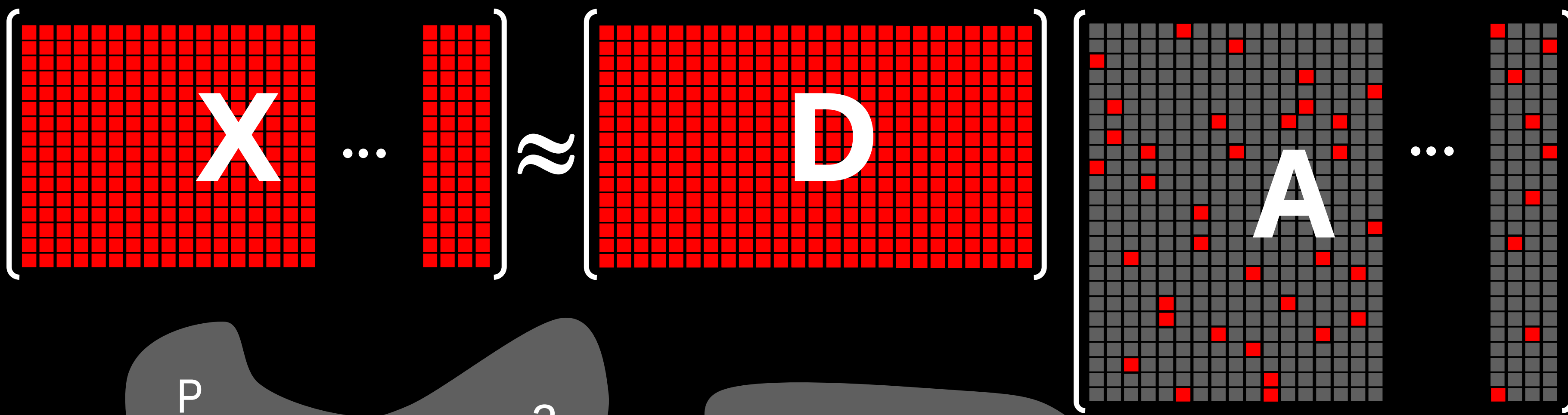
then **D** is unique.

Aharon, E., & Bruckstein (`05)

Comments:

- "Rich Enough": The signals from $\mathcal{M}$ could be clustered to $\binom{K}{L}$ groups that share the same support. At least L+1 examples per each are needed.
- This result is proved constructively, but the number of examples needed to pull this off is huge – we will show a far better method next.
- A parallel result that takes into account noise is yet to be constructed.

Practical Approach – Objective



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

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

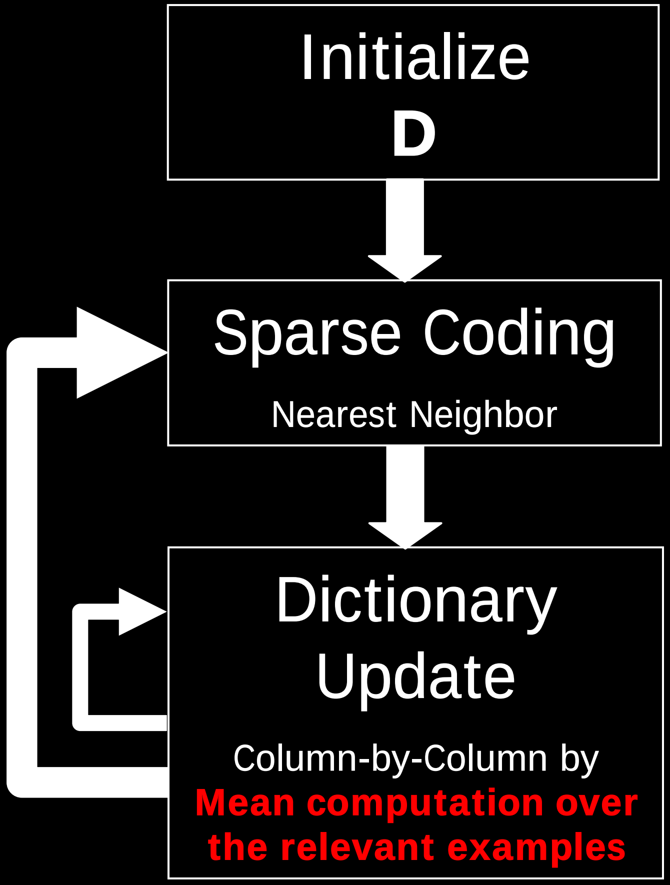
Each example has a sparse representation with no more than L atoms

