# Sparsity-Based Signal Models and the Sparse K-SVD Algorithm

Ron Rubinstein

Advisor: Prof. Michael Elad

Geometric Image Processing

October 2010

# Signal Models

**Signal models** are a fundamental tool for solving low-level signal processing tasks

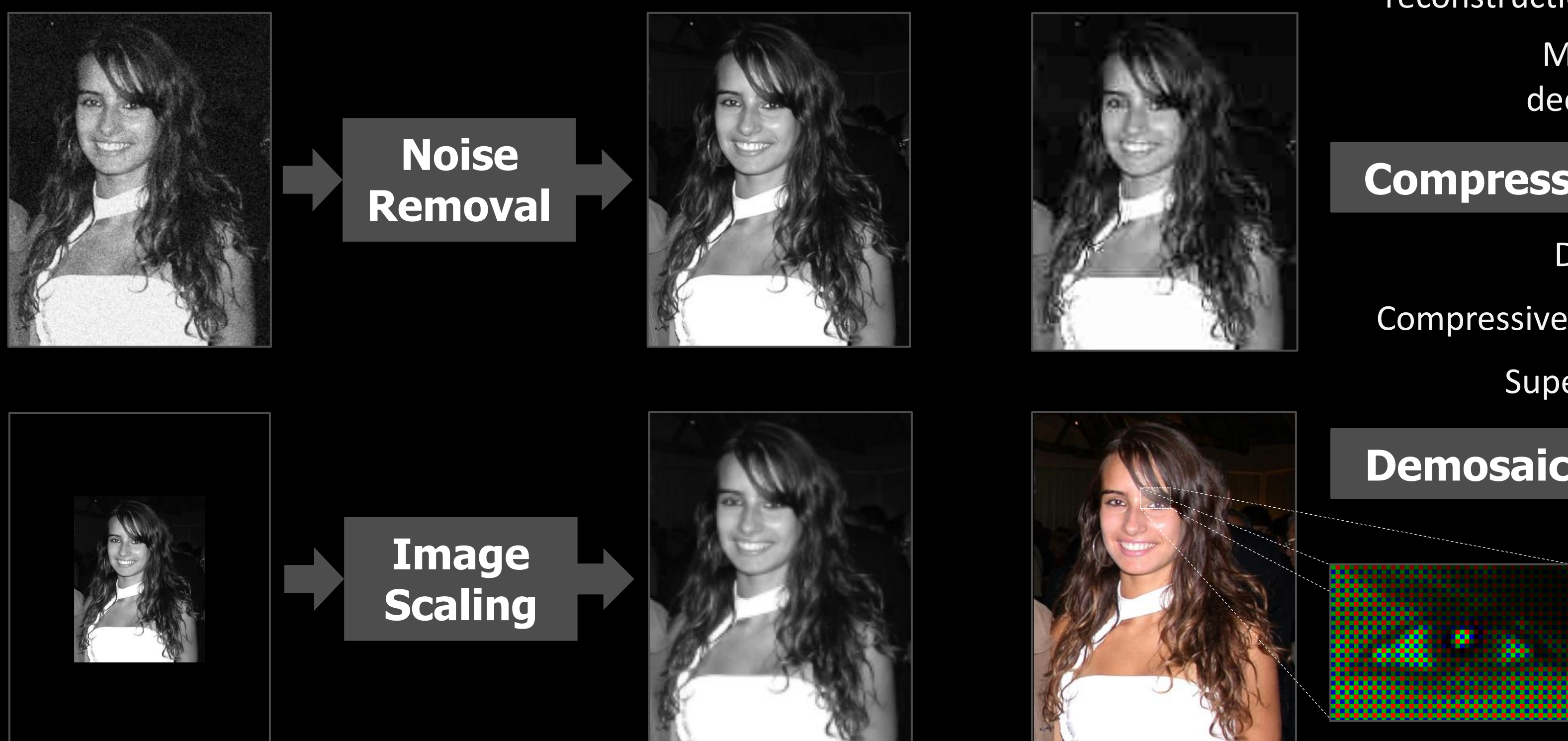
Detail enhancement

Deblurring

Inpainting

Tompgraphy reconstruction

Morphologic decomposition




Deinterlacing

Compressive sensing

Super-resolution

Source separation

# Signal Models

**Signal model:** a mathematical description of the behavior we expect from a “good” (uncontaminated ) signal in our system

**Smooth**

**Piecewise smooth**

**Smooth with point singularities**

**?**

# Signal Models are Everywhere

**Denoising:** $y = x + n$

$$\min_{\hat{x}} \frac{1}{2}\|y - \hat{x}\|_2^2 + R(\hat{x})$$

Regularizer = Image model

$y$ : Measured signal
$\hat{x}$ : Estimated signal

*Wavelet thresholding, total variation, BLS-GMS, K-SVD denoising…*

**General inverse problems:** $y = T\{x\} + n$

$$\min_{\hat{x}} \frac{1}{2}\|y - T\{\hat{x}\}\|_2^2 + R(\hat{x})$$

$x$ : Unknown signal
$T$ : Degradation operator

*Demosaicing, deblurring, inpainting, super-resolution,…*

# Signal Models are Everywhere

## Interpolation:

- Bilinear, bicubic: signals are piecewise-smooth
- Lanczos: signals are band-limited

## Compression:

- PNG: neighboring pixels have similar intensities
- JPEG: small image patches are smooth
- JPEG2K: images are smooth except for simple singularities

# Agenda

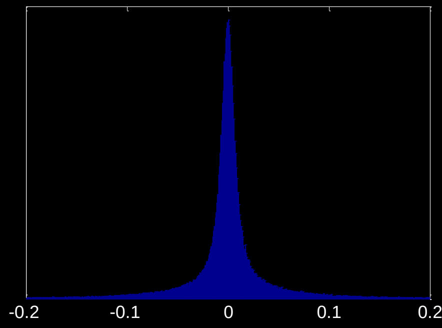
1. **Analysis and synthesis signal models**
   Two models are better than one
2. A few words on dictionary design
   On dictionaries and choices
3. Introduction to sparse representation
   Some background on sparsity and the K-SVD
4. The sparse dictionary model
   Introducing an efficient *and* adaptable dictionary!
5. Why sparse dictionaries are good for you
   Some uses and applications
6. Summary and conclusions

# Transform-Based Signal Models

Transform-based models compute a vector transform of the signal:

$$x \rightarrow \gamma(x)$$

Such signal models promote sparsity of $\gamma(x)$, in the sense that we expect the coefficients in $\gamma$ to decay rapidly.



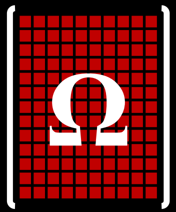

Wavelet Coefficient Distribution - Barbara

# Analysis and Synthesis Models

**Analysis models** use a set of linear filters, assumed to produce sparse inner products with the signal:

$$\mathbf{x} \Rightarrow \gamma = \boldsymbol{\Omega}\mathbf{x}$$

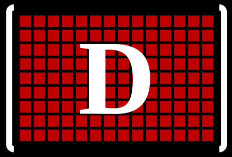
*The analysis dictionary contains the linear filters as its rows*




**Synthesis models** use a set of atomic signals, assumed to reproduce the input signal via a sparse linear combination:

$$\mathbf{x} \Rightarrow \mathbf{x} = \mathbf{D}\gamma$$

*The synthesis dictionary contains the atoms as its columns*



# Analysis versus Synthesis?

**Obvious question:** are the two models equivalent?

$$\gamma = \Omega x \quad \overset{?}{\Longleftrightarrow} \quad x = \Omega^{+} \gamma$$

$$D = \Omega^{+}$$

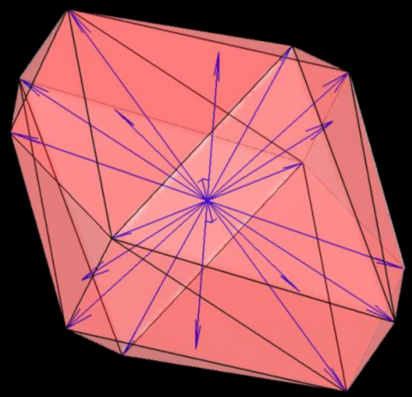
**Surprising answer:** NO!

*(Well, except for the invertible case, where we can use $D=\Omega^{-1}$)*

# Analysis versus Synthesis in the $L_1$ Case

For the widely-used $L_1$ case, [Elad, Milanfar & Rubinstein '07]

- Given any similar-sized $\mathbf{D}$ and $\mathbf{\Omega}$, we can find large number of signals where the two will substantially differ
- The analysis model is mathematically a subset of the synthesis model, but the mapping is exponentially complex
- There are practical cases where one may outperform the other



Geometric structure in the $L_1$ case

# Current Research on Analysis Models

**Sparsity measure:** how do we quantify the "sparsity" of the coefficients in $\Omega x$?

| $L_0$ (sparsity) | $L_p$ norms | Huber | Cauchy |
|---|---|---|---|

**Algorithms:** how do we efficiently solve the resulting optimization problems?

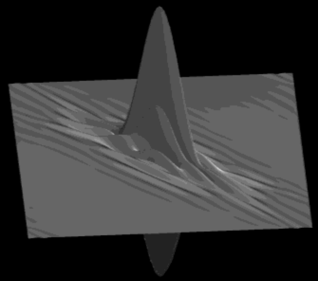
**Dictionary training:** can we use computational learning to infer the dictionary $\Omega$ from examples?

[Black & Roth '05] [Rubinstein & Elad '10]

# Agenda

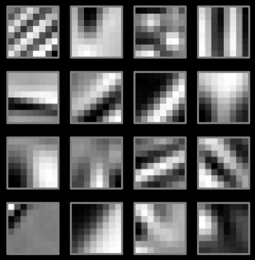
1. Analysis and synthesis signal models
   Two models are better than one
2. **A few words on dictionary design**
   On dictionaries and choices
3. Introduction to sparse representation
   Some background on sparsity and K-SVD
4. The sparse dictionary model
   Introducing an efficient *and* adaptable dictionary!
5. Why sparse dictionaries are good for you
   Some uses and applications
6. Summary and conclusions

# Designing Your Dictionary



Harmonic Analysis:

**Analytic dictionaries**



Machine Learning:

**Trained dictionaries**

[Rubinstein, Bruckstein & Elad '10]

# Analytic Dictionaries

**Analytic dictionaries** arise from a **mathematical model** of the signals

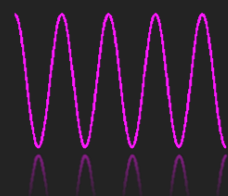
- ✓ Atoms have analytic formulations
- ✓ Known mathematical properties (e.g. coefficient decay rates)
- ✓ Fast algorithms for computing the transforms
- – Limited expressiveness: all signals behave the same

# Some Analytic Dictionaries

## Fourier

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

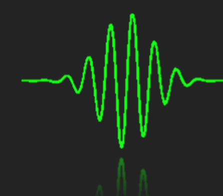
Smooth signals



## Gabor

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

Smooth signals



## Wavelets

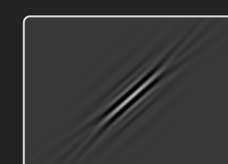
$$\phi_{m,n}(x) = \alpha^{m/2} f(\alpha^m x - \beta n)$$

Smooth + point singularities

## Curvelets [Candès & Donoho ’99]

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

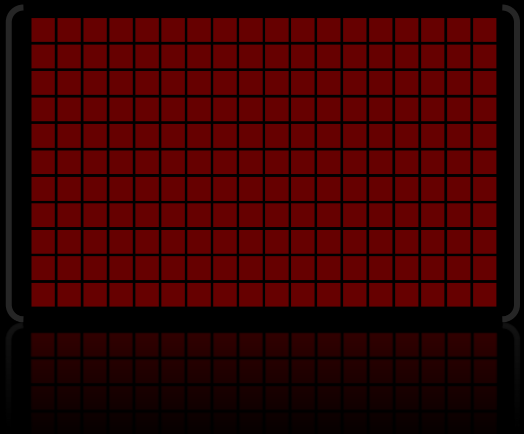
Smooth + curve singularities



# Trained Dictionaries

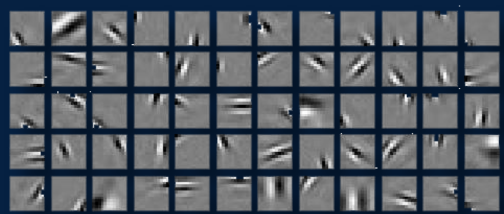
**Trained dictionaries** arise from a **set of examples** of the signal data

- ✓ Dictionary is learned from actual data
- ✓ Finer adaptation to the target signals
- ✓ Better performance in applications
- – Non-structured: higher complexity, single scale



# Dictionary Training Algorithms

**Olshausen & Field '96**



**MOD** **(Method of Optimal Directions)**

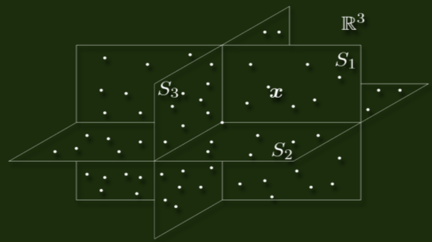
[Engan, Aase & Husøy '99]

**K-SVD**

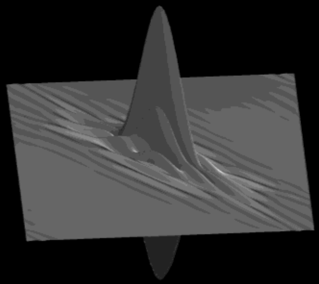
[Aharon, Elad & Bruckstein '06]

**Generalized PCA**

[Vidal, Ma & Sastry '05]

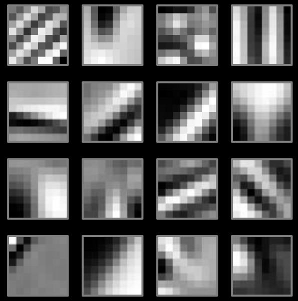


# Dictionary Design: Summary



## Analytic dictionaries

- ✓ Low complexity
- ✓ Optimal for specific classes of signals
- – Non-adaptive



## Trained dictionaries

- ✓ Adaptable to different signal types
- ✓ Better results in many applications
- – Non-structured

**Can we have it all ?**

# Agenda

1. Analysis and synthesis signal models
   Two models are better than one

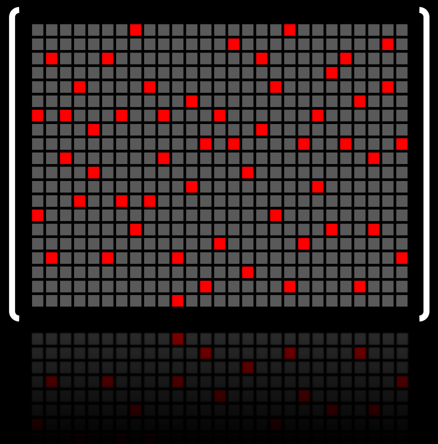
2. A few words on dictionary design
   On dictionaries and choices

3. **Introduction to sparse representation**
   Some background on sparsity and the K-SVD

4. The sparse dictionary model
   Introducing an efficient *and* adaptable dictionary!

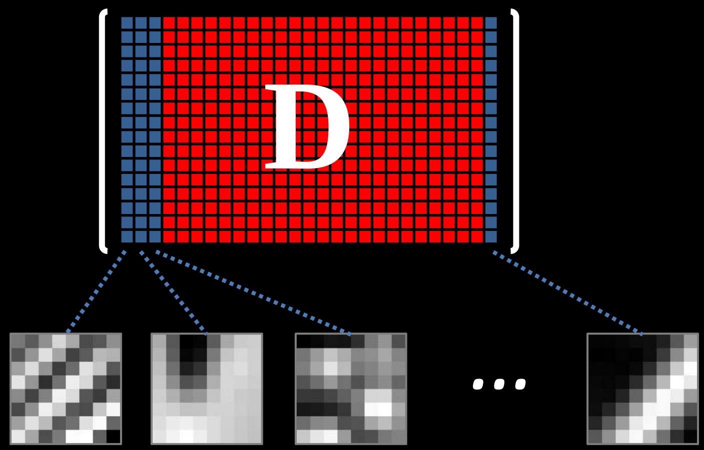
5. Why sparse dictionaries are good for you
   Some uses and applications

6. Summary and conclusions



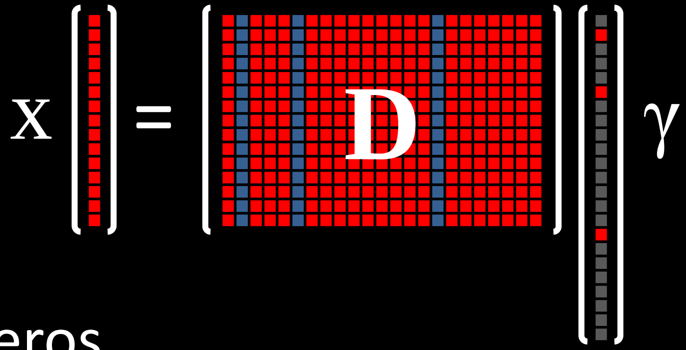
# The Sparse Representation Model

- We assumes the existence of a synthesis dictionary $\mathbf{D} \in \mathbb{R}^{N \times L}$ whose columns are the atom signals.