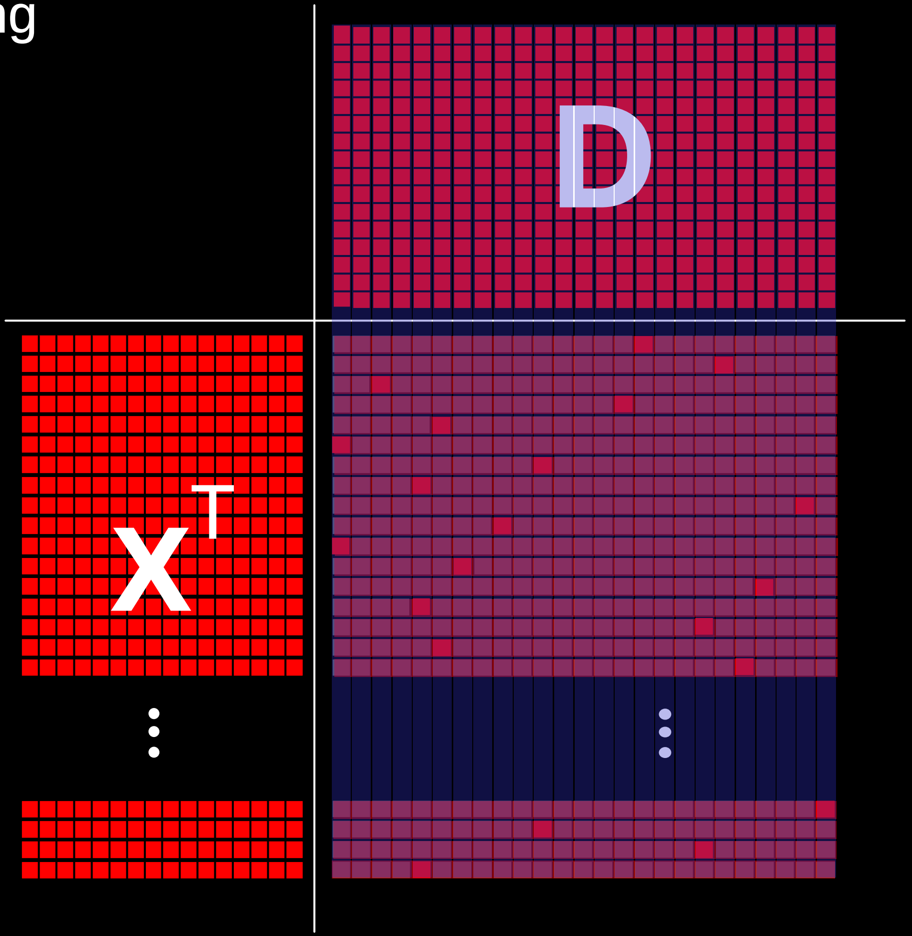
(n,K,L are assumed known, **D** has norm. columns)

Field & Olshausen (96')
Engan et. al. (99')
Lewicki & Sejnowski (00')
Cotter et. al. (03')
Gribonval et. al. (04')

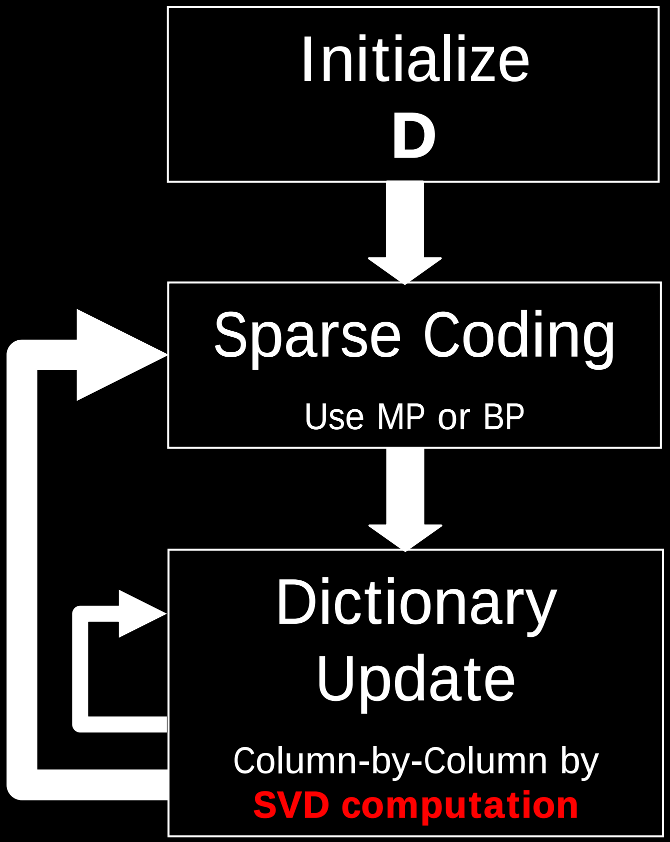
K–Means For Clustering

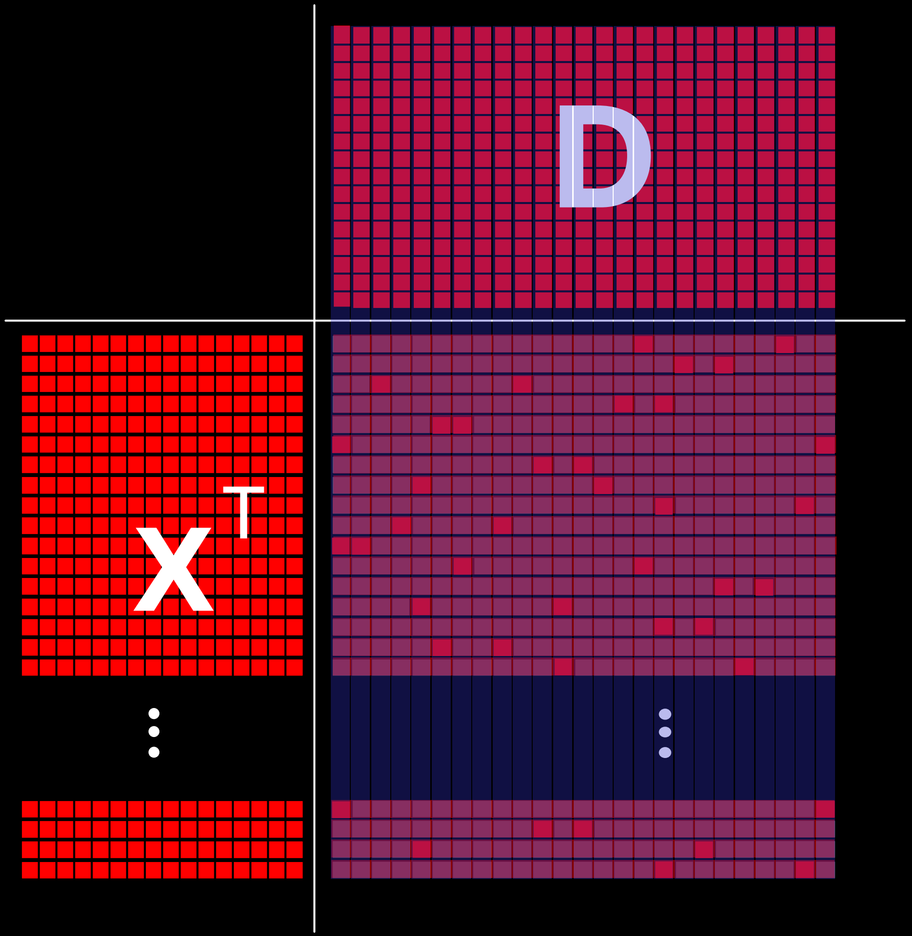
Clustering: An extreme sparse coding





The K–SVD Algorithm – General





Aharon, E., & Bruckstein (`04)