- We model natural signals as sparse linear combinations of the dictionary atoms:

$$\mathrm{x} = \mathbf{D}\gamma$$




- We seek exact sparsity of $\gamma$, meaning that it is assumed to contain mostly zeros.

# Sparse Coding

**Problem 1 (sparse coding):** given a signal $\mathbf{x}$, can we find its representation $\gamma$ over $\mathbf{D}$?

- The equation for $\gamma$ is underdetermined: $\mathbf{x} = \mathbf{D}\gamma$
- Among all the solutions, we want the sparsest one:

$$\min_{\gamma} \|\gamma\|_0 \quad \text{s.t.} \quad \mathbf{x} = \mathbf{D}\gamma$$

*The "$L_0$ norm" counts the number of non-zeros in a vector*

# Noisy Sparse Coding

**Problem 2 (noisy sparse coding):** what if we only have $\mathrm{y}$, a noisy version of $\mathrm{x}$?

$$\mathrm{y} = \mathrm{x} + \mathrm{n} = \mathbf{D}\gamma + \mathrm{n}$$

Additive Gaussian noise

- Use the sparsity assumption to recover $\gamma$ *and* approximate $\mathrm{x}$:

$$\min_{\gamma} \|\gamma\|_0 \quad \text{s.t.} \quad \|\mathrm{y} - \mathbf{D}\gamma\|_2 \le \epsilon \quad \Rightarrow \quad \hat{\mathrm{x}} = \mathbf{D}\gamma$$

# Sparse Coding Algorithms

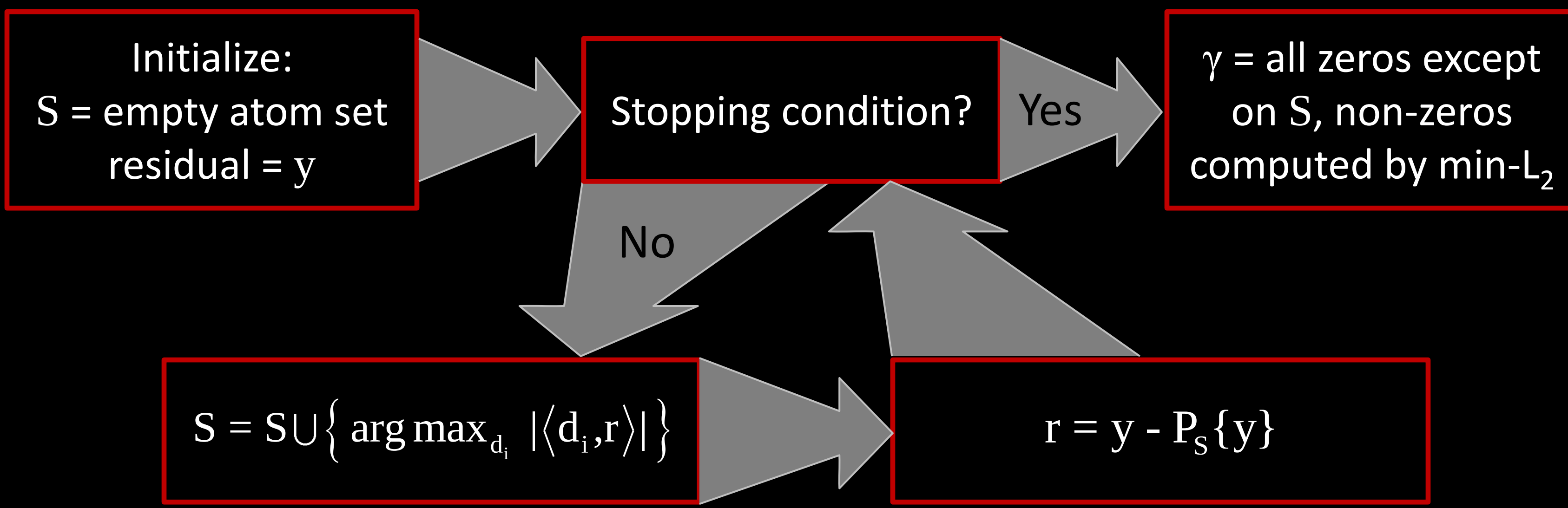
- The sparse coding problem NP-hard in general!
- Many efficient approximation algorithms:

| Greedy | Iterative | Relaxation | Others |
|---|---|---|---|
| MP/OMP | FOCUSS | Basis Pursuit | Thresholding |
| StOMP | BCR | LASSO | Shrinkage |
| LARS | Iterated Shrinkage | | RandOMP |
| CoSAMP | | | |

- Success bounds available, depending on the sparsity of $\gamma$ and the amount of noise
- Empirical performance typically much better than theory

# Orthogonal Matching Pursuit (OMP)

- Greedy algorithm – selects atoms one at a time
- Input: signal $y$, dictionary $\mathbf{D}$
- Output: sparse $\gamma$ such that $y \approx \mathbf{D}\gamma$

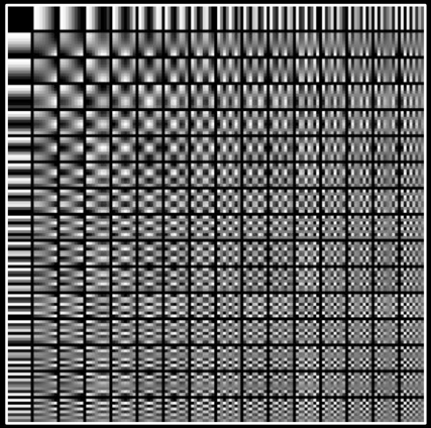


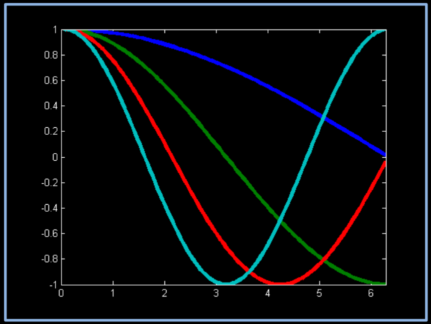
# Orthogonal Matching Pursuit (OMP)

- Block size: 8 x 8
- Overcomplete DCT dictionary, 256 atoms

8 atoms / block

PSNR = 32.5dB



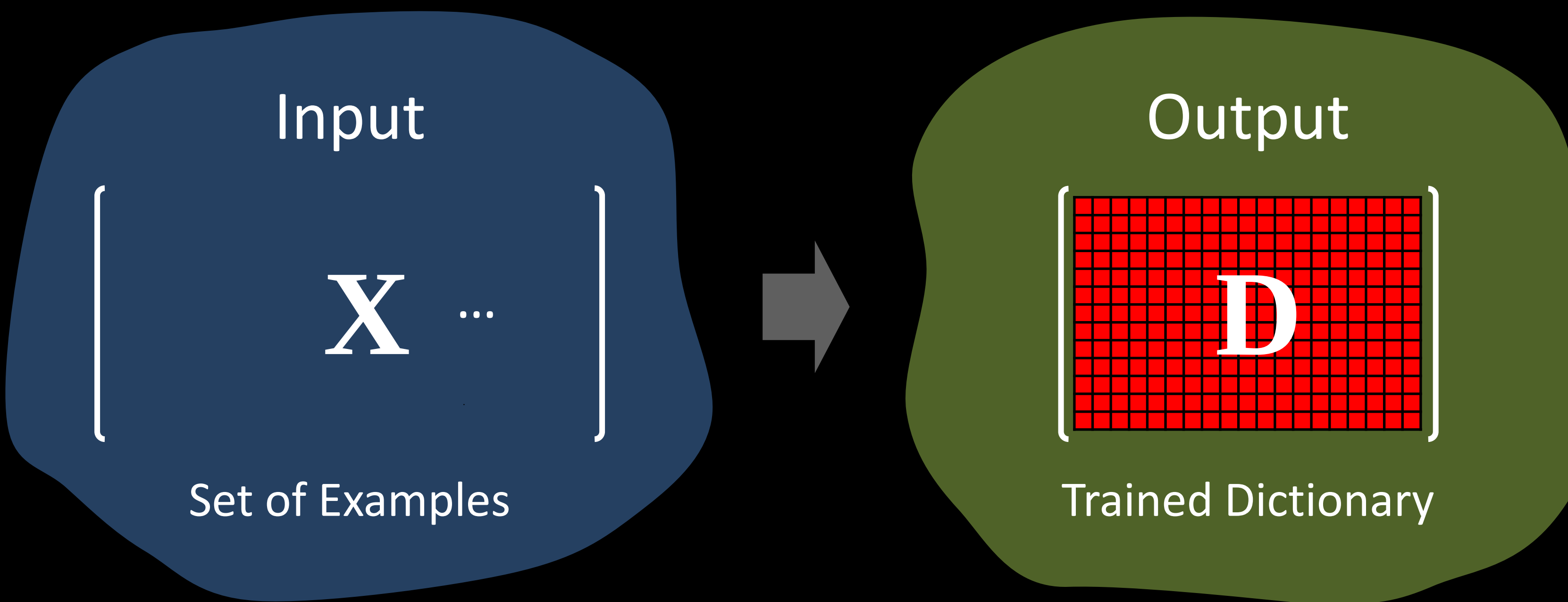



*Overcomplete DCT = extension of DCT with non-integer wave numbers*

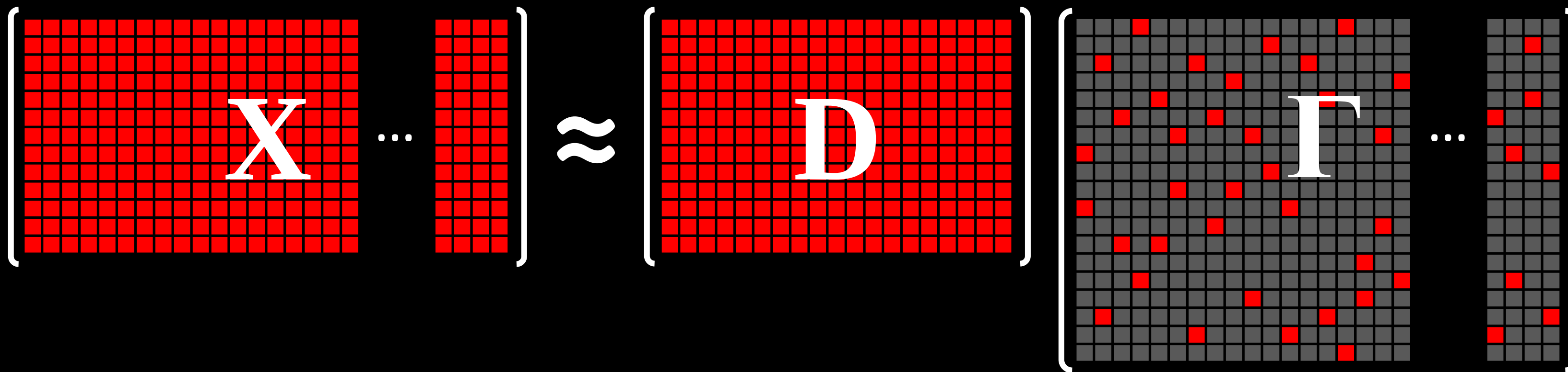
# Which Dictionary?

**The K-SVD algorithm:** [Aharon, Elad & Bruckstein '06]

Train an explicit dictionary from examples



# The K-SVD Training Algorithm



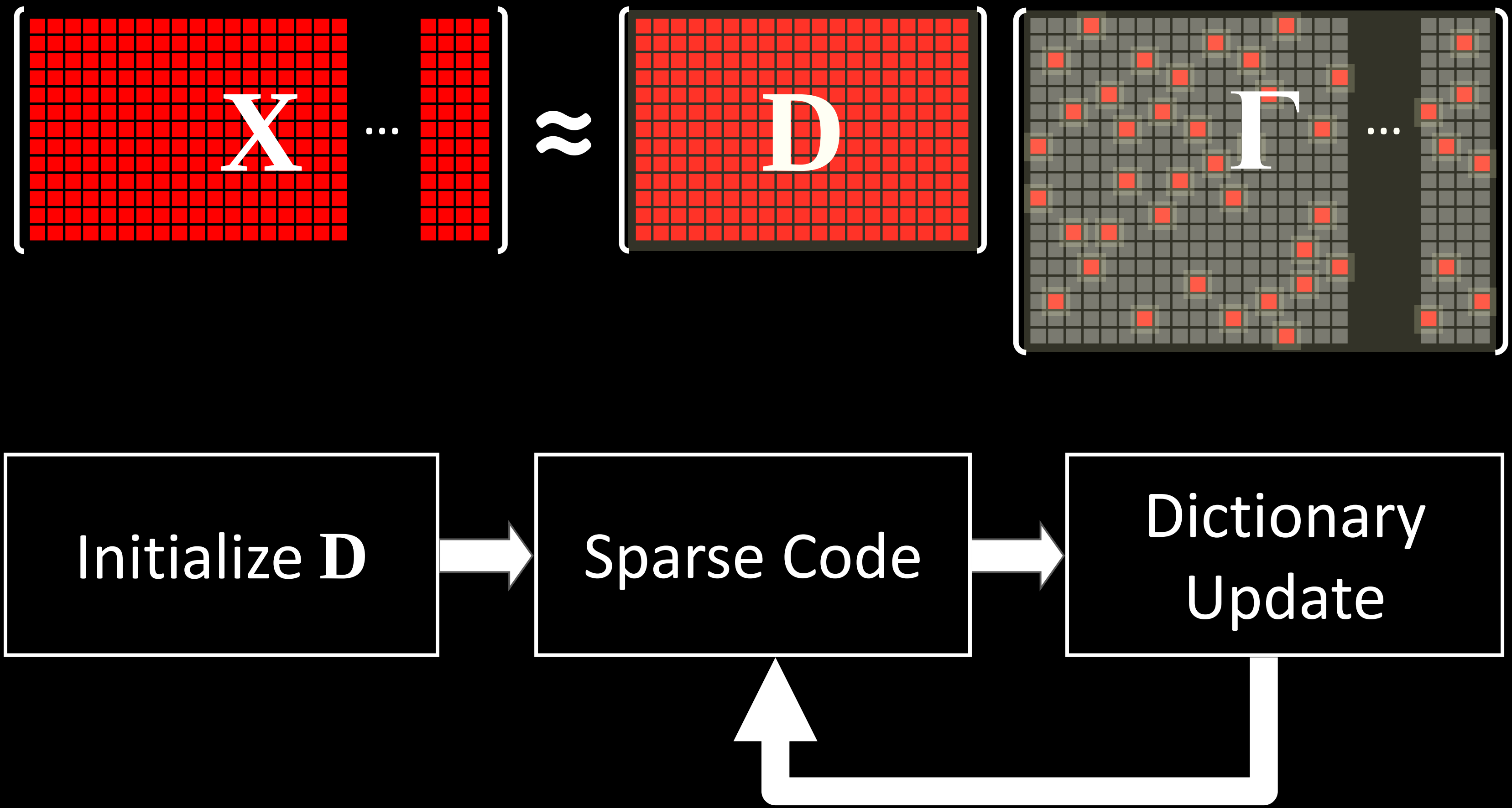

The target function to minimize:

$$\min_{\mathbf{D},\mathbf{\Gamma}} \left\| \mathbf{X} - \mathbf{D}\mathbf{\Gamma} \right\|_F^2 \quad \text{s.t.} \quad \forall i \;\; \left\| \gamma_i \right\|_0 \le T$$

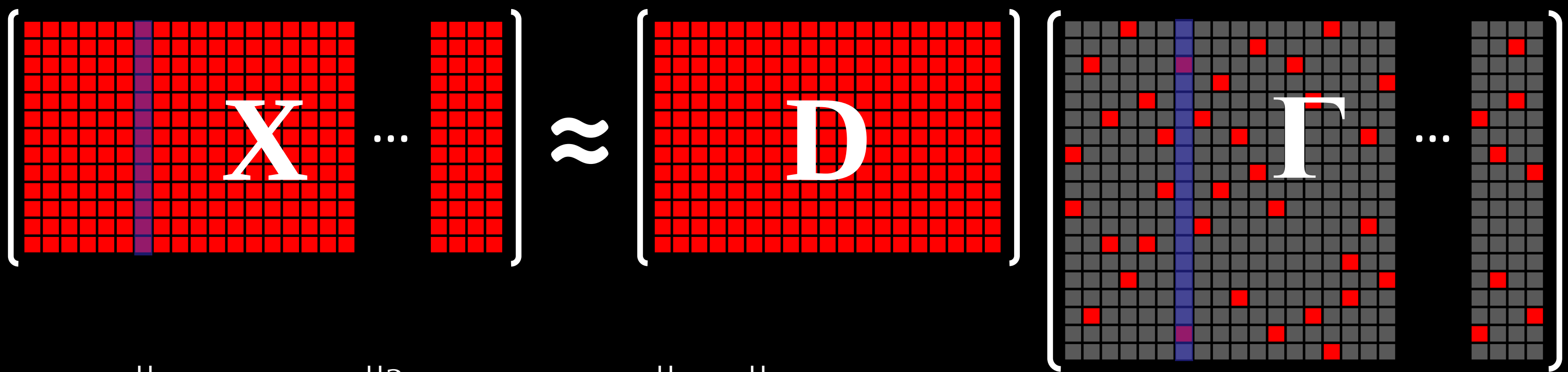
The examples are linear combinations of the atoms

Each representation uses at most $T$ atoms

# The K-SVD: Overview



# K-SVD: Sparse Coding Stage



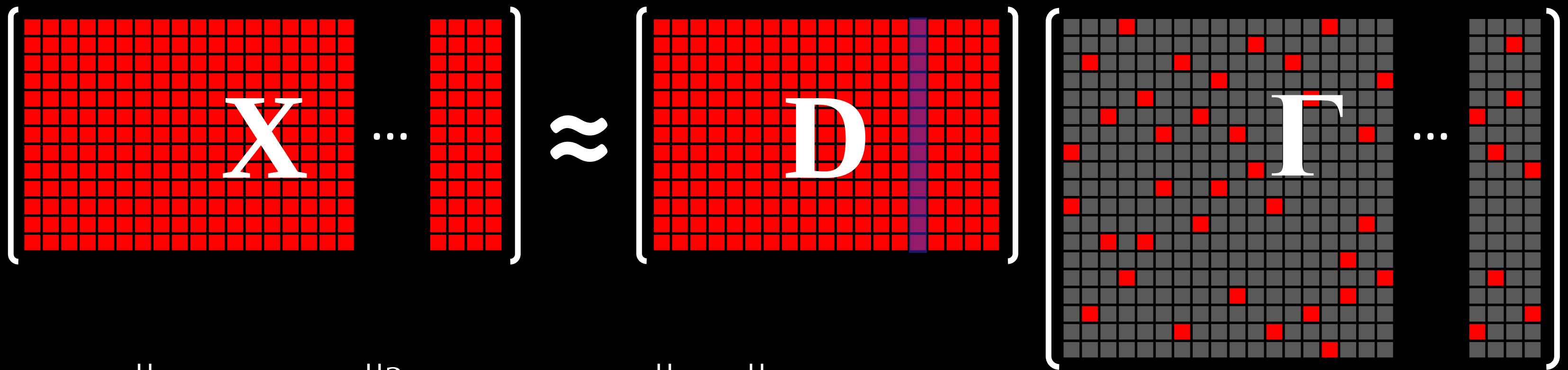

$$\min_{\boldsymbol{\Gamma}} \| \mathbf{X} - \mathbf{D}\boldsymbol{\Gamma} \|_F^2 \quad \text{s.t.} \quad \forall i \;\; \| \gamma_i \|_0 \le T$$

For the j$^{th}$ example

$$\min_{\gamma_j} \| \mathbf{x}_j - \mathbf{D}\gamma_j \|_F^2 \quad \text{s.t.} \quad \| \gamma_j \|_0 \le T$$

**Ordinary Sparse Coding !**

# K-SVD: Dictionary Update Stage



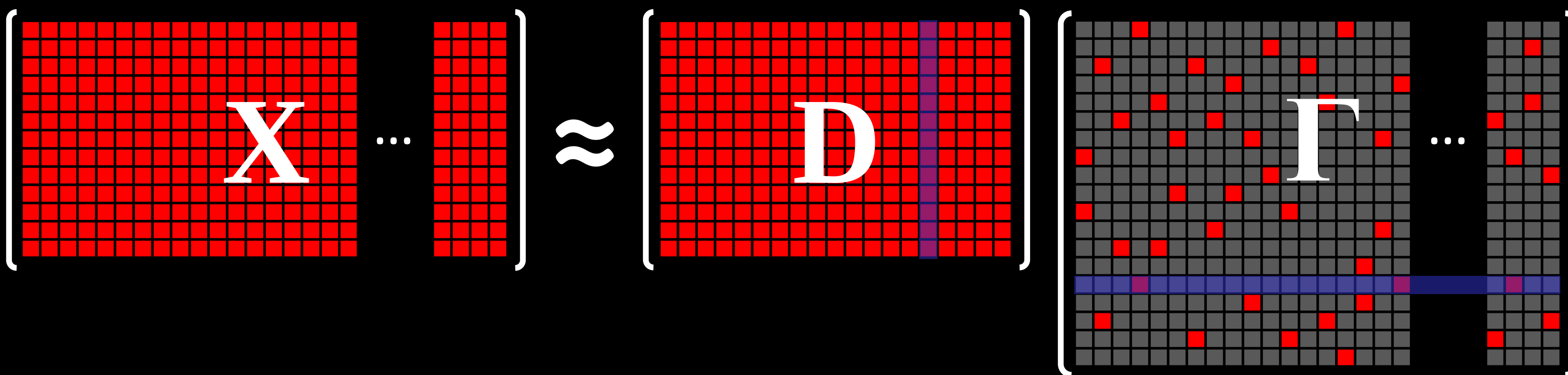

$$\min_{\mathbf{D}} \| \mathbf{X} - \mathbf{D}\boldsymbol{\Gamma} \|_F^2 \quad \text{s.t.} \quad \forall i \ \| \gamma_i \|_0 \le \mathrm{T}$$

For the k^th atom

$$\min_{\mathrm{d}_k} \| \mathbf{E}_k - \mathrm{d}_k \gamma_k^T \|_F^2 \quad , \qquad \mathbf{E}_k = \mathbf{X} - \sum_{j \neq k} \mathrm{d}_j \gamma_j^T$$

**(Residual)**

# K-SVD: Dictionary Update Stage




$$\min_{\mathrm{d}_k} \| \mathbf{E}_k - \mathrm{d}_k \gamma_k^T \|_F^2$$

We can do better!

$$\min_{\mathrm{d}_k, \gamma_k^T} \| \mathbf{E}_k - \mathrm{d}_k \gamma_k^T \|_F^2$$

But wait! What about sparsity?

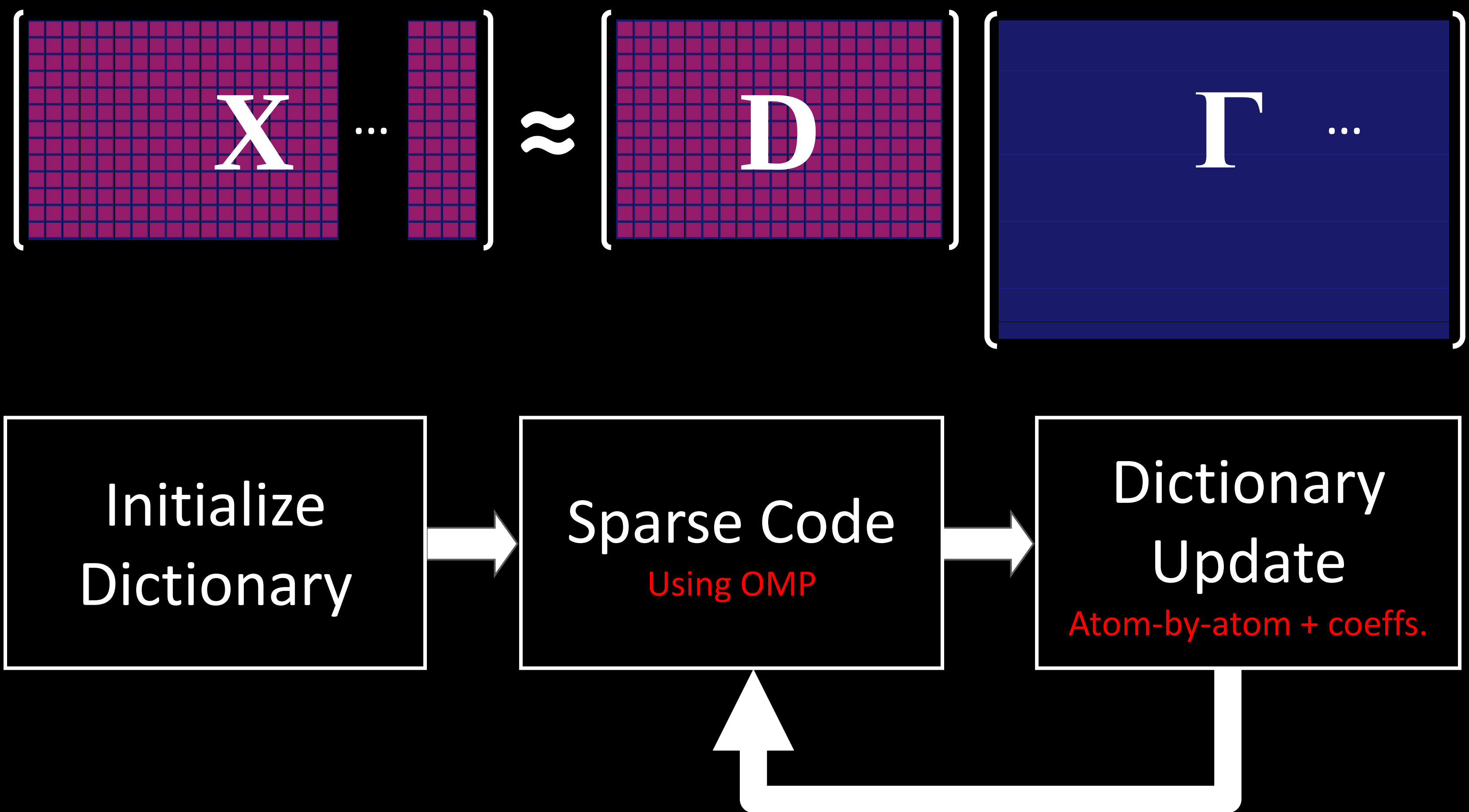
# K-SVD: Dictionary Update Stage

We want to solve:

$$\min_{d_k, \tilde{\gamma}_k^T} \left\| d_k \tilde{\gamma}_k^T - \tilde{E}_k \right\|_F^2$$

Only some of the examples use atom $d_k$ → When updating $\gamma_k$, only recompute the coefficients for those examples → Solve with **SVD**

# The K-SVD: Summary

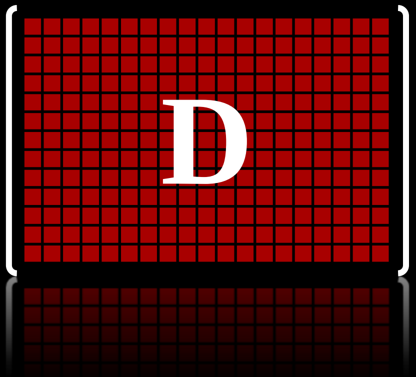


# The K-SVD: Applications

- ✓ Image denoising [Elad & Aharon ‘06]
- ✓ Image inpainting [Raboy ‘07]
- ✓ Tomography reconstruction [Liao & Sapiro ‘08]
- ✓ Demosaicing [Mairal, Elad & Sapiro ‘08]
- ✓ Facial image compression [Bryt & Elad ‘08]
- ✓ Video denoising [Protter & Elad ’09]
- ✓ Image scaling [Zeyde, Elad & Protter ‘10]
- ✓ Many others…

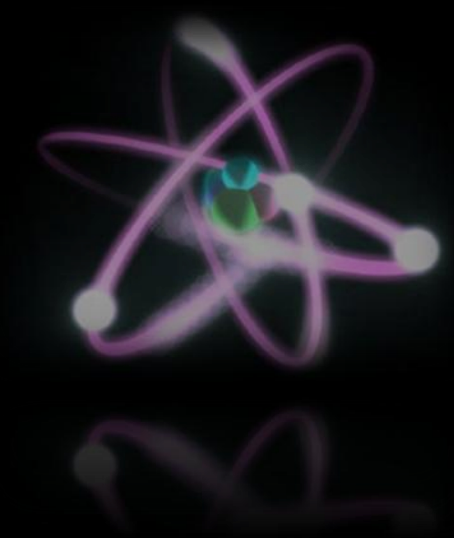
# Limitations of Explicit Dictionaries

- **Inefficient:** applied via explicit matrix multiplication
- **Unstructured:** complex to store and transmit
- **Over-parameterized:** many degrees of freedom require a lot of training examples
- **Single scale:** adapted to a specific signal size, inter-scale relations are not expressed