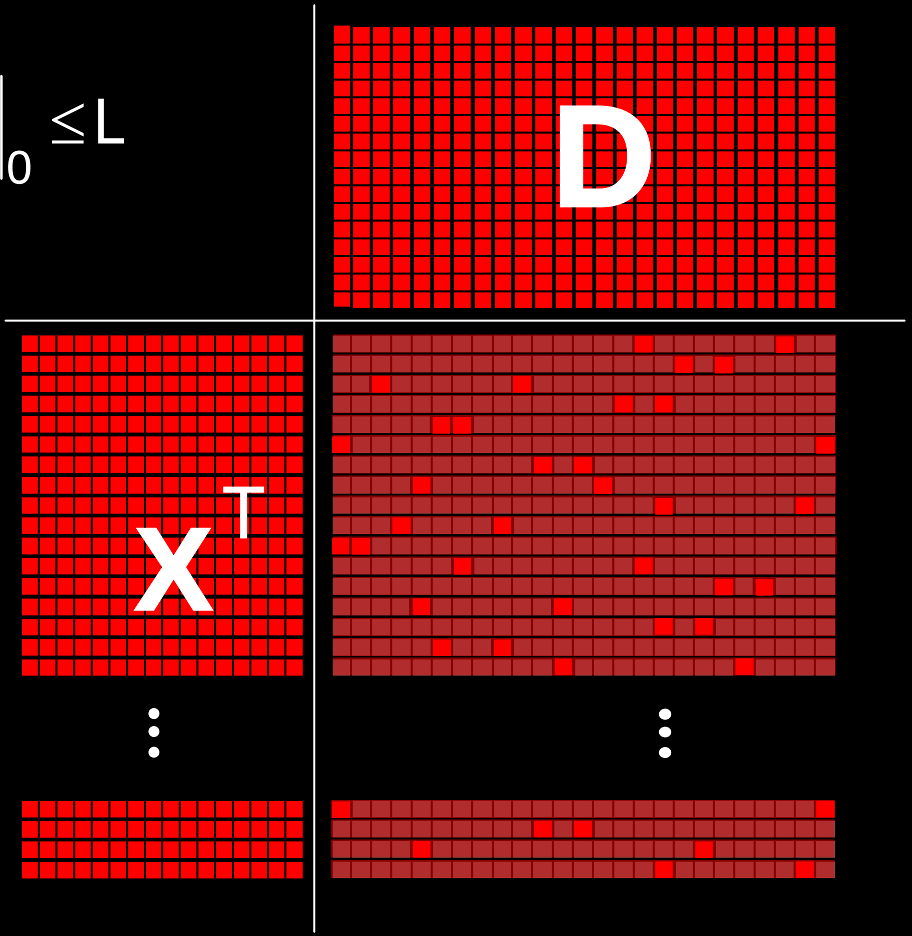
K–SVD Sparse Coding Stage

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

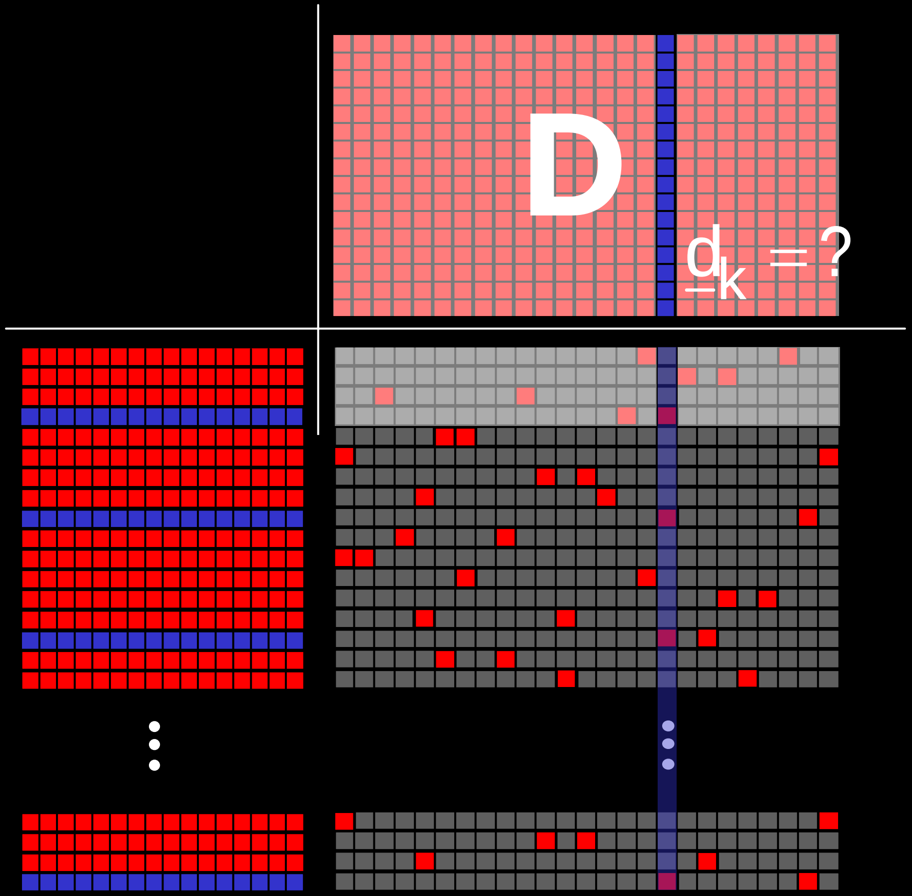
For the jth example we solve

$$\underset{\underline{\alpha}}{\text{Min}} \quad \left\| \mathbf{D}\underline{\alpha} - \underline{x}_j \right\|_2^2 \quad \text{s.t.} \quad \left\| \underline{\alpha} \right\|_0 \le L$$

Pursuit Problem !!!



K–SVD Dictionary Update Stage



G_k : The examples in $\{\underline{x}_j\}_{j=1}^{P}$ that use the column $\underline{d}_k$.

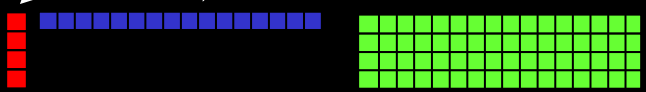
The content of $\underline{d}_k$ influences only the examples in G_k.

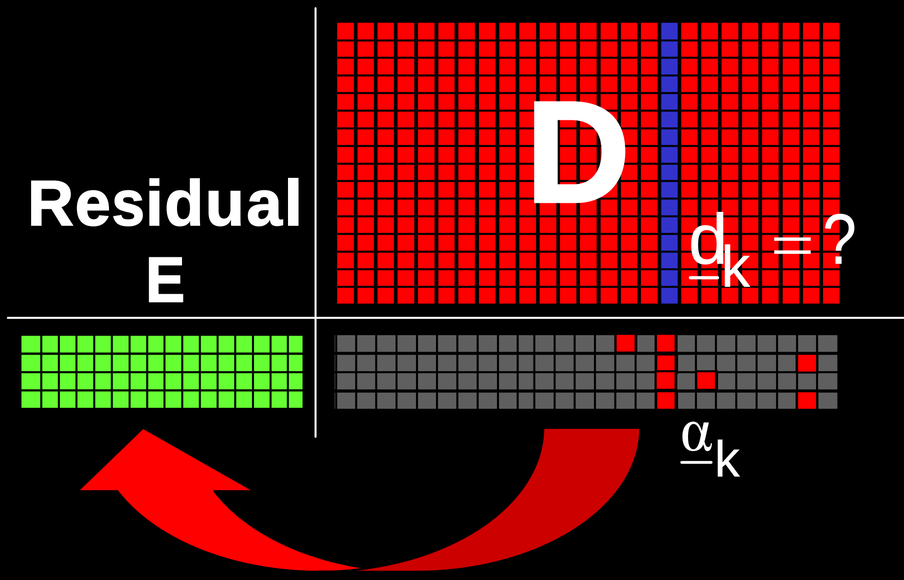
Let us fix all **A** apart from the k^{th} column and seek both $\underline{d}_k$ and the k^{th} column to better fit the **residual**!

K–SVD Dictionary Update Stage

We should solve:

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

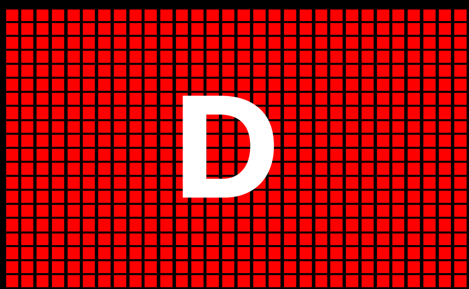




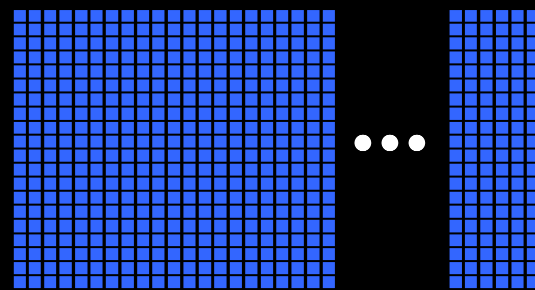
$\underline{d}_k$ is obtained by **SVD** on the examples' residual in G_k.

K–SVD: A Synthetic Experiment

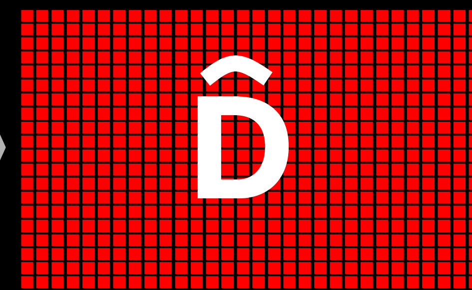
Create A 20×30 random dictionary with normalized columns



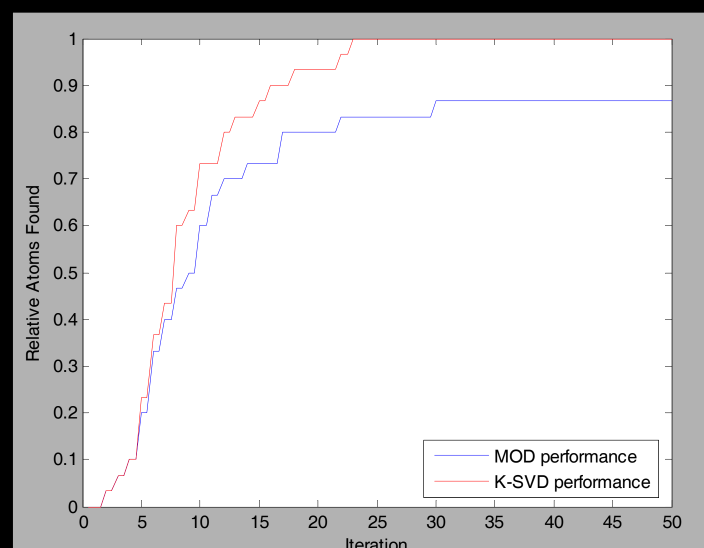
Generate 2000 signal examples with 3 atoms per each and add noise



Train a dictionary using the KSVD and MOD and compare



Results



To Summarize so far …

The *Sparseland* model for signals is relevant to us. In order to use it effectively we need to know **D**

How **D** can be found?