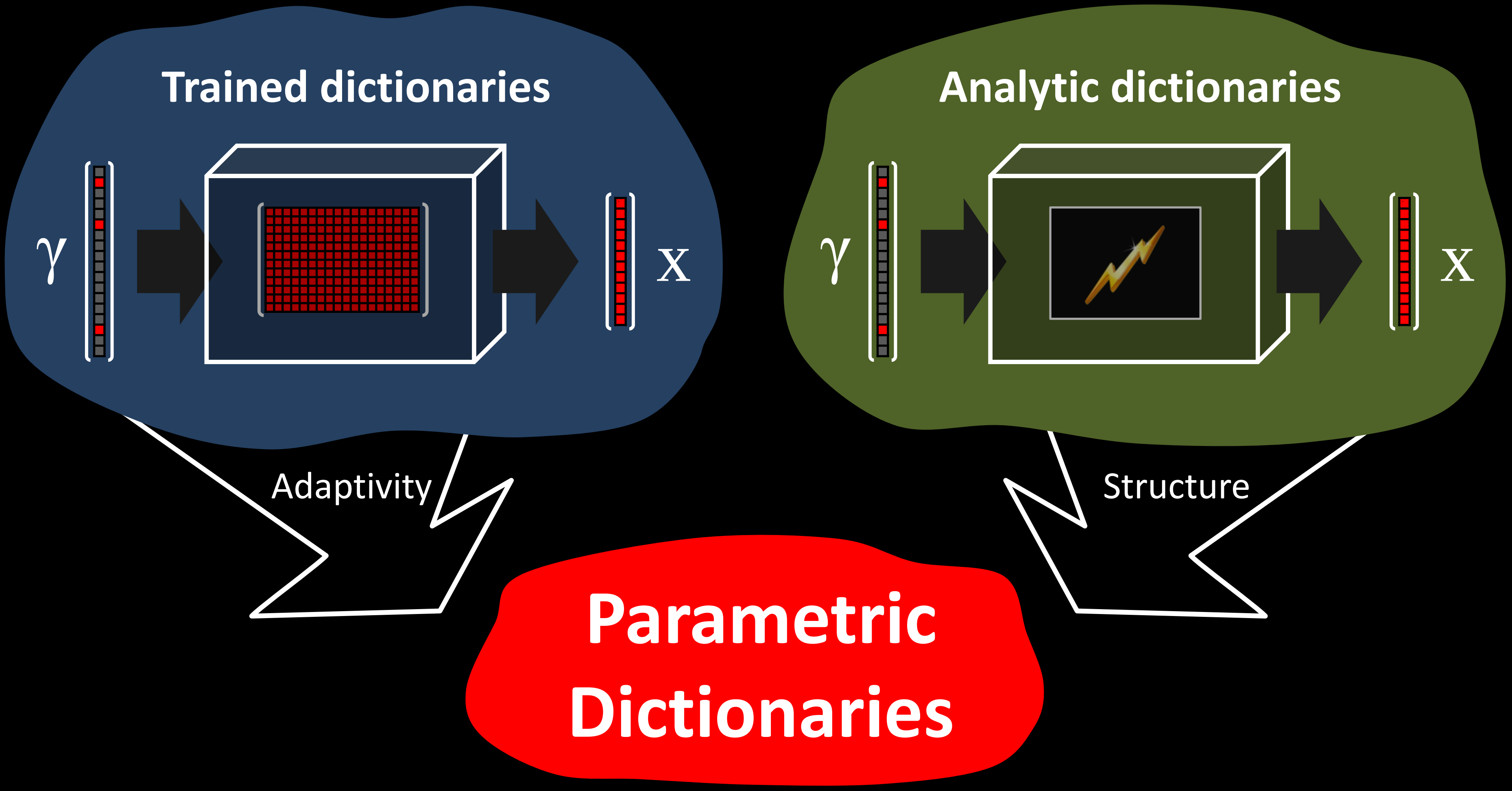


# Agenda

1. Analysis and synthesis signal models
   Two models are better than one
2. A few words on dictionary design
   On dictionaries and choices
3. Introduction to sparse representation
   Some background on sparsity and the K-SVD
4. **The sparse dictionary model**
   Introducing an efficient *and* adaptable dictionary!
5. Why sparse dictionaries are good for you
   Some uses and applications
6. Summary and conclusions



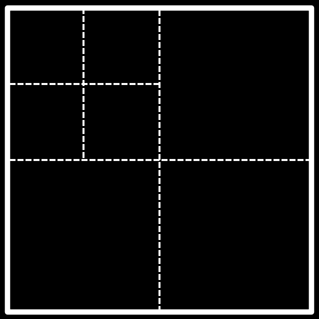
# Parametric Dictionaries



# Some Existing Parametric Dictionaries

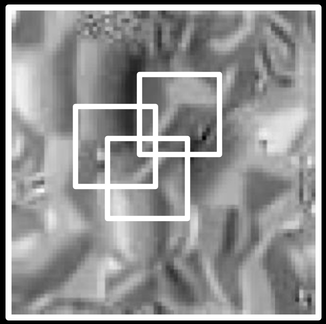
Semi-multiscale K-SVD

[Mairal, Sapiro & Elad, '08]



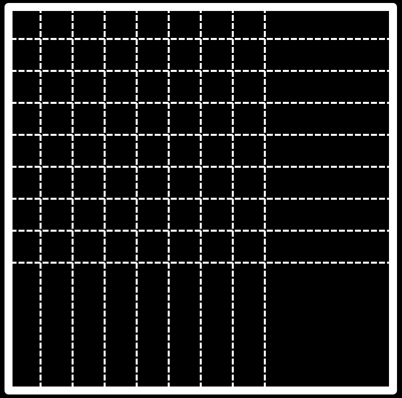
Signature dictionary

[Aharon & Elad, '08]



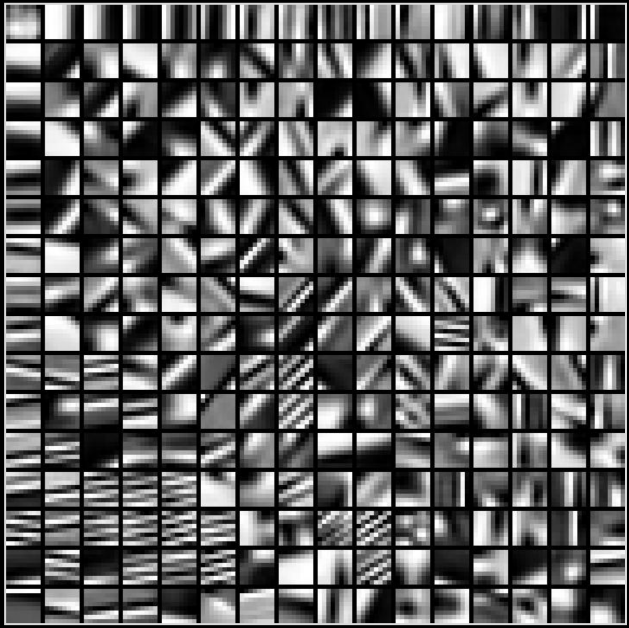
Iterative LS Dictionary Learning Algorithms (ILS-DLA)

[Engan, Skretting & Husøy, '07]



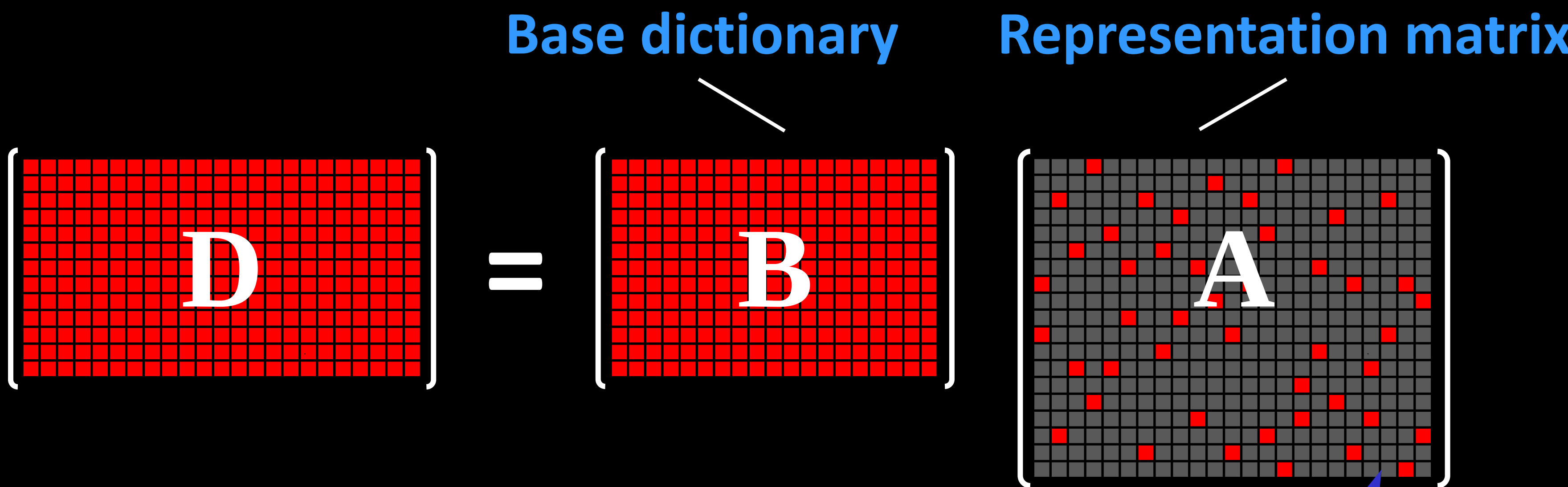
# Sub-Atomic Particles?

K-SVD dictionary



Could the trained atoms *themselves* be sparse over some simpler underlying dictionary?

# The Sparse Dictionary Model [Rubinstein, Zibulevsky & Elad '10]



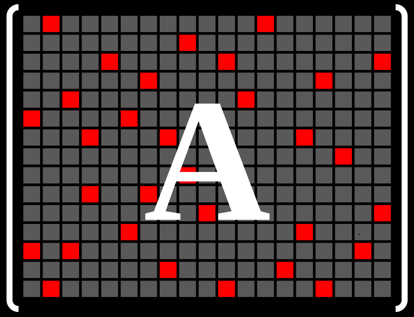

$$\mathbf{X} \approx \mathbf{B}\,\mathbf{A}\,\boldsymbol{\Gamma}$$

**Base dictionary** — **Sparse matrix** — **Sparse matrix**

**Double Sparsity**

# The Sparse Dictionary Model


**Efficiency:** depends mostly on the choice of base dictionary – typically an *analytic dictionary*.




**Adaptivity:** by modifying the representation matrix A.

# Complexity of Sparse Dictionaries

| Explicit Dictionaries | Sparse Dictionaries | Analytic Dictionaries |
| --- | --- | --- |
| $\Theta\left(N^2\right)$ | $\Theta\left(N \cdot N^{1/d} + pN\right)$ | $\Theta\left(N \log N\right)$<br>$\Theta\left(N\right)$ … |

*Signal is d-dimensional*

*Signal size: $n \times n \times \ldots \times n = n^d = N$*

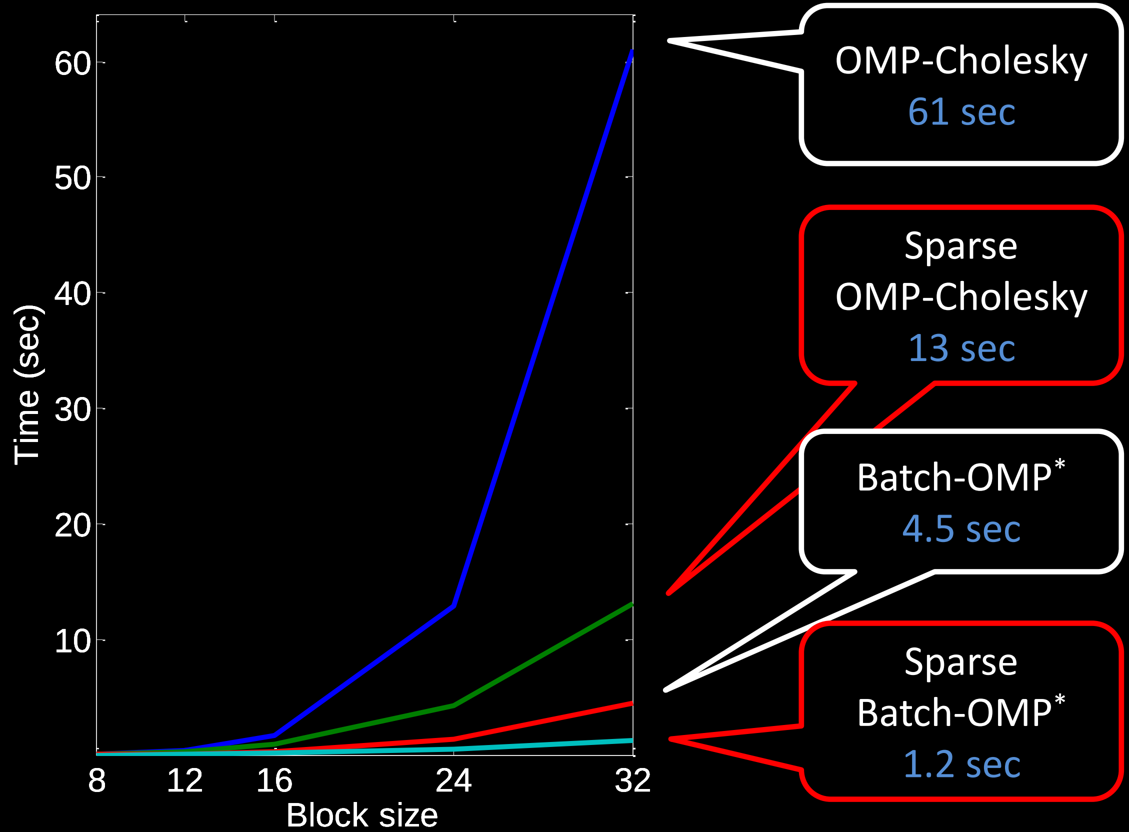
*Dictionary size: $N \times L$, $L=O(N)$*

*Base dictionary size: $N \times L$, separable*

*Cardinality of each sparse atom: $p$*

# Efficiency of Sparse Dictionaries

Example: patches from *pirate* – encoding time per 1,000 blocks.