Use the K-SVD algorithm

Will it work well?

We have established a uniqueness result and shown how to practically train **D** using the K-SVD

What next?

(a) Deploy to applications

(b) Generalize in various ways: multiscale, non-negative factorization, speed-up, ...

(c) Performance guarantees?

Agenda

1. A Visit to *Sparseland*

   Motivating Sparsity & Overcompleteness

2. Problem 1: Transforms & Regularizations

   How & why should this work?

3. Problem 2: What About D?

   The quest for the origin of signals

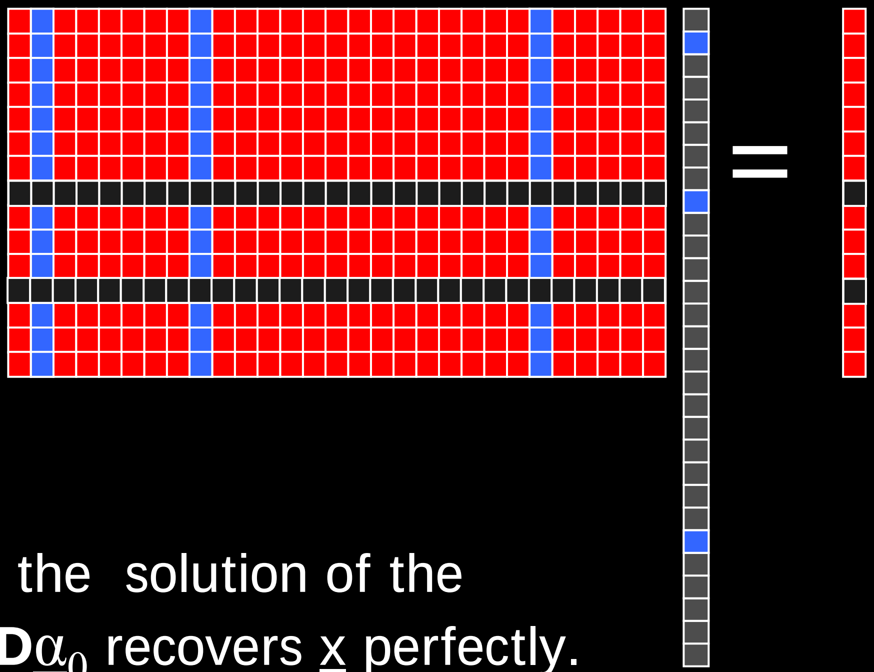
4. **Problem 3: Applications**

   Image filling in, denoising, separation, compression, …

Application 1: Image Inpainting

- Assume: the signal $\underline{x}$ has been created by $\underline{x}=D\underline{\alpha}_0$ with very sparse $\underline{\alpha}_0$.

$$\mathbf{D}\,\underline{\alpha}_0 = \underline{x}$$



- Missing values in $\underline{x}$ imply missing rows in this linear system.
- By removing these rows, we get

$$\tilde{\mathbf{D}}\underline{\alpha} = \underline{\tilde{x}}$$

- Now solve

$$\underset{\underline{\alpha}}{\text{Min}}\left\|\underline{\alpha}\right\|_0 \quad \text{s.t.} \quad \underline{\tilde{x}} = \tilde{\mathbf{D}}\underline{\alpha}$$

- If $\underline{\alpha}_0$ was sparse enough, it will be the solution of the above problem! Thus, computing $\mathbf{D}\underline{\alpha}_0$ recovers $\underline{x}$ perfectly.

Application 1: The Practice

- Given a noisy image $\underline{y}$, we can clean it using the Maximum A-posteriori Probability estimator by solving

$$\hat{\underline{\alpha}} = \underset{\underline{\alpha}}{\text{ArgMin}} \left\|\underline{\alpha}\right\|_p^p + \lambda \left\|\underline{y} - \mathbf{D}\underline{\alpha}\right\|_2^2 \quad \Rightarrow \quad \hat{\underline{x}} = \mathbf{D}\hat{\underline{\alpha}}$$

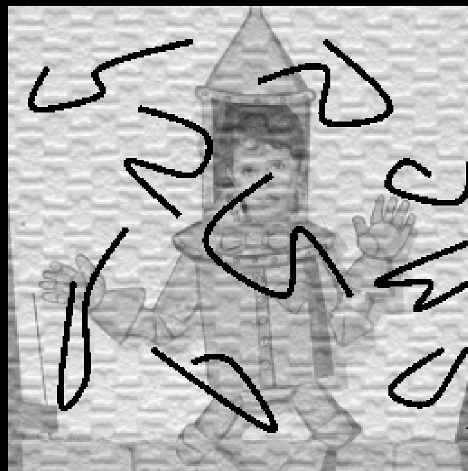
- What if some of the pixels in that image are missing (filled with zeros)? Define a mask operator as the diagonal matrix **W**, and now solve instead