*Signal size:*
*n x n =* $n^2$ *= N*

*Dictionary size:*
*N x 2N*

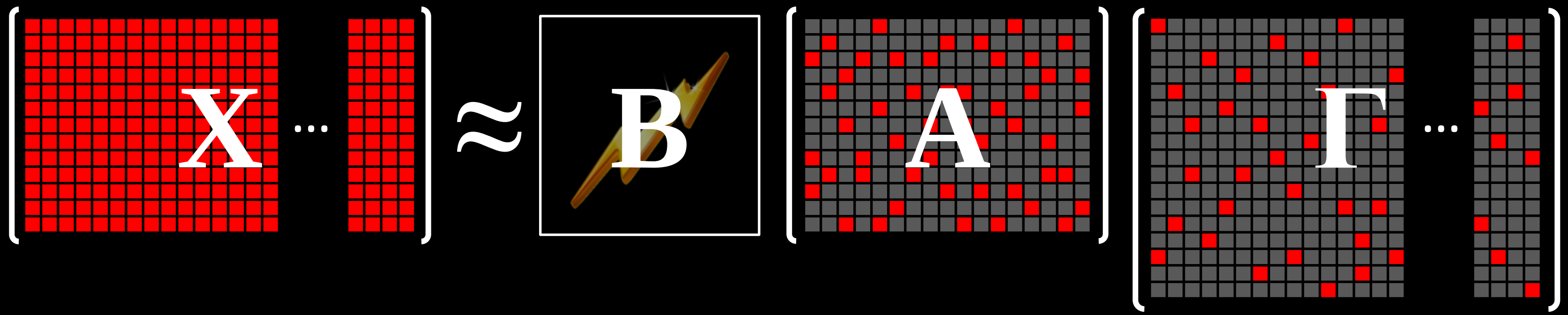
*Base dictionary:*
*2-D Separable*

*Atom sparsity: n*

*Encoding target:*
*n/2 atoms*

* [Rubinstein, Zibulevsky & Elad '08]

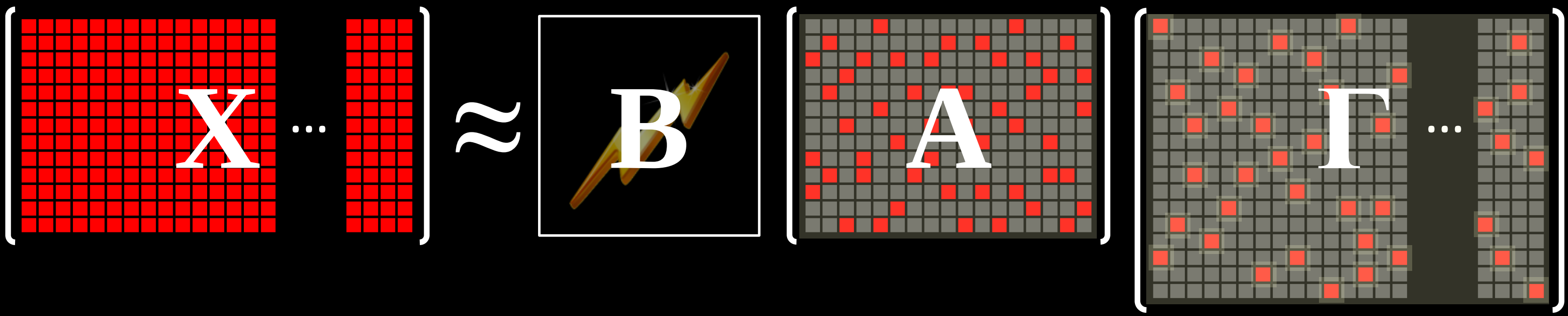
# Training a Sparse Dictionary

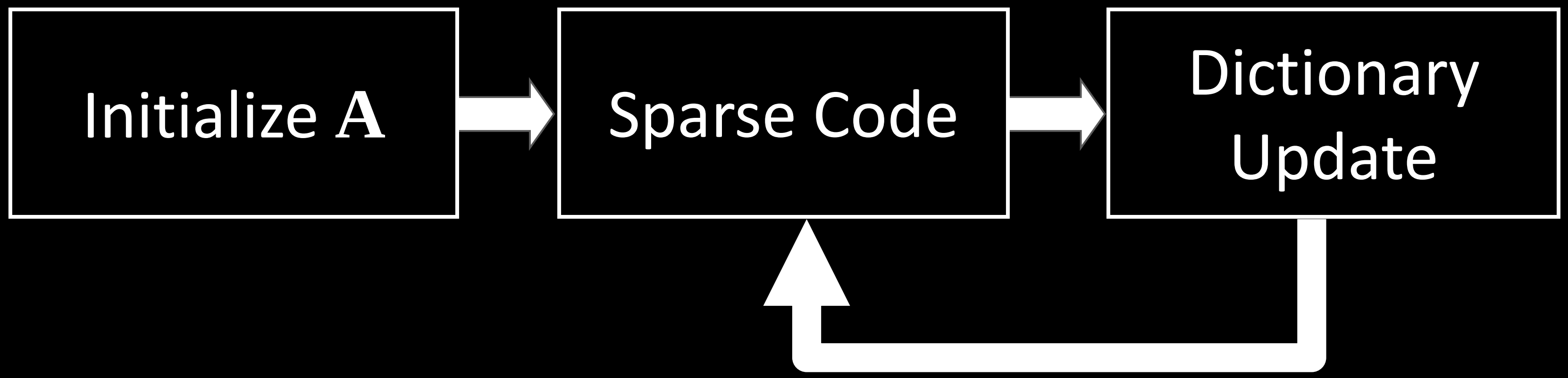



The target function to minimize:

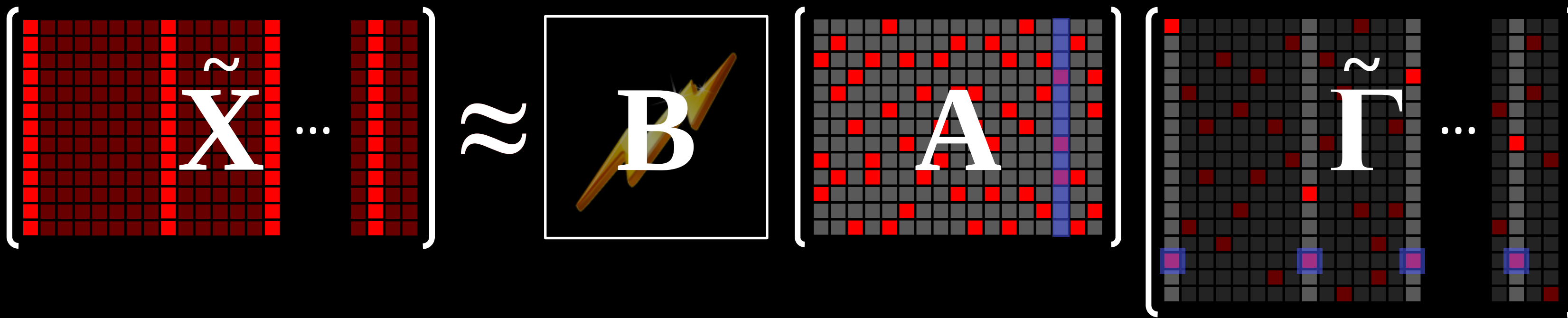
$$\min_{\mathbf{A},\boldsymbol{\Gamma}} \|\mathbf{X}-\mathbf{B}\mathbf{A}\boldsymbol{\Gamma}\|_F^2 \quad \text{s.t.} \quad \begin{cases} \forall i \ \ \|\gamma_i\|_0 \le \mathrm{T} \\ \forall j \ \ \|\mathrm{a}_j\|_0 \le \mathrm{P} \end{cases}$$

# The Sparse K-SVD



# Sparse K-SVD: Atom Update Stage




$$\min_{A,\Gamma} \| X - BA\Gamma \|_F^2 \quad \text{s.t.} \quad \begin{cases} \forall i \ \| \gamma_i \|_0 \le T \\ \forall j \ \| a_j \|_0 \le P \end{cases}$$

For the k^th atom

$$\min_{a_k, \tilde{\gamma}_k^T} \| \tilde{E}_k - B a_k \tilde{\gamma}_k^T \|_F^2 \quad \text{s.t.} \quad \| a_k \|_0 \le P \quad , \quad \tilde{E}_k = X - \sum_{j \ne k} B a_j \tilde{\gamma}_j^T$$

# Sparse K-SVD: Atom Update Stage

$$\min_{a_k,\tilde{\gamma}_k^T} \left\| \tilde{\mathbf{E}}_k - \mathbf{B} a_k \tilde{\gamma}_k^T \right\|_F^2 \quad \text{s.t.} \quad \| a_k \|_0 \le P$$

**Block relaxation:**

$$\min_{a_k} \| \tilde{\mathbf{E}}_k - \mathbf{B} a_k \tilde{\gamma}_k^T \|_F^2 \quad \text{s.t.} \quad \| a_k \|_0 \le P$$

$$\min_{\tilde{\gamma}_k^T} \| \tilde{\mathbf{E}}_k - \mathbf{B} a_k \tilde{\gamma}_k^T \|_F^2$$

Some Math

Sparse Coding Over B !

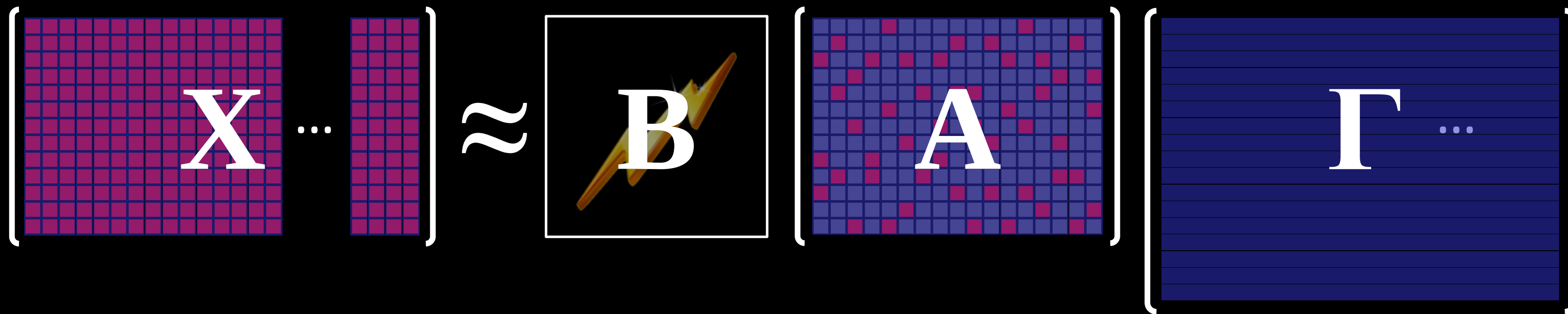
$$\min_{a_k} \| \tilde{\mathbf{E}}_k \tilde{\gamma}_k - \mathbf{B} a_k \|_F^2 \quad \text{s.t.} \quad \| a_k \|_0 \le P$$

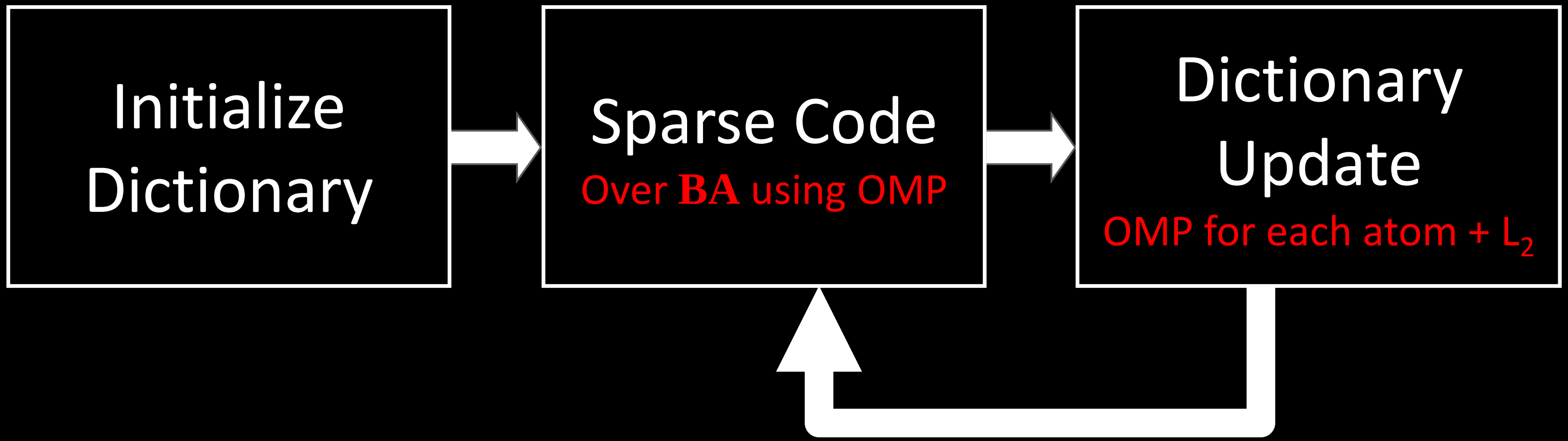
Ordinary $L_2$

$$\tilde{\gamma}_k = \tilde{\mathbf{E}}_k^T \mathbf{B} a_k$$

(Assumes $\mathbf{B}a_k$ is normalized)

# The Sparse K-SVD: Summary

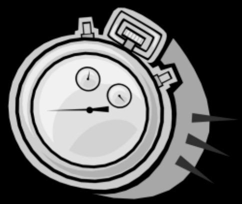


# Agenda

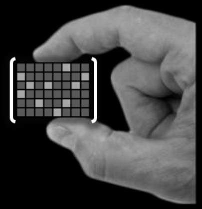
1. Analysis and synthesis signal models
   Two models are better than one
2. A few words on dictionary design
   On dictionaries and choices
3. Introduction to sparse representation
   Some background on sparsity and the K-SVD
4. The sparse dictionary model
   Introducing an efficient *and* adaptable dictionary!
5. **Why sparse dictionaries are good for you**
   Some uses and applications
6. Summary and conclusions

# Sparse versus Explicit Dictionaries



**Efficient**

Enable processing of larger signals



**Compact**

Easy to store and transmit

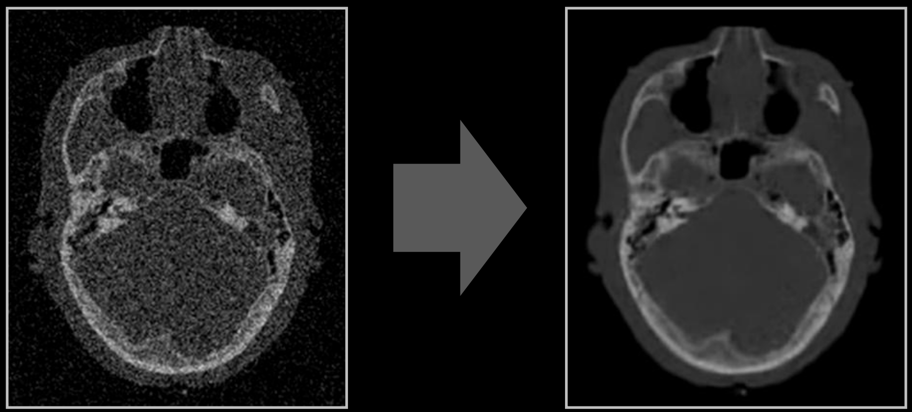
**Stable**

Require less training examples due to reduced overfitting

**Structured**

Allow meaningful constructions to be described

# Application: 3-D Image Denoising

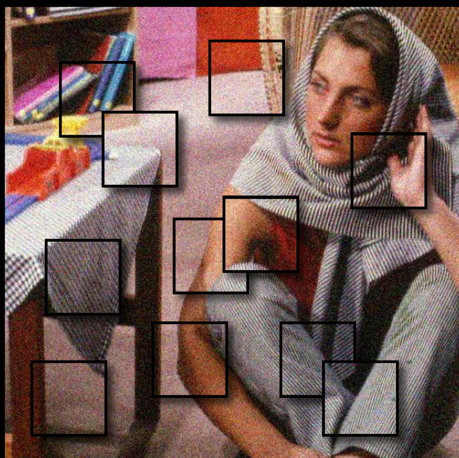


Challenge:

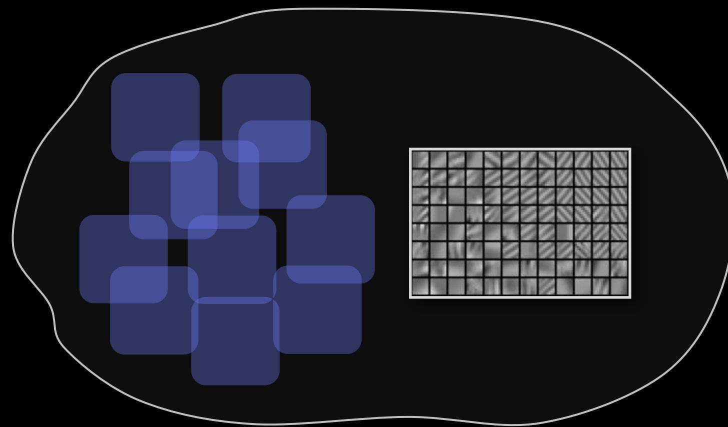
3-D signals much larger than 2-D =
Larger time & memory requirements

* Images curtsey of the NIH

# The K-SVD-Denoise Algorithm



PSNR = 20.2dB



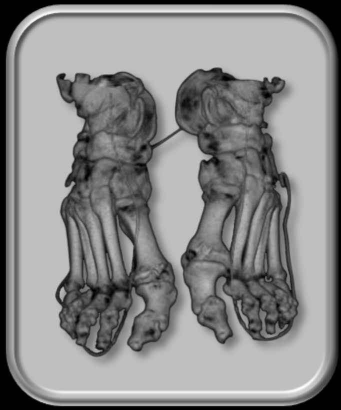
PSNR = 30.3dB

Extract Noisy Blocks → Train K-SVD Dictionary → Denoise Using Trained Dictioanry → Rearrange and Average Denoised Blocks

# Experiment Setup

Test CT volumes:

Male-head



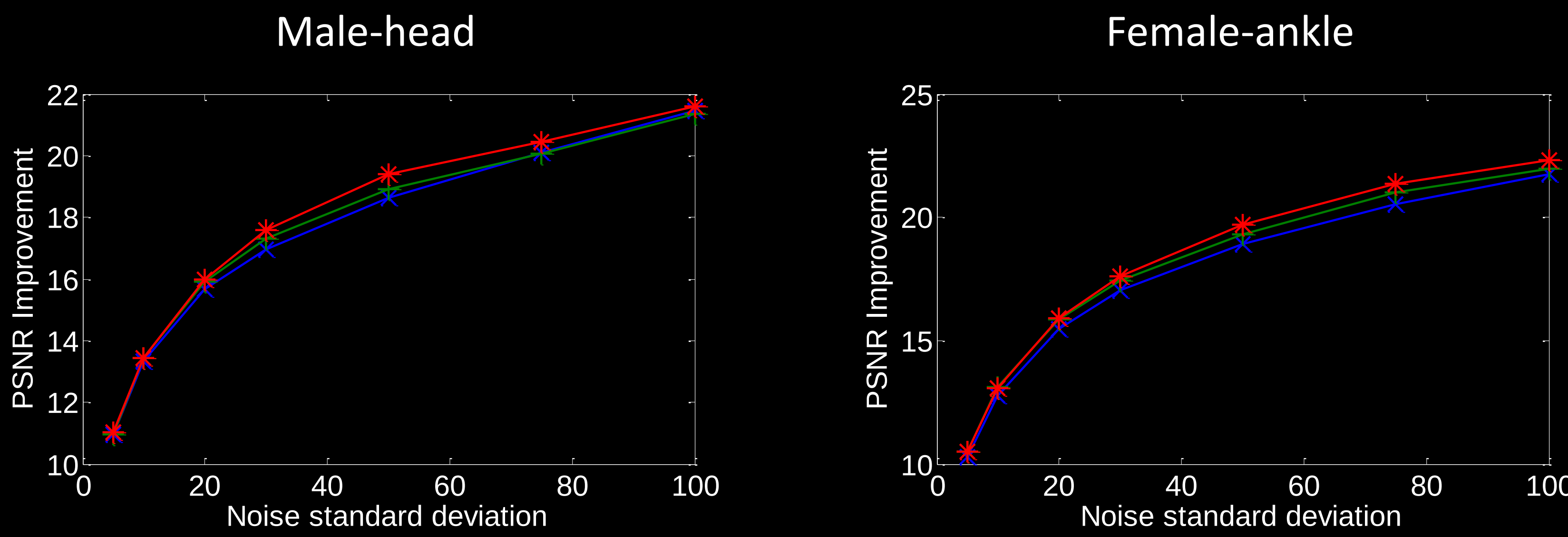
Female-ankle

| | |
|---|---|
| Block size | 8 x 8 x 8 |
| Dictionary size | 512 x 1000 |
| Atom sparsity (sparse K-SVD) | 16 |
| No. training signals | 80,000 |
| Training iterations | 15 |
| Step size | 2 |
| Base dictionary (sparse K-SVD) | O-DCT |

# Denoising Results

Graphs show the improvement in PSNR over the noisy volume (in dB)

Sparse K-SVD
Standard K-SVD
Overcomplete DCT



# Denoising Running Times: Female-Ankle

| | Overcomplete DCT | Sparse K-SVD | K-SVD |
|---|---|---|---|
| $\sigma=10$ | 10:32 | 27:49 | 2:02:28 |
| $\sigma=20$ | 4:28 | 11:32 | 48:36 |
| $\sigma=30$ | 2:45 | 7:09 | 29:11 |
| $\sigma=50$ | 1:34 | 4:23 | 16:59 |
| $\sigma=75$ | 1:06 | 3:19 | 11:44 |
| $\sigma=100$ | 0:53 | 2:52 | 9:36 |

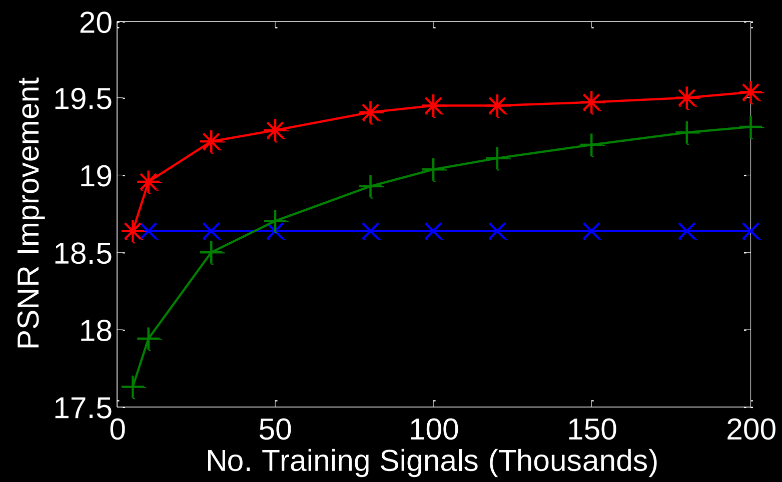
** Test platform: Intel Core 2 (single thread), Matlab 2010a, combined C+Matlab implementation.*

# Denoising Results versus # Training Signals

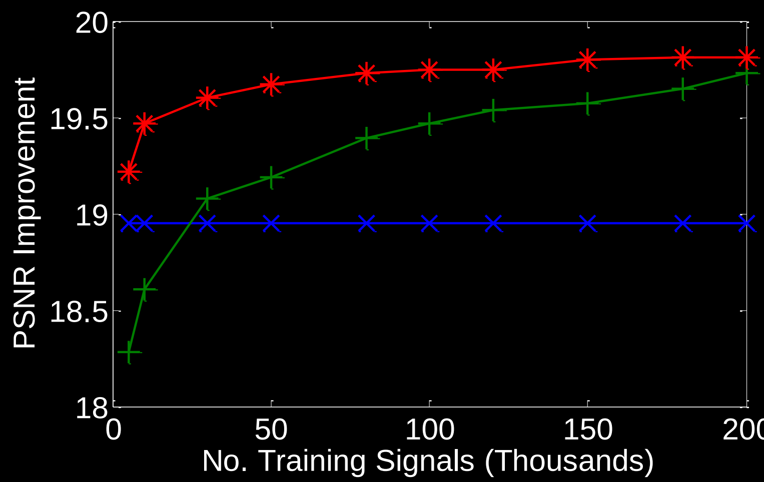
Graphs show the improvement in PSNR over the noisy volume (in dB). Input noise is σ=50 (PSNR=14.15dB)

Sparse K-SVD
Standard K-SVD
Overcomplete DCT

Male-head



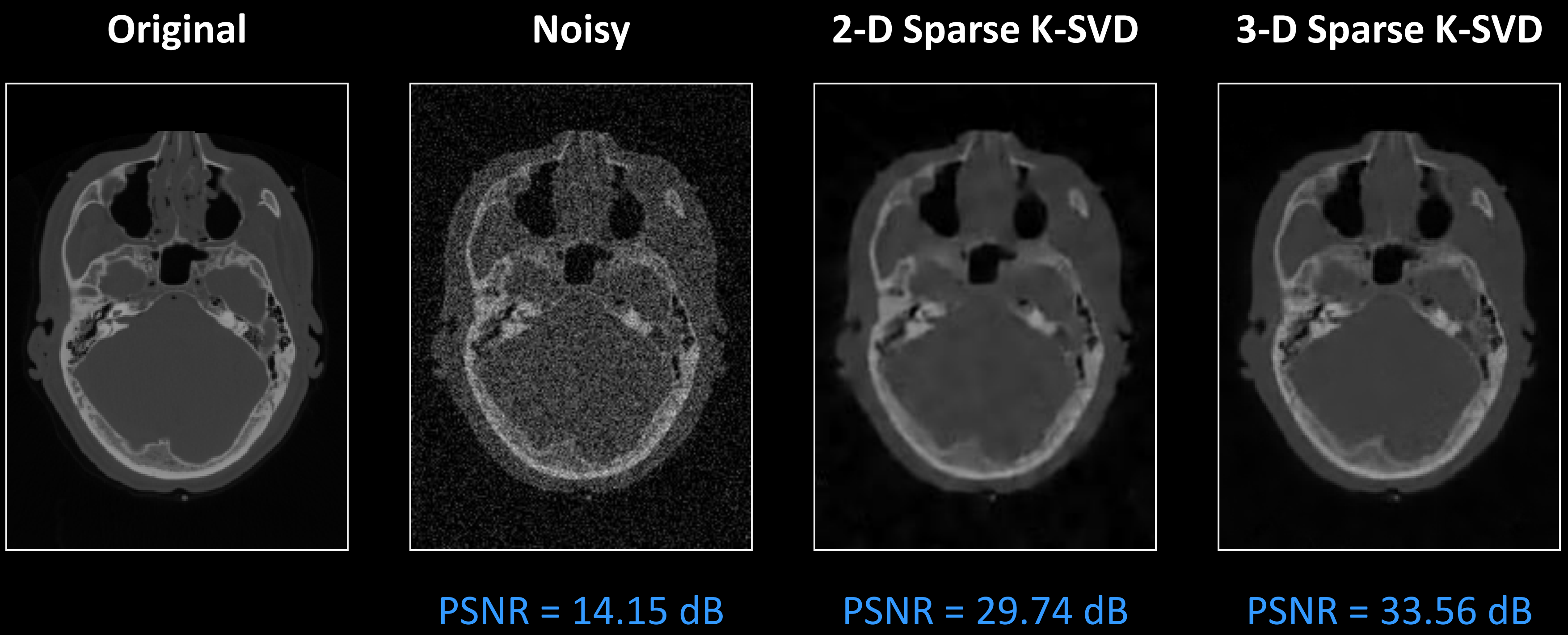

Female-ankle



# Some Actual Results

Noise with standard deviation $\sigma = 50$

Showing slice from Male-Head:



# Application: Image Compression


In collaboration with

Inbal Horev

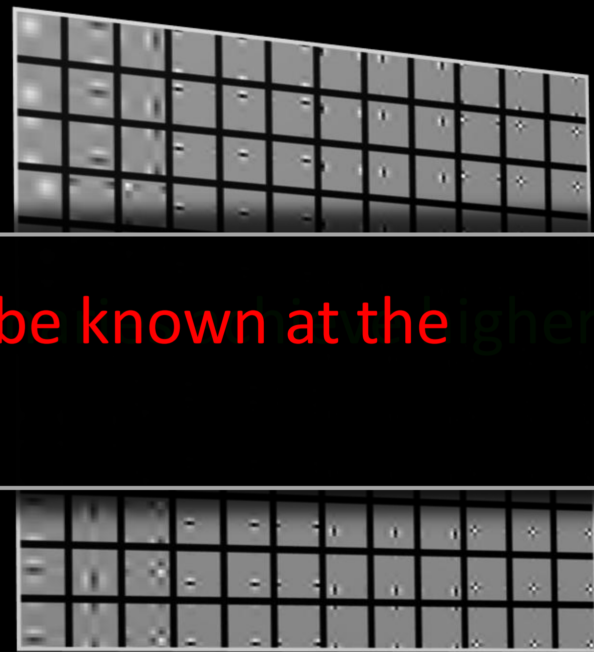
Ori Bryt

Basic concept:

Apply a (possibly lossy) sparsifying transform which reduces the entropy of the representation

# Application: Image Compression

Previously …



Caveat: the dictionary must be known at the decoder!