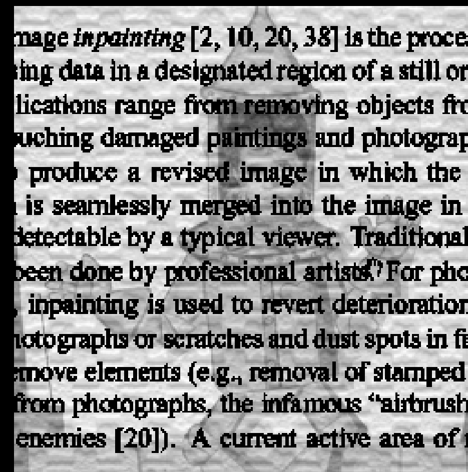
$$\hat{\underline{\alpha}} = \underset{\underline{\alpha}}{\text{ArgMin}} \left\|\underline{\alpha}\right\|_p^p + \lambda \left\|\underline{y} - \mathbf{W}\mathbf{D}\underline{\alpha}\right\|_2^2 \quad \Rightarrow \quad \hat{\underline{x}} = \mathbf{D}\hat{\underline{\alpha}}$$

- When handling general images, there is a need to concatenate two dictionaries to get an effective treatment of both texture and cartoon contents – This leads to separation [E., Starck, & Donoho ('05)].

Inpainting Results

Source



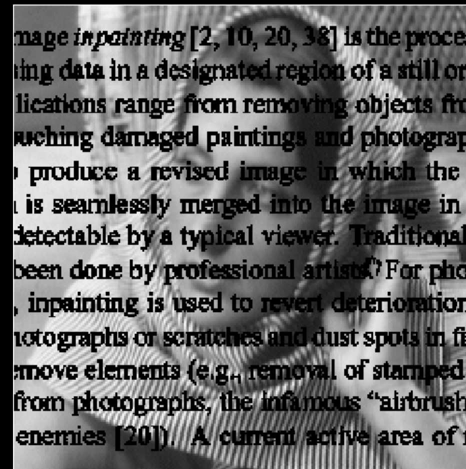


Predetermined dictionary: Curvelet (cartoon) + Overlapped DCT (texture)

Outcome

Inpainting Results

Source



predetermined dictionary: Curvelet (cartoon) + Overlapped DCT (texture)

Outcome

Inpainting Results

Application 2: Image Denoising

- Given a noisy image $\underline{y}$, we have already mentioned the ability to clean it by solving

$$\hat{\underline{\alpha}} = \underset{\underline{\alpha}}{\text{ArgMin}} \left\|\underline{\alpha}\right\|_p^p + \lambda \left\|\underline{y} - \mathbf{D}\underline{\alpha}\right\|_2^2 \quad \Rightarrow \quad \hat{\underline{x}} = \mathbf{D}\hat{\underline{\alpha}}$$

- When using the K-SVD, it cannot train a dictionary for large support images – How do we go from local treatment of patches to a global prior?

- The solution: Force shift-invariant sparsity - on each patch of size N-by-N (e.g., N=8) in the image, including overlaps.

Application 2: Image Denoising

$$\underline{\hat{\alpha}} = \underset{\underline{\alpha}}{\text{ArgMin}} \left\|\underline{\alpha}\right\|_p^p + \lambda \left\|\underline{y} - \mathbf{D}\underline{\alpha}\right\|_2^2 \Rightarrow \underline{\hat{x}} = \mathbf{D}\underline{\hat{\alpha}}$$

Our MAP penalty becomes

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

Extracts a patch in the ij location

Our prior

Application 2: Image Denoising

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

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

$\underline{x}$ and α_{ij} known

$\mathbf{D}$ and α_{ij} 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

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

K-SVD

Compute $\underline{x}$ by

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

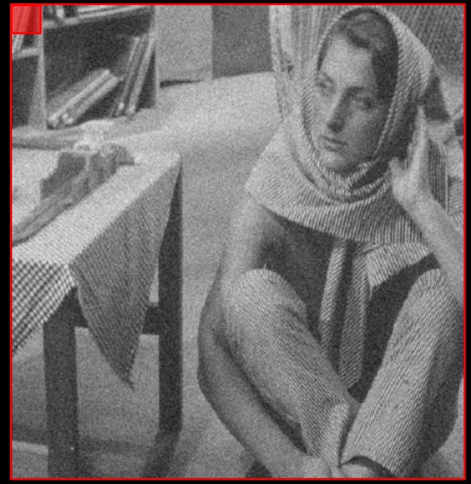
which is a simple averaging of shifted patches

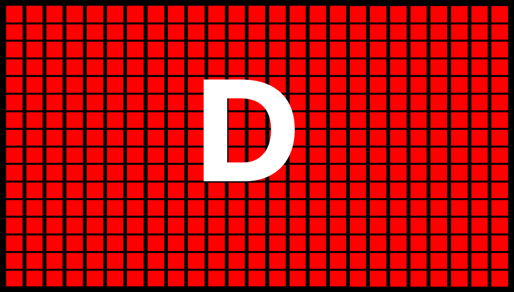
Application 2: The Algorithm

Sparse-code every patch of 8-by-8 pixels

Update the dictionary based on the above codes

Compute the output image $\underline{x}$

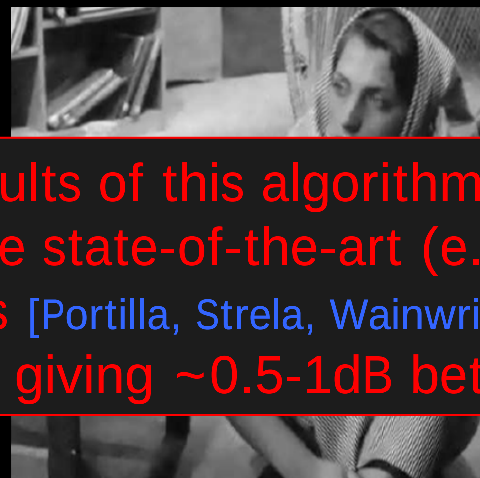




Complexity of this algorithm: $O(N^2 \times L \times \text{Iterations})$ per pixel. For N=8, L=1, and 10 iterations, we need 640 operations per pixel.

Denoising Results

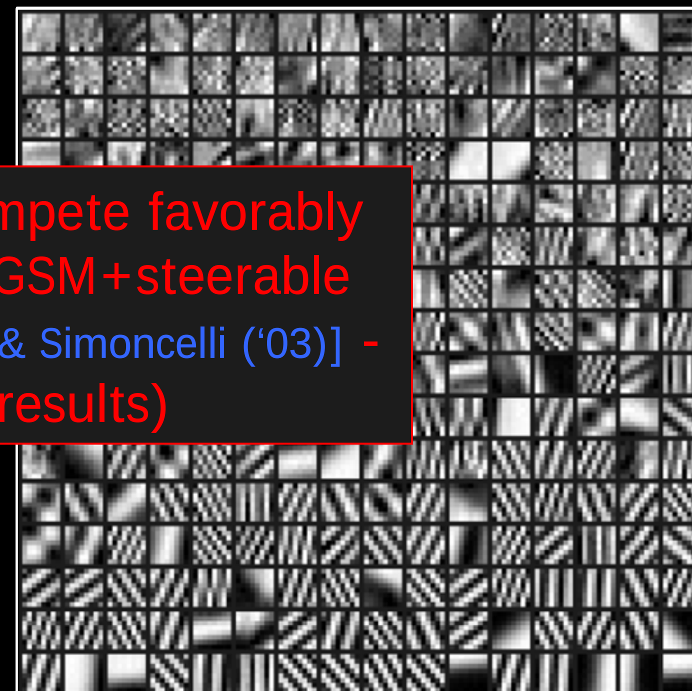
Source



Result 30.829dB

Noisy image

$\sigma = 20$

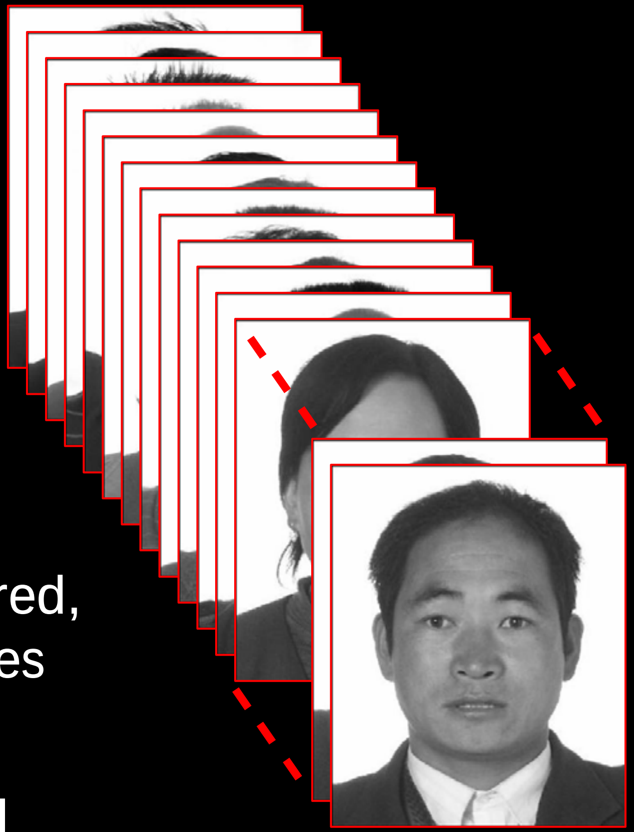


The obtained dictionary after 10 iterations

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)

Application 3: Compression

- The problem: Compressing photo-ID images.
- General purpose methods (JPEG, JPEG2000) do not take into account the specific family.
- By adapting to the image-content (PCA/K-SVD), better results could be obtained.
- For these techniques to operate well, train dictionaries locally (per patch) using a training set of images is required.
- In PCA, only the (quantized) coefficients are stored, whereas the K-SVD requires storage of the indices as well.
- Geometric alignment of the image is very helpful and should be done.



Application 3: The Algorithm

Detect main features and warp the images to a common reference (20 parameters)

↓

Divide the image into disjoint 15-by-15 patches. For each compute mean and dictionary

↓

Per each patch find the operating parameters (number of atoms L, quantization Q)