**JPEG: DCT dictionary**

**Linear approximation**

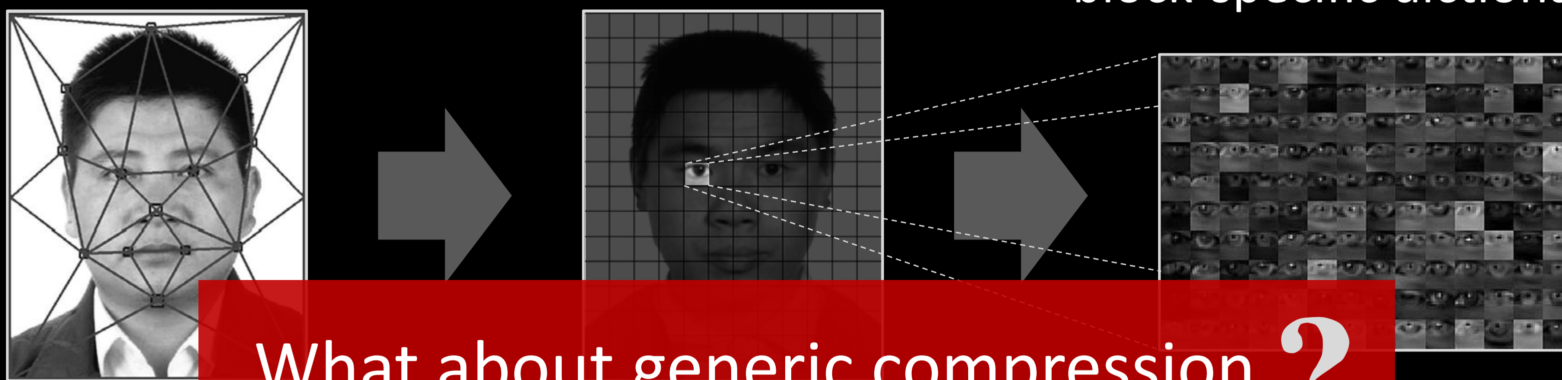
**JPEG-2K: Wavelet dictionary**

**Non-linear approximation**

# Solution 1: The Facial Compression Scheme

[Bryt & Elad '08]

Geometrical alignment → Partitioning → Sparse coding using fixed block-specific dictionaries