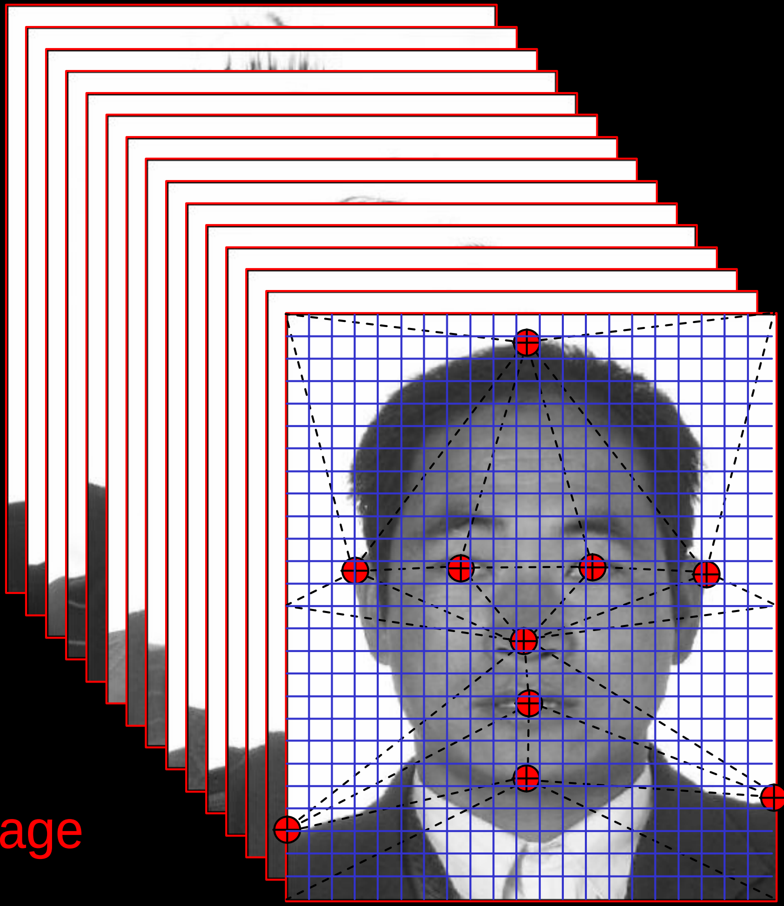
↓

Warp, remove the mean from each patch, sparse code using L atoms, and apply Q)

On the training set

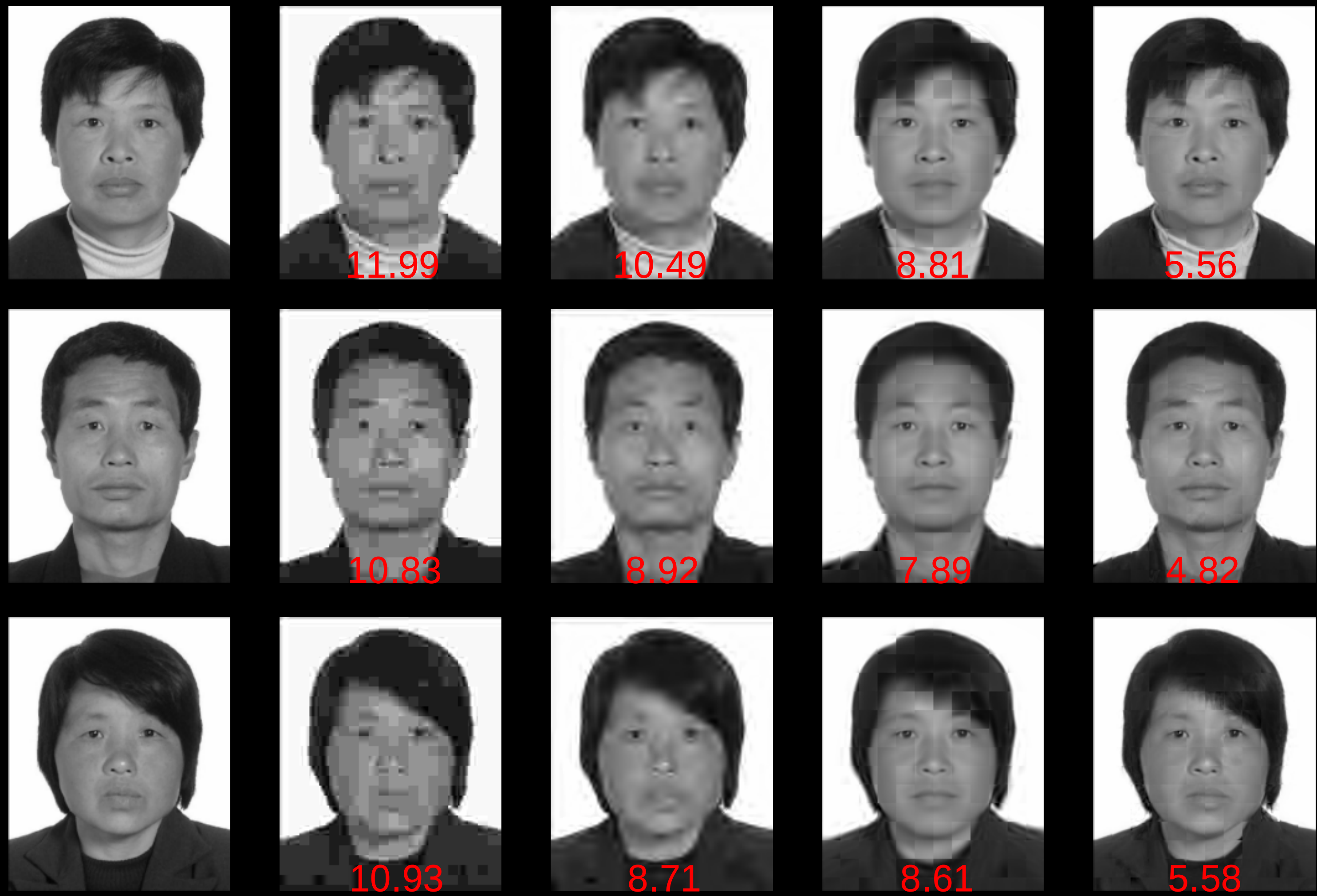
On the test image

Training set (2500 images)



Compression Results

Results for **820** Bytes per each file



Compression Results

Results for **550** Bytes per each file

Compression Results

Results for **400** Bytes per each file

Today We Have Discussed

1. A Visit to *Sparseland*

   Motivating Sparsity & Overcompleteness

2. Problem 1: Transforms & Regularizations

   How & why should this work?

3. Problem 2: What About **D**?

   The quest for the origin of signals

4. Problem 3: Applications

   Image filling in, denoising, separation, compression, ...

Summary

Sparsity and Over-completeness are important ideas that can be used in designing better tools in signal/image processing

There are difficulties in using them!

We are working on resolving those difficulties:

- Performance of pursuit alg.
- Speedup of those methods,
- Training the dictionary,
- Demonstrating applications,
- …

The dream?

Future transforms and regularizations will be data-driven, non-linear, overcomplete, and promoting sparsity.