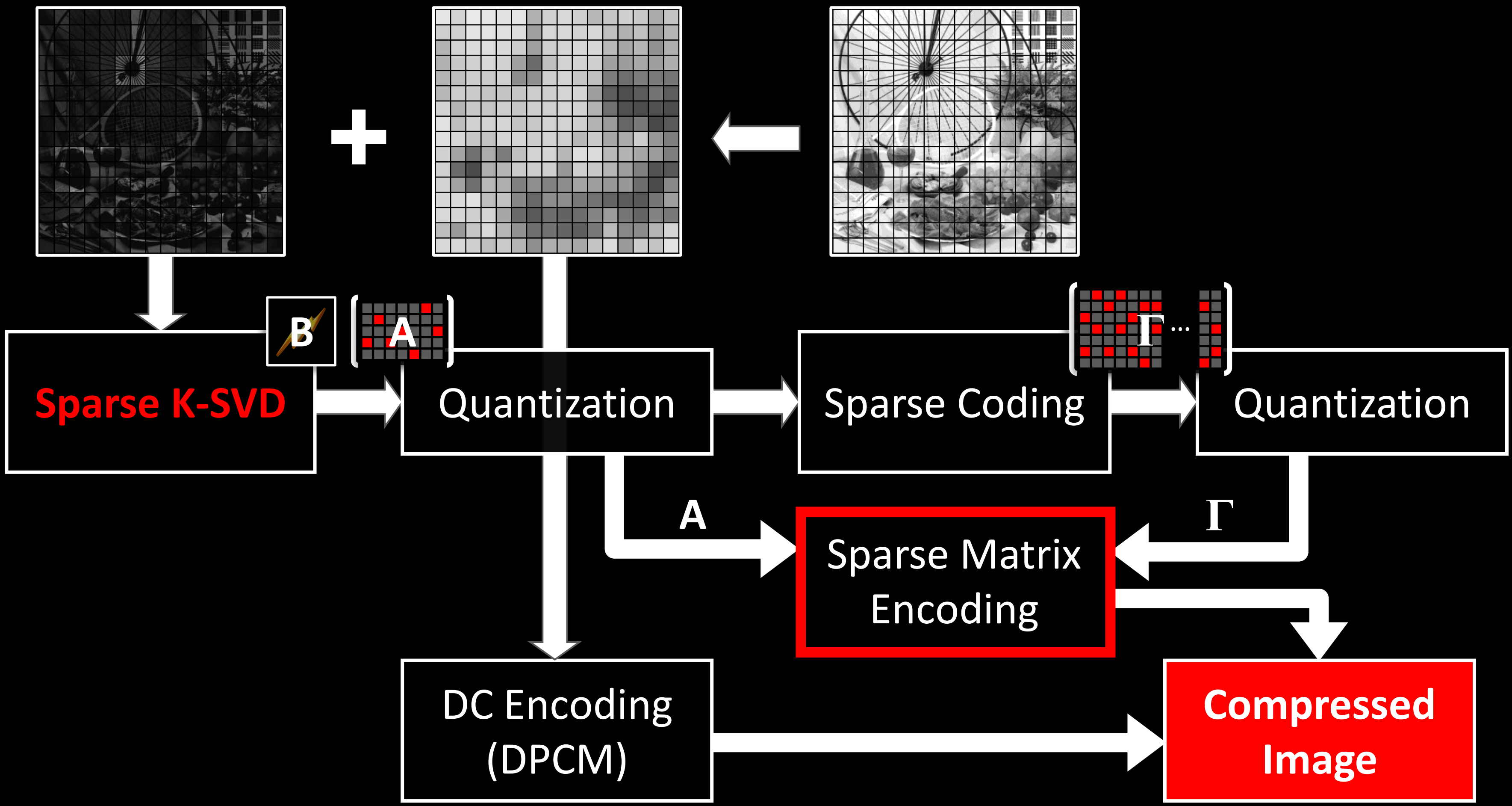


**What about generic compression?**

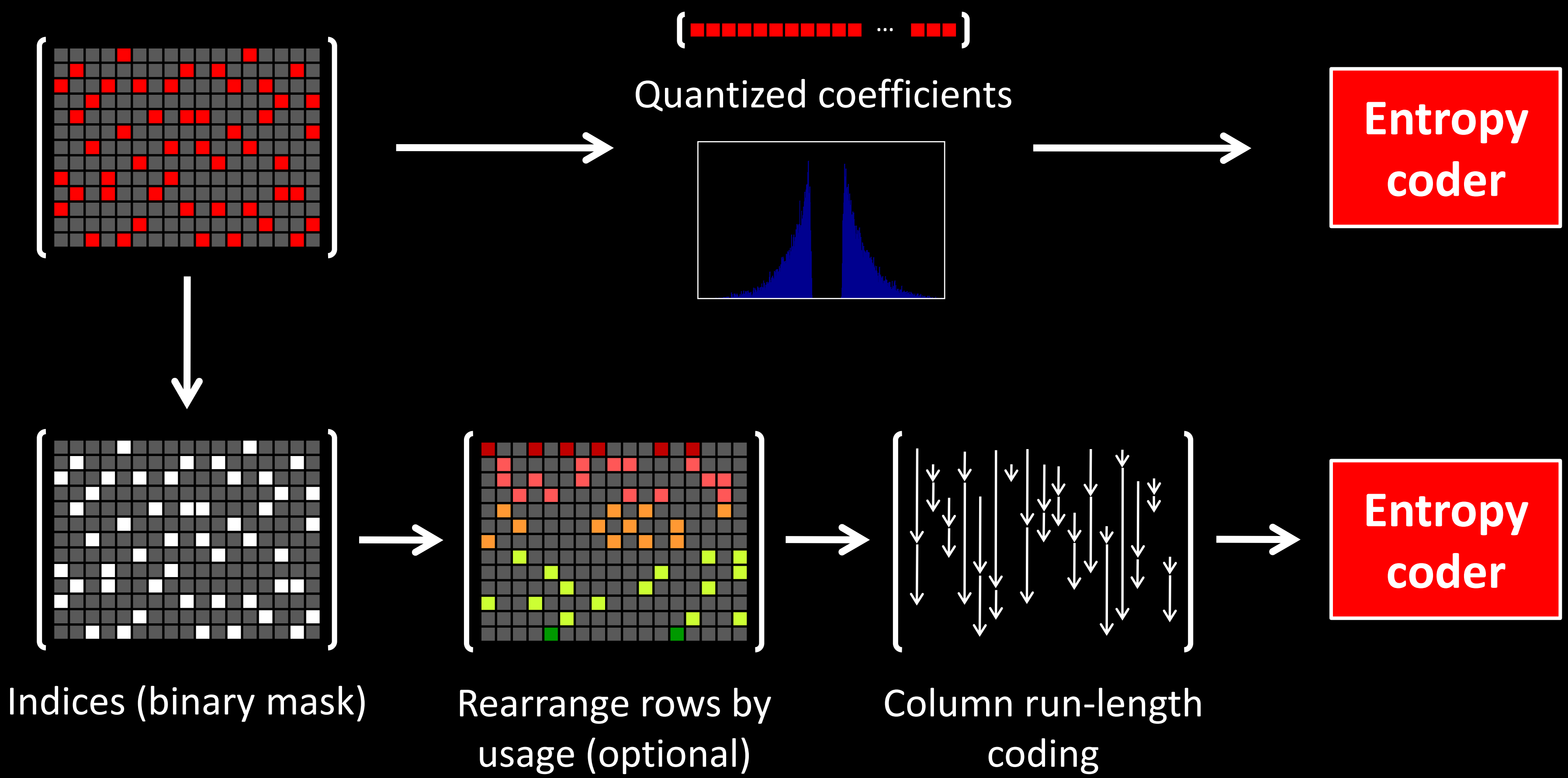
Original — Adaptive

24.8 dB | 26.3 dB | **33.1 dB**

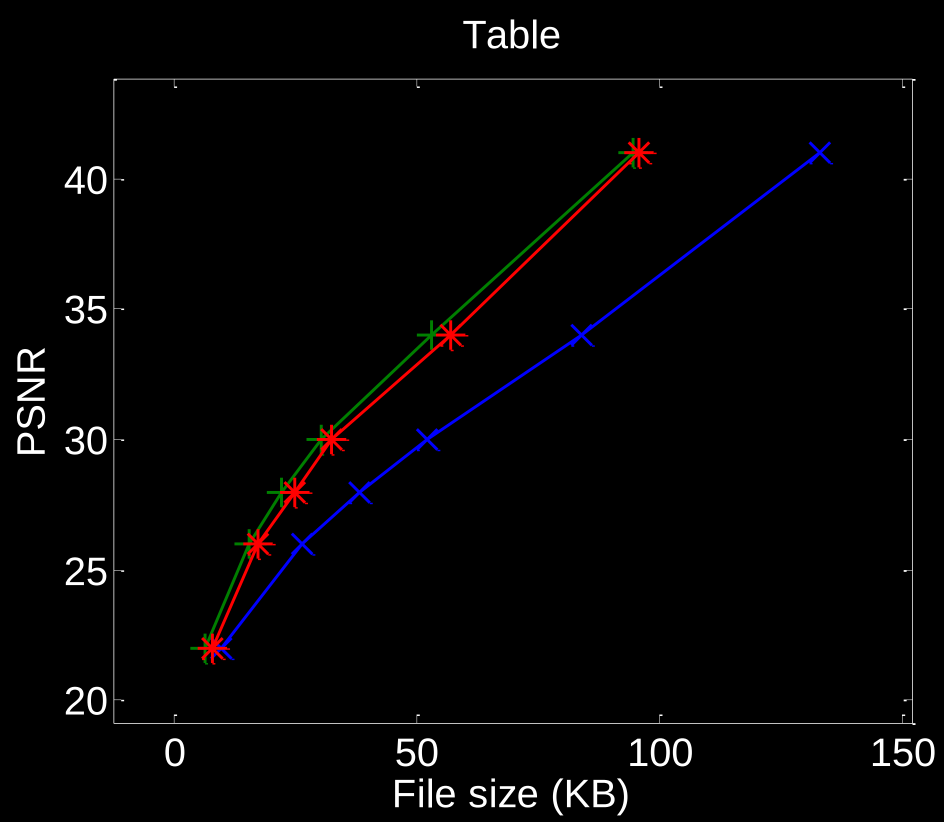
# The Sparse K-SVD Compression Scheme



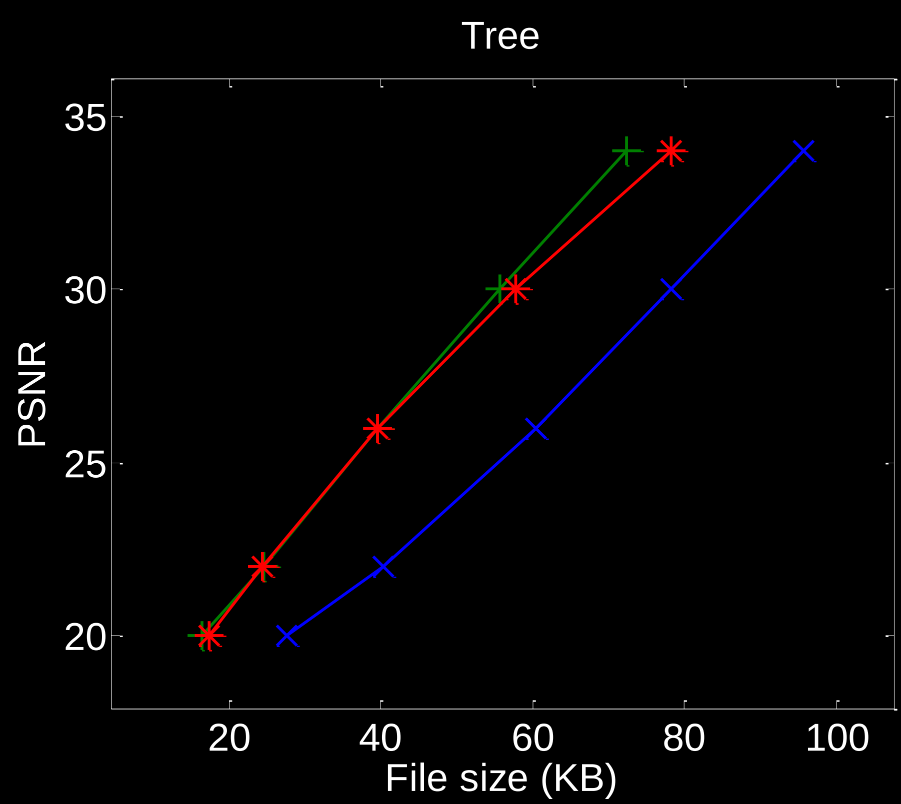
# Sparse Matrix Encoding



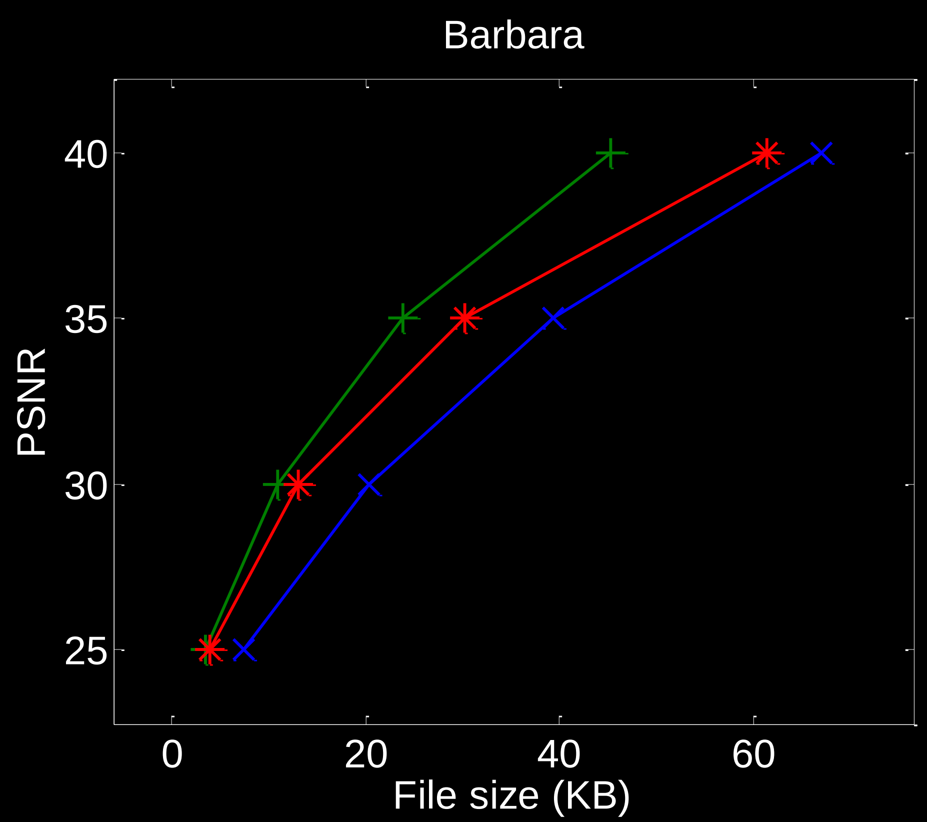
# Preliminary Results



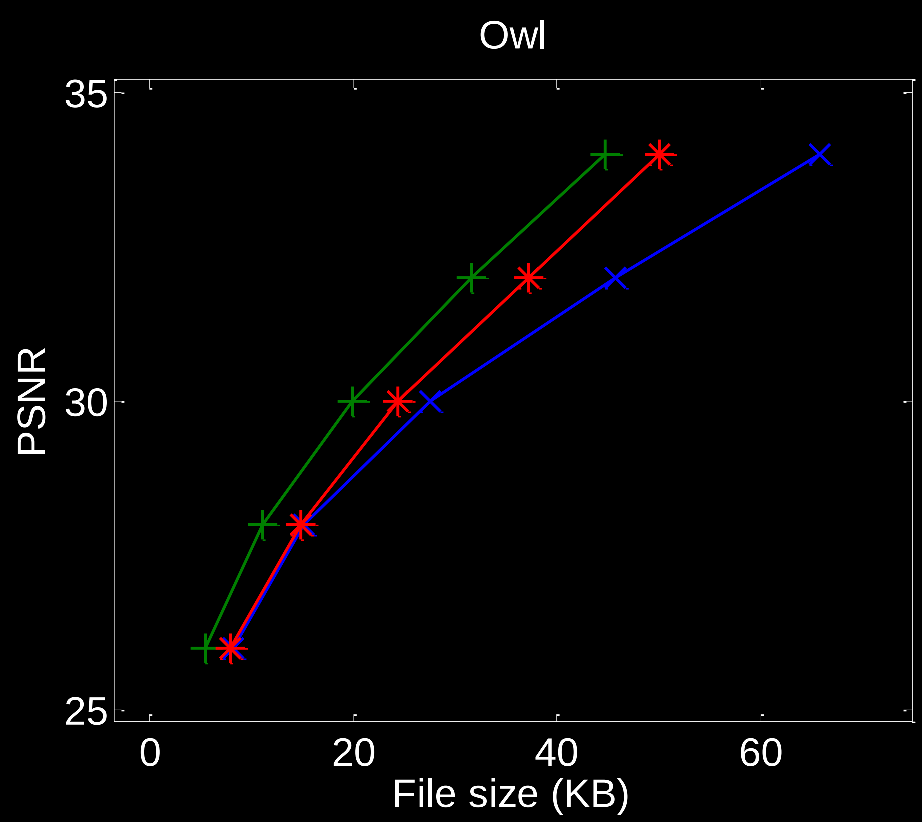

# Preliminary Results



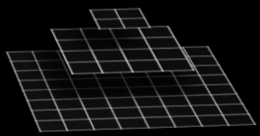

# Preliminary Results




# Preliminary Results

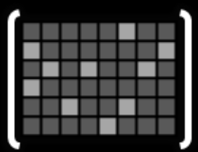



# Further Improvements



Two-scale (easier) and multi-scale dictionaries

Possibly locally- adaptive



Combining signal-adaptive and fixed dictionaries

Fixed dictionaries shared by encoder and decoder

Reduced index entropy

Using specialized sparse-coding algorithm

Image deblocking

Strength may be locally adapted based on # of coefficients used

Compression in a transformed domain

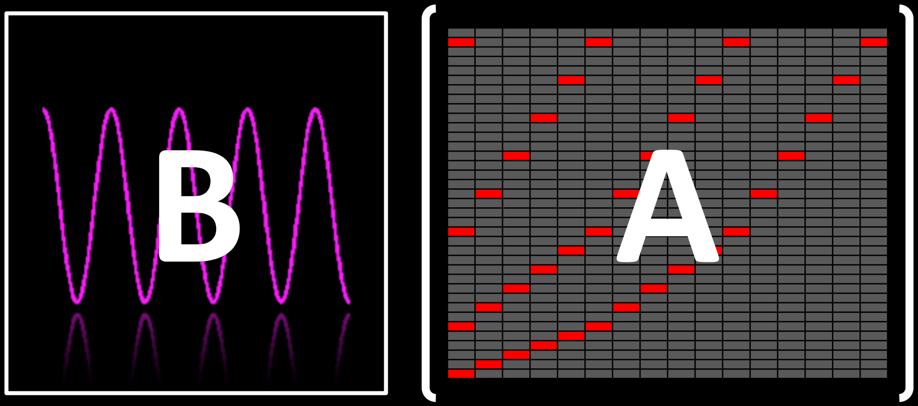
E.g. a wavelet transform of the image

# Other Applications: Music Transcription

Goal: train a dictionary of musical notes for a given instrument

Observation: dictionary has a known sparse structure – each musical note is the superposition of a specific set of frequencies (base frequency + overtones)

Michal Genussov




[Genussov & Cohen '10]

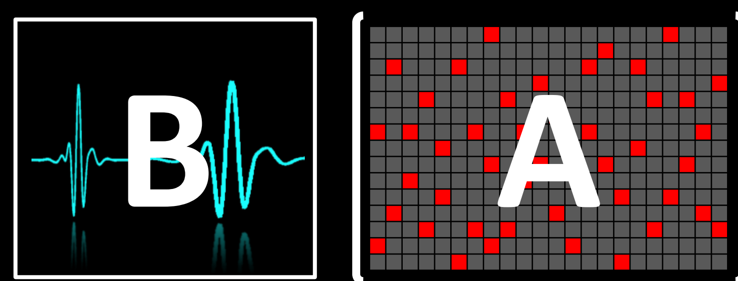
# Other Applications: Multiscale Dictionaries

Goal: train a dictionary with a multi-scale structure

Motivation: multi-scale dictionaries efficiently describe phenomena at various scales, can exploit inter-scale dependencies, and may be independent of block size

Idea: train **A** over a multi-scale base dictionary. Localization / inter-scale regularity can be enforced

Boaz Ophir



# Agenda

1. Analysis and synthesis signal models
   Two models are better than one
2. A few words on dictionary design
   On dictionaries and choices
3. Introduction to sparse representation
   Some background on sparsity and the K-SVD
4. The sparse dictionary model
   Introducing an efficient *and* adaptable dictionary!
5. Why sparse dictionaries are good for you
   Some uses and applications
6. **Summary and conclusions**

# Summary

- Signal models play a critical role in low-level signal and image processing
- Two competing models: analysis and synthesis, synthesis vastly studied, analysis an emerging direction
- Designing a good dictionary is critical for the success of either model
- Traditional choice: analytic versus trained dictionaries

# Summary

- We can do better with parametric dictionaries! The sparse dictionary bridges the gap between the two options
- The Sparse K-SVD algorithm efficiently trains sparse dictionaries
- Applications: denoising, compression, specialized dictionary structures.

# Thank You!

Questions?

# Extra slides

# Double Sparsity versus Double Cardinality

**Question:** if x is sparse over $\mathbf{BA}$...

$$x \approx \mathbf{B}\mathbf{A}\gamma$$

and $\mathbf{A}$ and $\gamma$ are both sparse, then x is sparse over $\mathbf{B}$ too!

So, why do we need $\mathbf{A}$?

# Double Sparsity versus Double Cardinality

**Answer:** $\mathrm{x}$ is much less sparse over $\mathbf{B}$ than over $\mathbf{BA}$!

$$\|a_i\|_0 = p \ , \ \ \|\gamma\|_0 = q \quad \Rightarrow \quad \|\mathbf{A}\gamma\|_0 \approx pq$$

- Sparse representation methods rely on sparsity for their success. So, they are less effective when operating directly over $\mathbf{B}$.
- The matrix $\mathbf{A}$ allows us to sidestep this issue, taking us back to the sparse zone!

# Double Sparsity versus Double Cardinality

**Example:** The denoising tradeoff

$$\mathbf{y} \approx \mathbf{D}\gamma$$

*Noisy signal* (y)

*Recovered representation* ($\gamma$)

How many non-zeros should $\gamma$ have?

| Higher cardinality: More accurate description of the original signal | ⟷ | Lower cardinality: Less remaining noise = stronger denoising |
|---|---|---|

# Double Sparsity versus Double Cardinality

Denoise with overcomplete-DCT

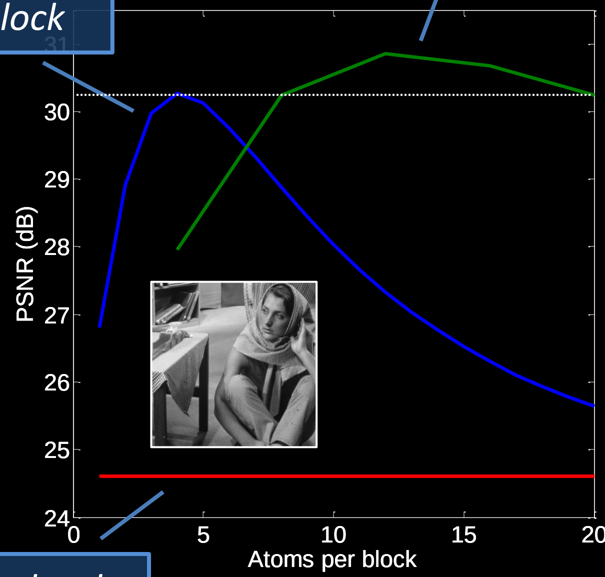
Increased cardinality rapidly deteriorates performance

Denoise with a sparse dictionary over an O-DCT base dictionary

More atoms can be used, while noise is much better controlled

*Denoising directly over an O-DCT dictionary with variable atom# / block*

*Denoising over sparse dictionary with O-DCT base dictionary. Plotting effective atom# / block over the O-DCT dictionary*




*Noise level*

# Proof of Equivalence for $a_k$

Let $\mathbf{X}$ and $\mathbf{Y}$ be matrices, and let $\mathrm{u}$ and $\mathrm{v}$ be vectors. Also, assume that $\mathrm{v}^T\mathrm{v}=1$. Then we can show that

$$\left\|\mathbf{X}-\mathbf{Y}\mathrm{u}\mathrm{v}^T\right\|_F^2 = \left\|\mathbf{X}\mathrm{v}-\mathbf{Y}\mathrm{u}\right\|_F^2 + \mathrm{Tr}\left(\mathbf{X}^T\mathbf{X}\right) - \mathrm{v}^T\mathbf{X}^T\mathbf{X}\mathrm{v}$$

$$= \left\|\mathbf{X}\mathrm{v}-\mathbf{Y}\mathrm{u}\right\|_F^2 + f\left(\mathbf{X},\mathrm{v}\right)$$

So, for $\tilde{\gamma}_k{}^T\tilde{\gamma}_k = 1$, we have

$$\|\tilde{\mathbf{E}}_k - \mathbf{B}\mathrm{a}_k\tilde{\gamma}_k^T\|_F^2 = \|\tilde{\mathbf{E}}_k\tilde{\gamma}_k - \mathbf{B}\mathrm{a}_k\|_F^2 + f\left(\tilde{\mathbf{E}}_k,\tilde{\gamma}_k\right)$$

*Independent of $a_k$!*

# Proof of Equivalence for $a_k$

Since we are minimizing for $a_k$, this means that the two minimization problems are equivalent:

$$\min_{a_k} \| \tilde{\mathbf{E}}_k - \mathbf{B} a_k \tilde{\gamma}_k^T \|_F^2 \quad \text{s.t.} \quad \| a_k \|_0 \le P$$

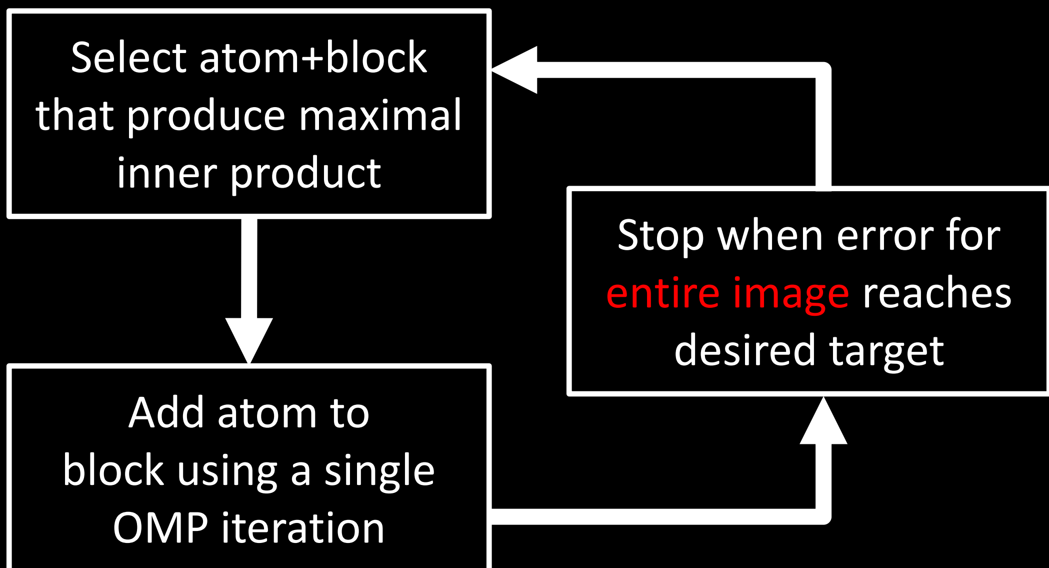
$$\Updownarrow$$

$$\min_{a_k} \| \tilde{\mathbf{E}}_k \tilde{\gamma}_k^T - \mathbf{B} a_k \|_F^2 \quad \text{s.t.} \quad \| a_k \|_0 \le P$$

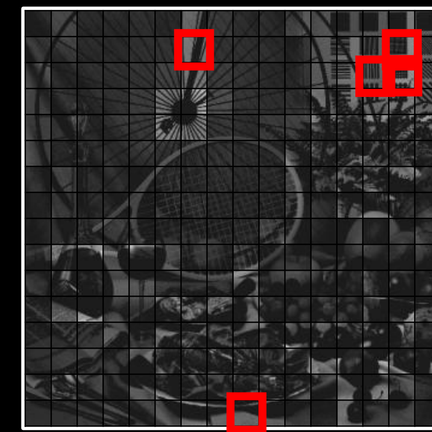
We note that the normalization assumption on $\tilde{\gamma}_k$ is easily overcome by transferring energy between $a_k$ and $\tilde{\gamma}_k$.

# Sparse Coding Stage: Global OMP

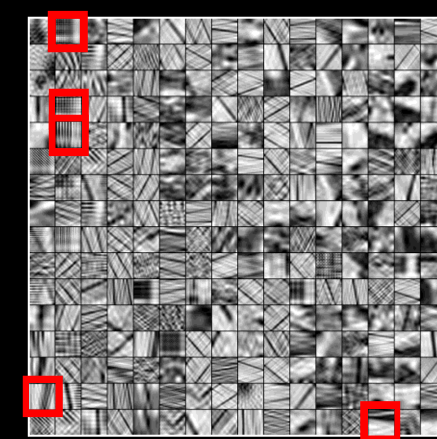
Motivation: adapt the distribution of coefficients to the complexity of the block




Efficiency is achieve by implementing the computations locally only for the affected block



Image



Dictionary

# Global OMP + Quantization

We overcome the PSNR reduction caused by quantization using iteration: