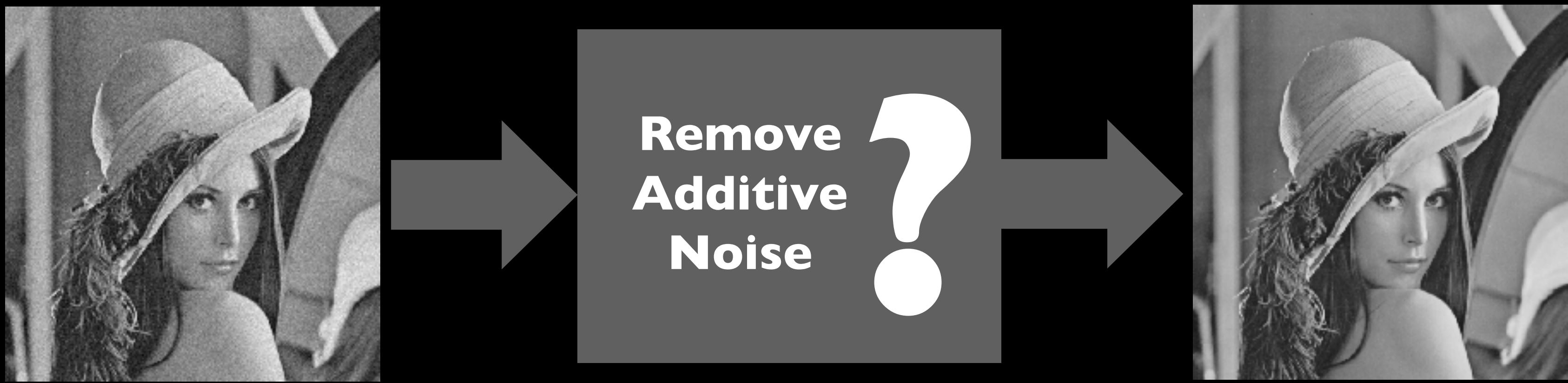
BIL 717
Image Processing

Erkut Erdem
Dept. of Computer Engineering
Hacettepe University

Sparse Coding

Acknowledgement: The slides are adapted from the ones prepared by M. Elad.

Noise Removal?



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

Denoising By Energy Minimization

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

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

Relation to measurements

Prior or regularization

$\underline{y}$: Given measurements

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

- This is in-fact a Bayesian point of view, adopting the Maximum-A-posteriori 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 G(x)

During the past several decades we have made all sort of guesses about the prior G(x) for images:

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

Energy

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

Smoothness

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

Adapt+ Smooth

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

Robust Statistics

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

Total-Variation

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

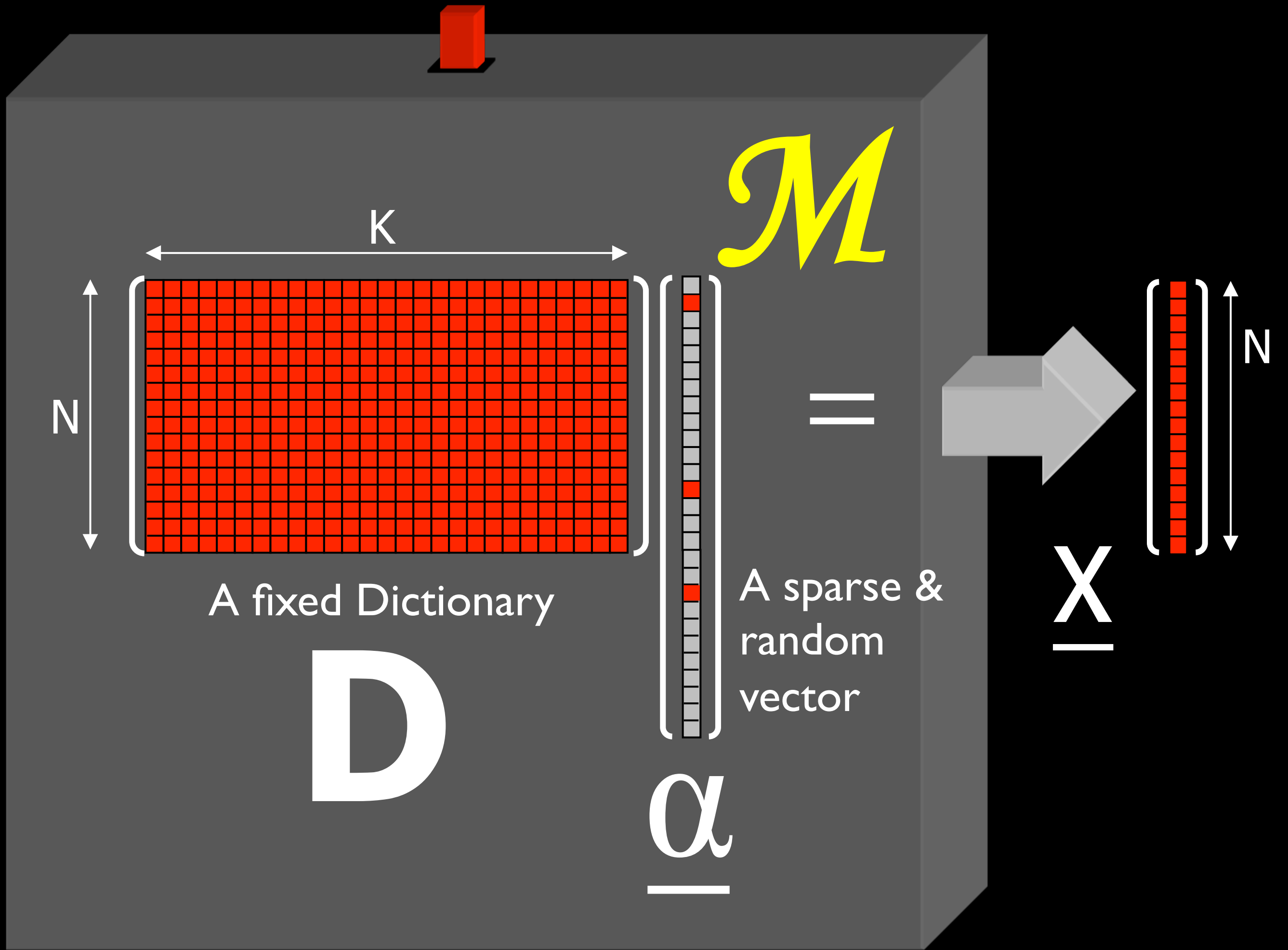
Wavelet Sparsity

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

Sparse & Redundant

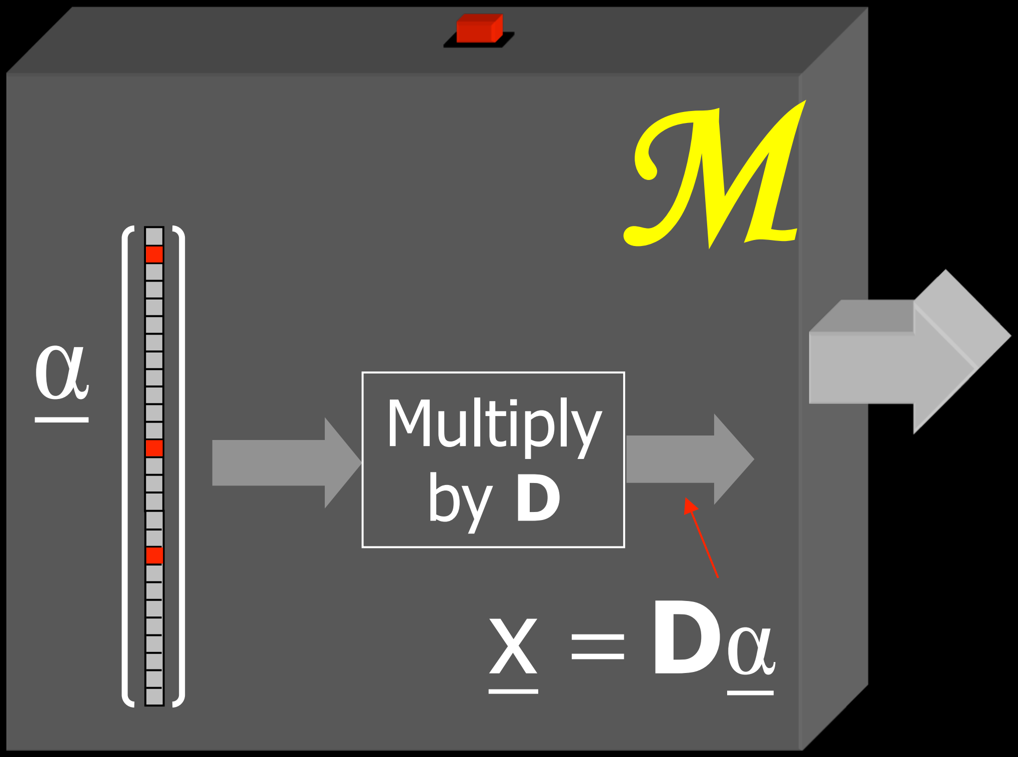
- Hidden Markov Models,
- Compression algorithms as priors,
- …

Sparse Modeling of Signals



- 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.
- We shall refer to this model as *Sparseland*

Sparseland Signals are Special



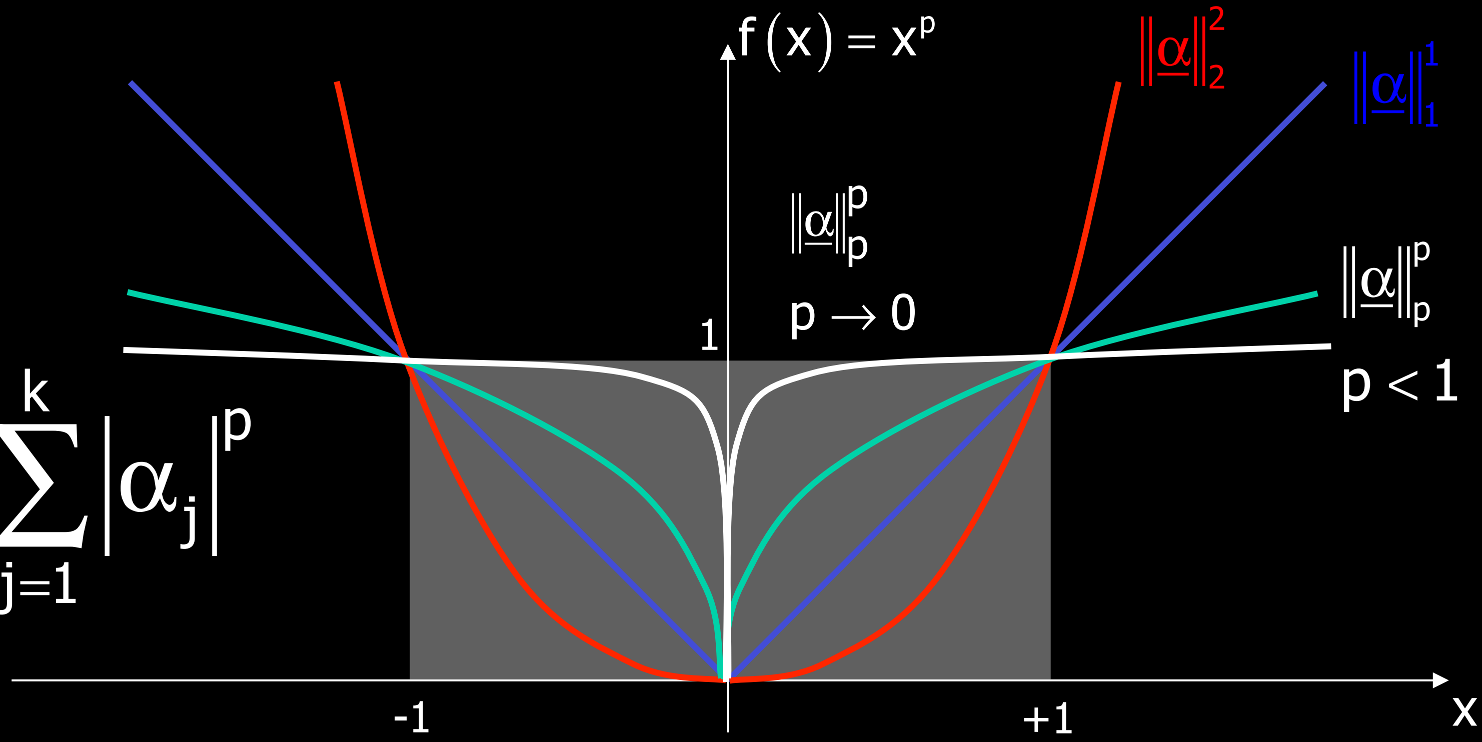
Interesting Model:

- **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 union of many low-dimensional Gaussians.
- **Familiar:** We have been using this model in other context for a while now (wavelet, JPEG, …).

Sparse & Redundant Rep. Modeling?

Our signal model is thus: $\underline{x} = \mathbf{D}\underline{\alpha}$ where $\underline{\alpha}$ is sparse

Sparse & Redundant Rep. Modeling?

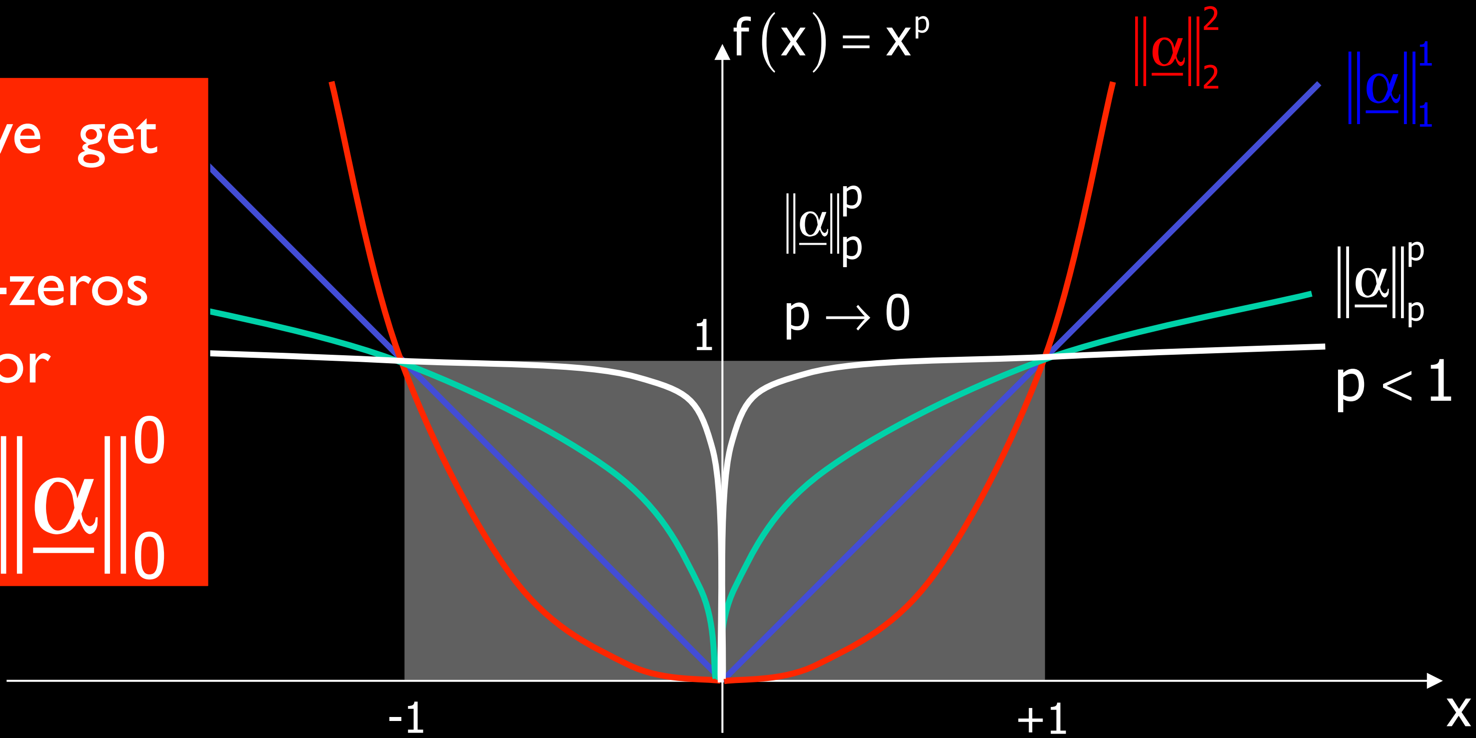


$$\|\underline{\alpha}\|_p^p = \sum_{j=1}^{k} |\alpha_j|^p$$

Our signal model is thus: $\underline{x} = \mathbf{D}\underline{\alpha}$ where $\underline{\alpha}$ is sparse

Sparse & Redundant Rep. Modeling?

As p → 0 we get a count of the non-zeros in the vector → $\|\underline{\alpha}\|_0^0$



Our signal model is thus: $\underline{x} = \mathbf{D}\underline{\alpha}$ where $\|\underline{\alpha}\|_0^0 \leq L$

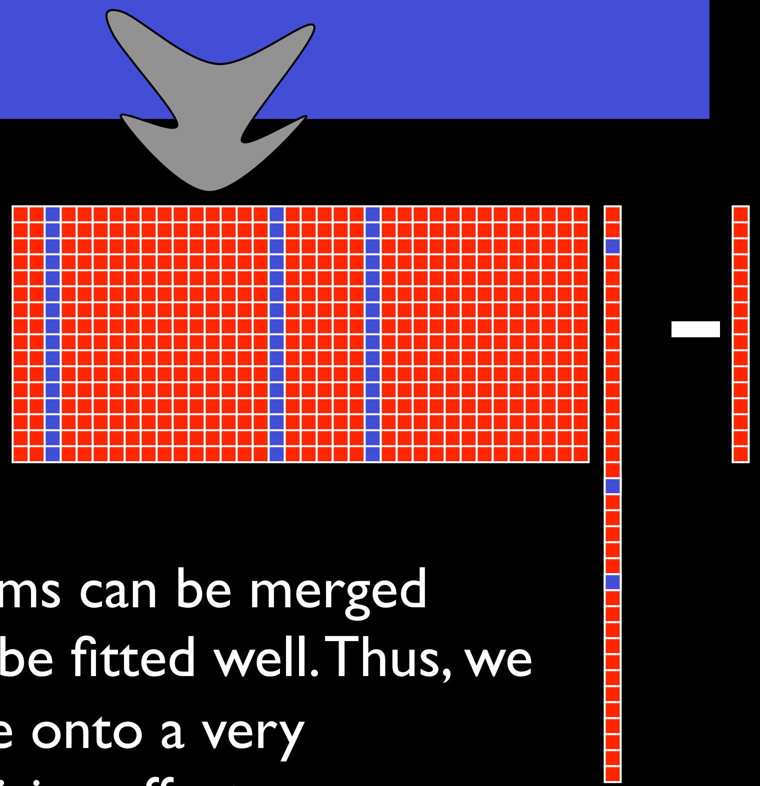
Back to Our MAP Energy Function

- L_0 norm effectively counts the number of non-zeros in $\underline{\alpha}$.

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

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

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



- The core idea: while few (L out of K) atoms can be merged to form the true signal, the noise cannot be fitted well. Thus, we obtain an effective projection of the noise onto a very low-dimensional space, thus getting denoising effect.

Wait! There are Some Issues

- **Numerical Problems:** How should we solve or approximate the solution of the problem

$$\min_{\underline{\alpha}} \left\|\mathbf{D}\underline{\alpha} - \underline{y}\right\|_2^2 \text{ s.t. } \left\|\underline{\alpha}\right\|_0^0 \le L \quad \text{or} \quad \min_{\underline{\alpha}} \left\|\underline{\alpha}\right\|_0^0 \text{ s.t. } \left\|\mathbf{D}\underline{\alpha} - \underline{y}\right\|_2^2 \le \varepsilon^2$$

$$\text{or} \quad \min_{\underline{\alpha}} \lambda\left\|\underline{\alpha}\right\|_0^0 + \left\|\mathbf{D}\underline{\alpha} - \underline{y}\right\|_2^2$$

- **Theoretical Problems:** Is there a unique sparse representation? If we are to approximate the solution somehow, how close will we get?

- **Practical Problems:** What dictionary **D** should we use, such that all this leads to effective denoising? Will all this work in applications?

To Summarize So Far …

Image denoising (and many other problems in image processing) requires a model for the desired image

What can we do?

Use a model for signals/images based on sparse and redundant representations

Great! No?

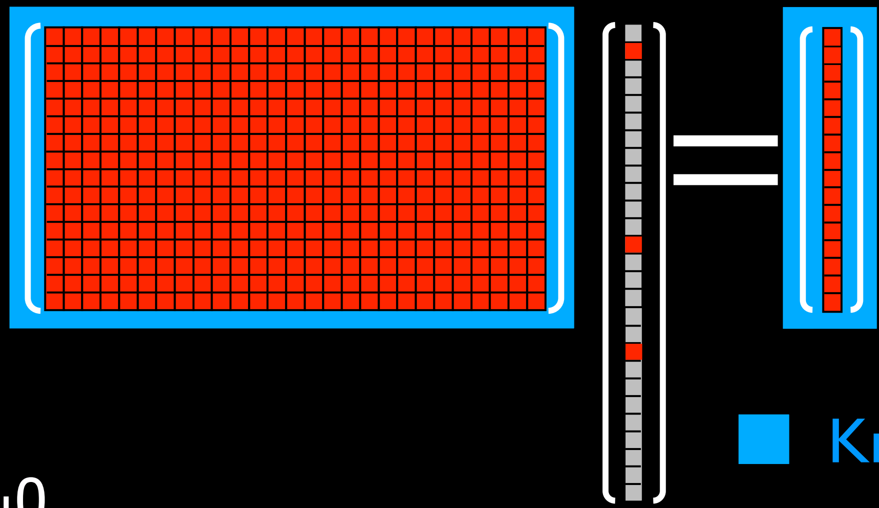
There are some issues:

1. Theoretical
2. How to approximate?
3. What about **D**?

Lets Start with the Noiseless Problem

Suppose we build a signal by the relation

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



■ Known

We aim to find the signal's representation:

$$\underline{\hat{\alpha}} = \underset{\underline{\alpha}}{\text{ArgMin}} \|\underline{\alpha}\|_0^0 \quad \text{s.t.} \quad \underline{x} = \mathbf{D}\underline{\alpha}$$

Uniqueness

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

It might happen that eventually $\|\underline{\hat{\alpha}}\|_0^0 < \|\underline{\alpha}\|_0^0$.

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

$$\underline{\hat{\alpha}} = \underset{\underline{\alpha}}{\text{ArgMin}} \left\| \underline{\alpha} \right\|_0^0 \ \text{ s.t. } \ \underline{x} = \mathbf{D}\underline{\alpha}$$

Uniqueness

If we found a representation that satisfy

$$\left\| \underline{\hat{\alpha}} \right\|_0 < \frac{\sigma}{2}$$

Then necessarily it is unique (the sparsest).

Donoho & E. ('02)

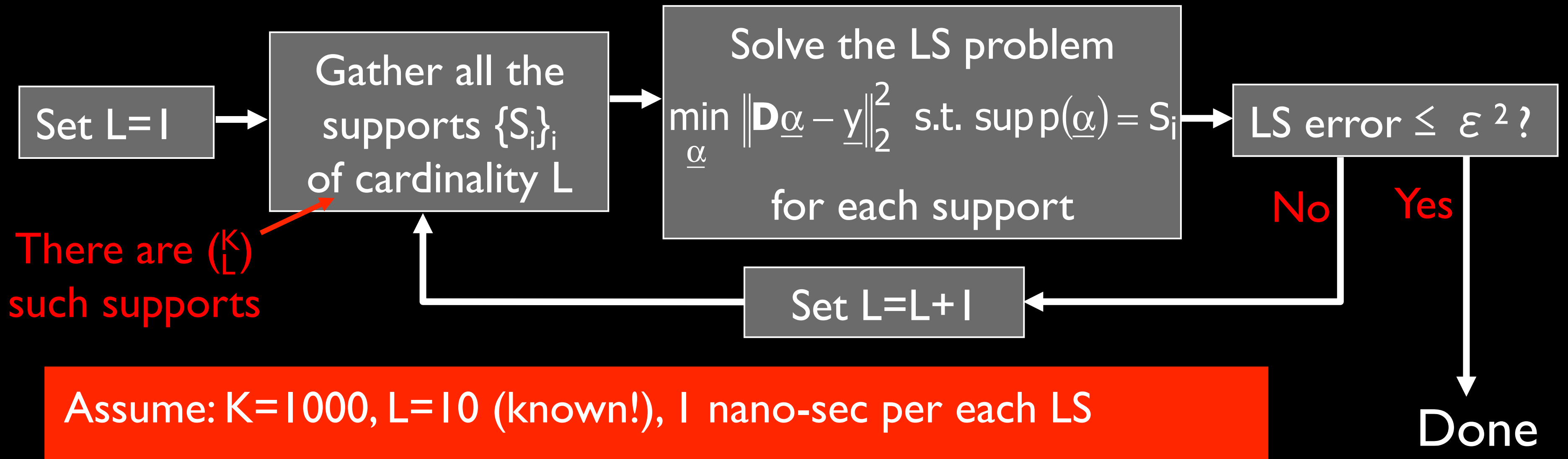
This result implies that if $\mathcal{M}$ generates signals using "sparse enough" $\underline{\alpha}$, the solution of the above will find it exactly.

Our Goal

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

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

Here is a recipe for solving this problem:



Assume: K=1000, L=10 (known!), 1 nano-sec per each LS

We shall need ~8e+6 years to solve this problem !!!!!

Lets Approximate

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

Relaxation methods

Smooth the L_0 and use continuous optimization techniques

Greedy methods

Build the solution one non-zero element at a time

Relaxation – The Basis Pursuit (BP)

Instead of solving

$$\underset{\underline{\alpha}}{\text{Min}} \ \|\underline{\alpha}\|_0^0 \ \text{ s.t. } \ \|\mathbf{D}\underline{\alpha} - \underline{y}\|_2 \leq \varepsilon$$

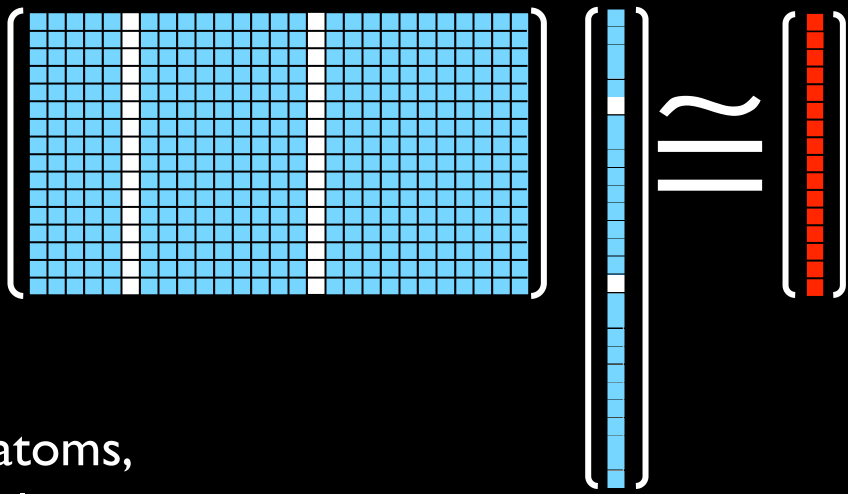
→

Solve Instead

$$\underset{\underline{\alpha}}{\text{Min}} \ \|\underline{\alpha}\|_1 \ \text{ s.t. } \ \|\mathbf{D}\underline{\alpha} - \underline{y}\|_2 \leq \varepsilon$$

- This is known as the Basis-Pursuit (BP) [Chen, Donoho & Saunders ('95)].
- The newly defined problem is convex (quad. programming).
- Very efficient solvers can be deployed:
  - Interior point methods [Chen, Donoho, & Saunders ('95)] [Kim, Koh, Lustig, Boyd, & D. Gorinevsky (`07)].
  - Sequential shrinkage for union of ortho-bases [Bruce et.al. ('98)].
  - Iterative shrinkage [Figuerido & Nowak ('03)] [Daubechies, Defrise, & De-Mole ('04)] [E. ('05)] [E., Matalon, & Zibulevsky ('06)] [Beck & Teboulle (`09)] …

Go Greedy: Matching Pursuit (MP)



- The MP is one of the greedy algorithms that finds one atom at a time [Mallat & Zhang ('93)].
- Step 1: find the one atom that best matches the signal.
- Next steps: given the previously found atoms, find the next **one** to best fit the residual.
- The algorithm stops when the error $\left\| \mathbf{D}\underline{\alpha} - \underline{y} \right\|_2$ is below the destination threshold.
- The Orthogonal MP (OMP) is an improved version that re-evaluates the coefficients by Least-Squares after each round.

Pursuit Algorithms

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

There are various algorithms designed for approximating the solution of this problem:

- Greedy Algorithms: Matching Pursuit, Orthogonal Matching Pursuit (OMP), Least-Squares-OMP, Weak Matching Pursuit, Block Matching Pursuit [1993-today].
- Relaxation Algorithms: Basis Pursuit (a.k.a. LASSO), Dnatzig Selector & numerical ways to handle them [1995-today].
- Hybrid Algorithms: StOMP, CoSaMP, Subspace Pursuit, Iterative Hard-Thresholding [2007-today].
- …

Pursuit Algorithms

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

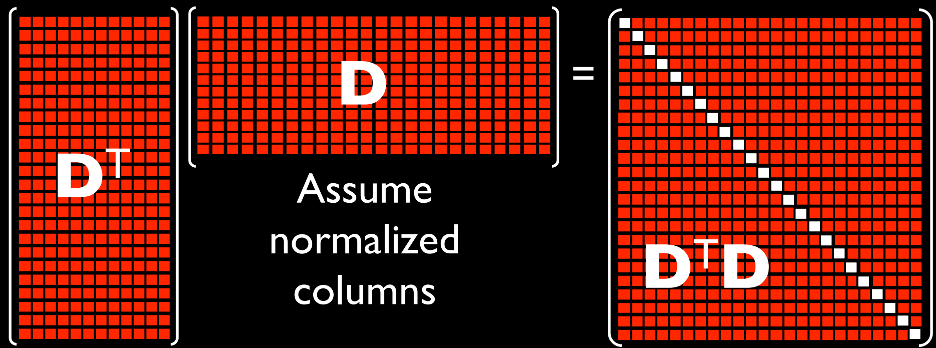
There are various algorithms designed for approximating the solution of this prob

- Greedy A MP), Least-Squ 993- today].
- Relaxatio numerical
- Hybrid Al Thresholdi
- …

Why should they work ?

The Mutual Coherence

- Compute



- The Mutual Coherence μ 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:

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

BP and MP Equivalence (No Noise)

$$\underline{\hat{\alpha}} = \underset{\underline{\alpha}}{\text{ArgMin}} \left\| \underline{\alpha} \right\|_0^0 \ \text{ s.t. } \ \underline{x} = \mathbf{D}\underline{\alpha}$$

BP and MP Equivalence (No Noise)

Equivalence

Given a signal $\underline{x}$ with a representation $\underline{x} = \mathbf{D}\underline{\alpha}$, assuming that $\|\underline{\alpha}\|_0^0 < 0.5\left(1 + 1/\mu\right)$, BP and MP are guaranteed to find the sparsest solution.

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

- MP and BP are different in general (hard to say which is better).
- The above result corresponds to the worst-case, and as such, it is too pessimistic.
- Average performance results are available too, showing much better bounds [Donoho (`04)] [Candes et.al. (‘04)] [Tanner et.al. (‘05)] [E. (‘06)] [Tropp et.al. (‘06)] … [Candes et. al. (‘09)].

BP Stability for the Noisy Case

$$\min_{\underline{\alpha}} \lambda \left\|\underline{\alpha}\right\|_1 + \left\|\mathbf{D}\underline{\alpha} - \underline{y}\right\|_2^2$$

BP Stability for the Noisy Case

Stability

Given a signal $\underline{y} = \mathbf{D}\underline{\alpha} + \underline{v}$ with a representation satisfying $\|\underline{\alpha}\|_0^0 < 1/3\mu$ and a white Gaussian noise $\underline{v} \sim N(0, \sigma^2 \mathbf{I})$, BP will show* stability, i.e.,

$$\|\hat{\underline{\alpha}}_{BP} - \underline{\alpha}\|_2^2 < \text{Const}(\lambda) \cdot \log K \cdot \|\underline{\alpha}\|_0^0 \cdot \sigma^2$$

Ben-Haim, Eldar & E. (‘09)

* With very high probability

- For σ=0 we get a weaker version of the previous result.
- This result is the oracle’s error, multuiplied by $C \cdot \log K$.
- Similar results exist for other pursuit algorithms (Dantzig Selector, Orthogonal Matching Pursuit, CoSaMP, Subspace Pursuit, …)

To Summarize So Far …

Image denoising (and many other problems in image processing) requires a model for the desired image

What do we do?

Use a model for signals/images based on sparse and redundant representations

Problems?

We have seen that there are approximation methods to find the sparsest solution, and there are theoretical results that guarantee their success.

What next?

The Dictionary **D** should be found somehow !!!

What Should D Be?

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

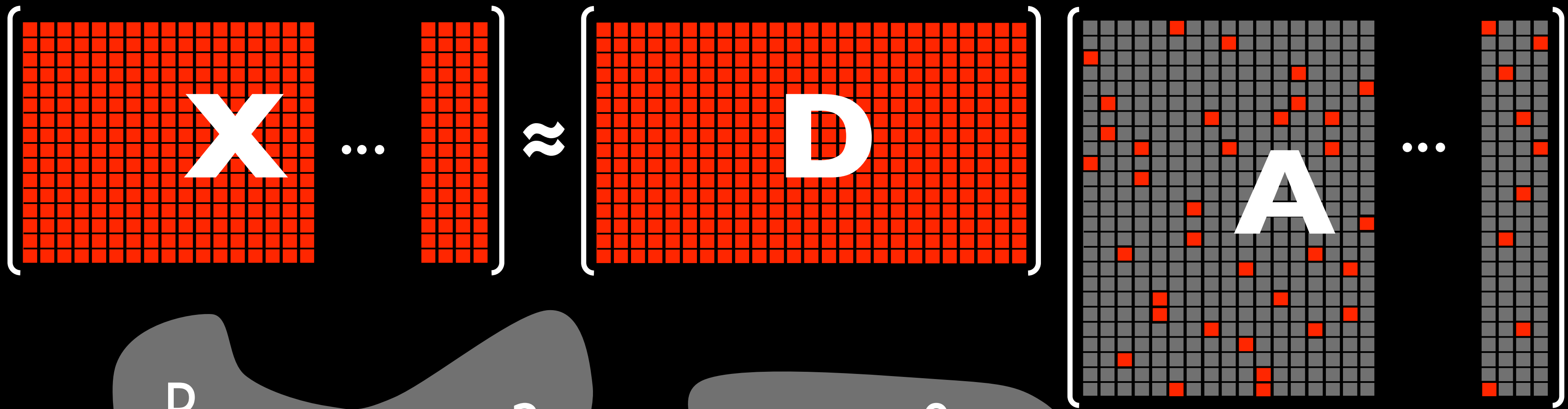
Our Assumption: Good-behaved Images have a sparse representation

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, Shearlets …)

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

Measure of Quality for D



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

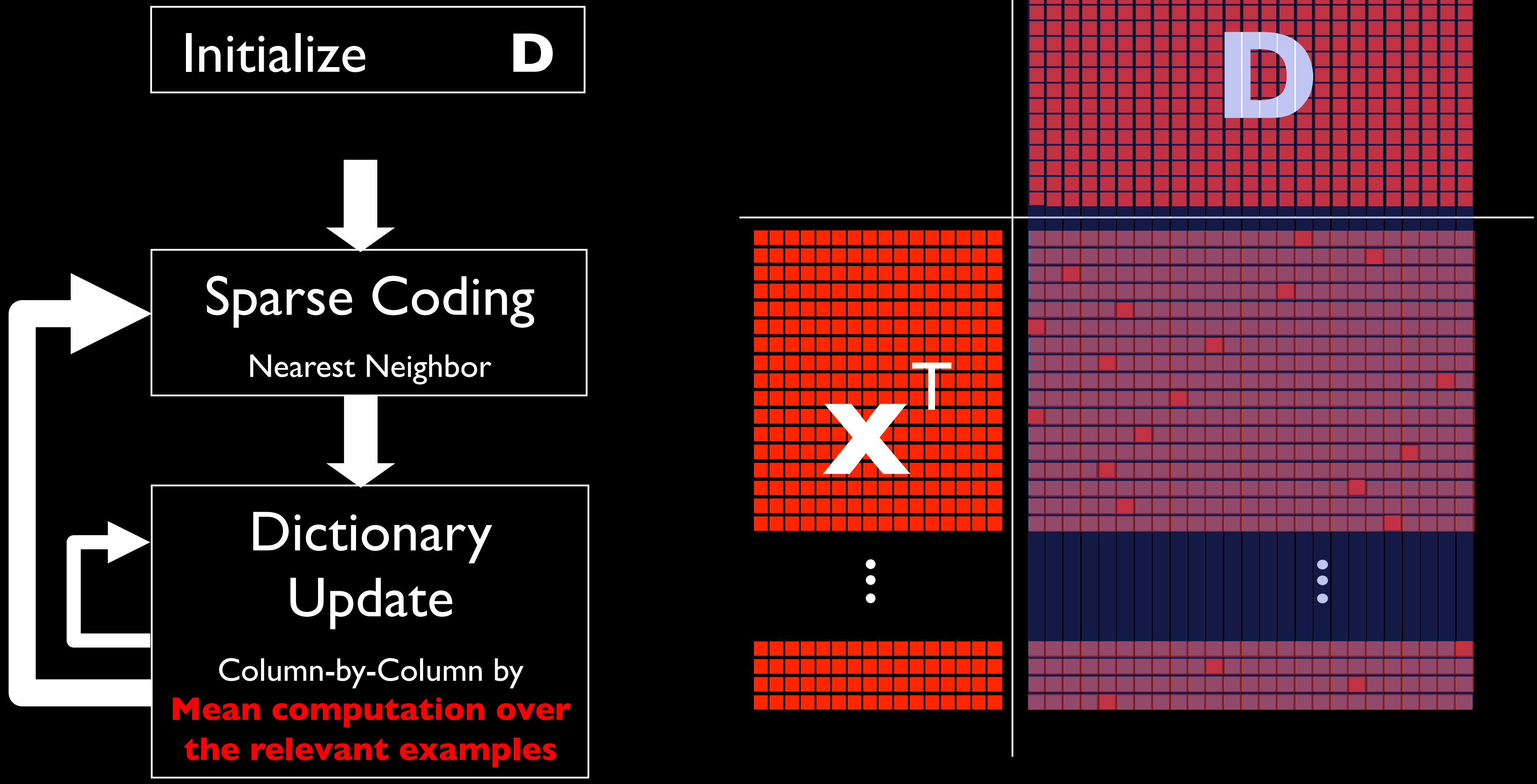
Each example is linear combination of atoms from **D**

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, E. & Bruckstein ('04)]
[Aharon, E. & Bruckstein ('05)]

K–Means For Clustering

Clustering: An extreme sparse representation



The K–SVD Algorithm – General

[Aharon, E. & Bruckstein ('04,'05)]

Initialize D

↓

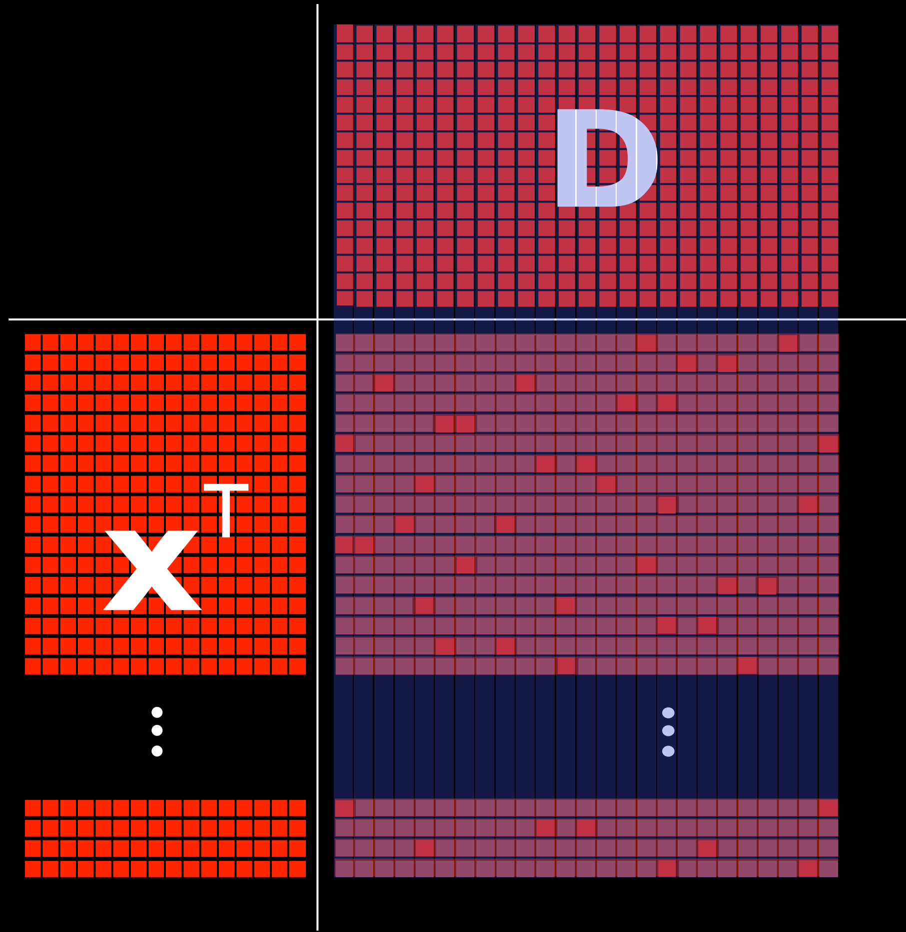
Sparse Coding

Use Matching Pursuit

↓

Dictionary Update

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



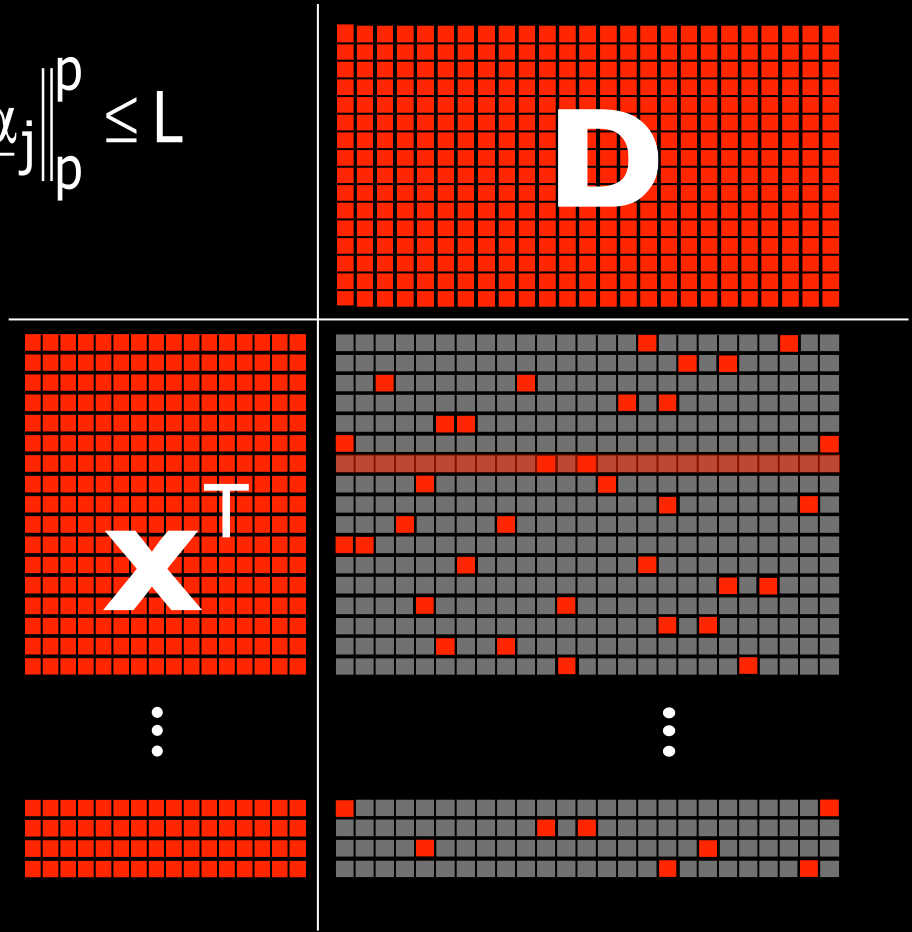
K–SVD: Sparse Coding Stage

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

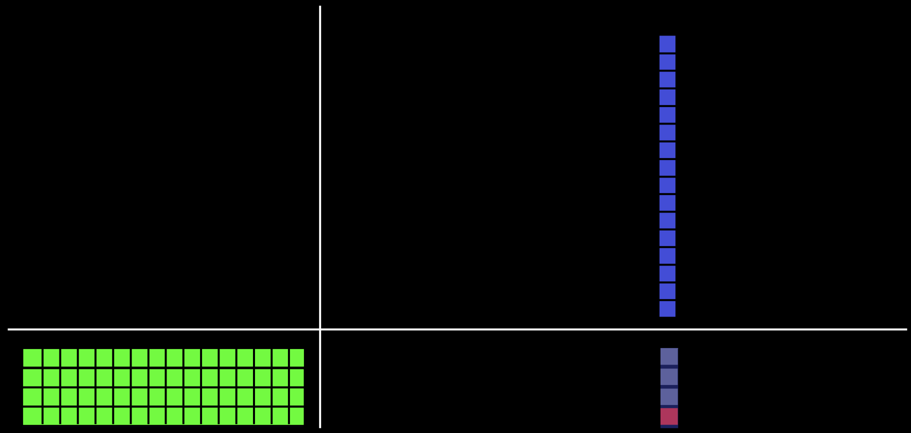
D is known!
For the j^{th} item
we solve

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

Solved by
A Pursuit Algorithm



K–SVD: Dictionary Update Stage

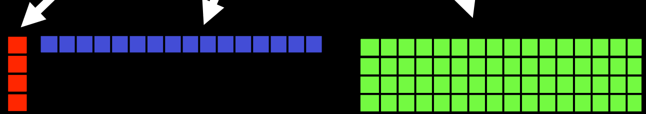


Refer only to the examples that use the column $\underline{d}_k$

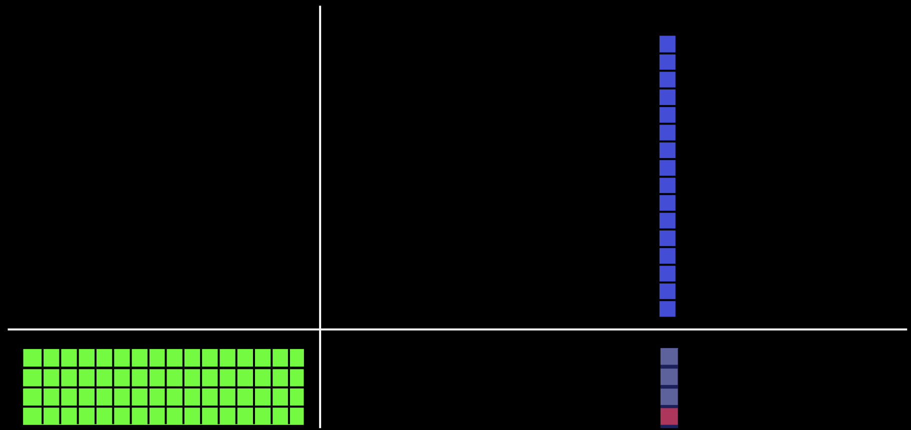
Fixing all **A** and **D** apart from the kth column, and seek both $\underline{d}_k$ and the kth column in **A** to better fit the **residual**!

We should solve:

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



K–SVD: Dictionary Update Stage

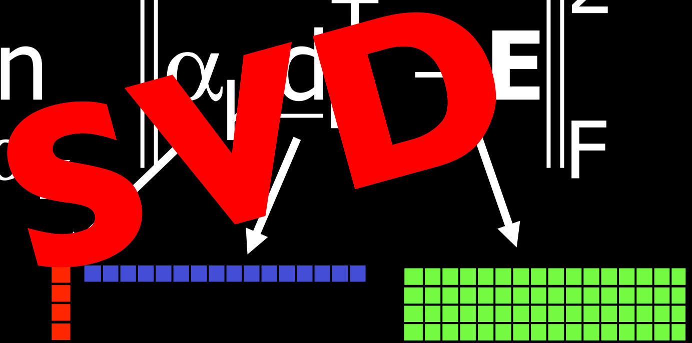


Refer only to the examples that use the column $\underline{d}_k$

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**!

We should solve:

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



To Summarize So Far …

Image denoising (and many other problems in image processing) requires a model for the desired image

What do we do?

Use a model for signals/images based on sparse and redundant representations

Problems?

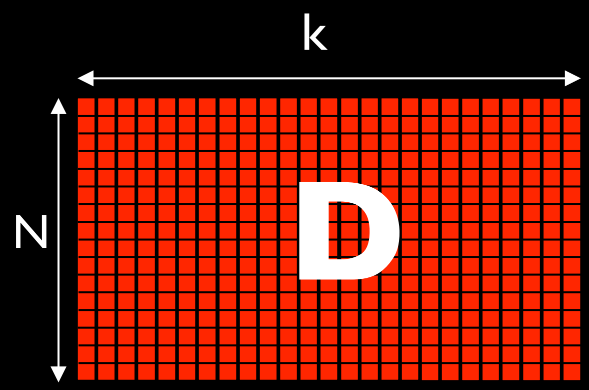
We have seen that there are approximation methods to find the sparsest solution, and there are theoretical results that guarantee their success.

What next?

Will it all work in applications?

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.

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

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 $\Rightarrow$ ~10^6 examples to use – more than enough.
- This works much better!

K-SVD Image Denoising

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

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

$\underline{x}$ and α_{ij} known

D and α_{ij} known

Compute α_{ij} per patch

$$\underline{\alpha}_{ij} = \underset{\underline{\alpha}}{\text{Min}}\left\|\mathbf{R}_{ij}\underline{x}-\mathbf{D}\underline{\alpha}\right\|_2^2 \quad \text{s.t.} \quad \left\|\underline{\alpha}\right\|_0^0 \le L$$

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

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

which is a simple averaging of shifted patches

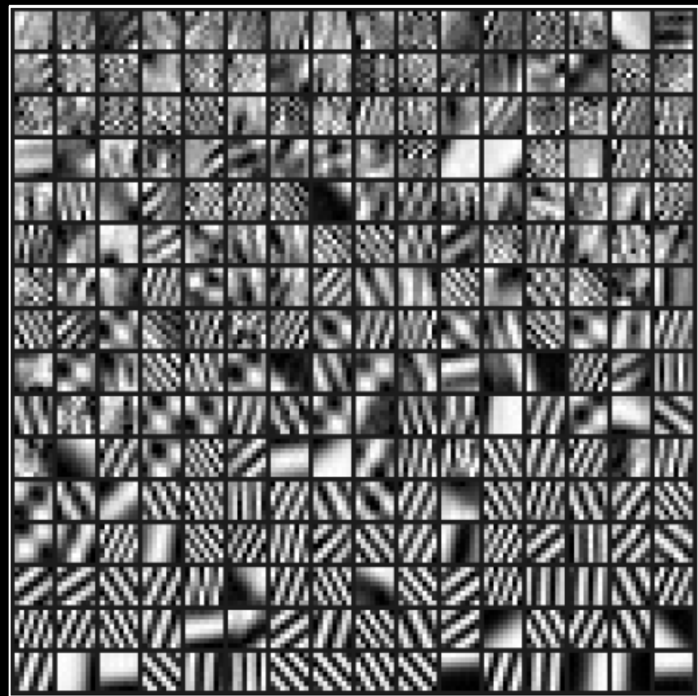
Image Denoising (Gray) [E. & Aharon ('06)]

Source

Result 30.829dB

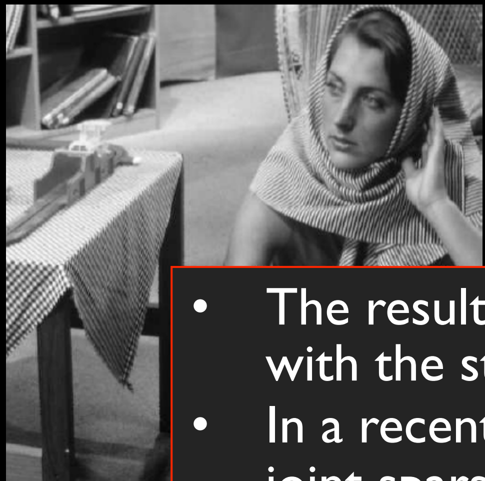
Noisy image

$\sigma = 20$

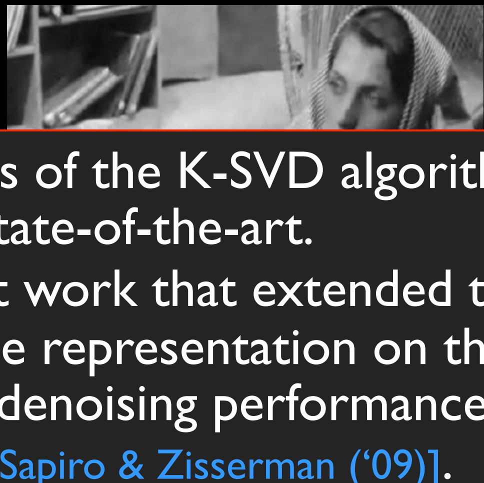


The obtained dictionary after 10 iterations

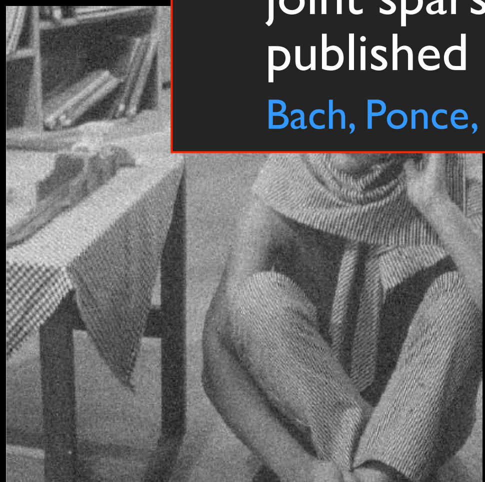
Image Denoising (Gray) [E. & Aharon ('06)]



Source

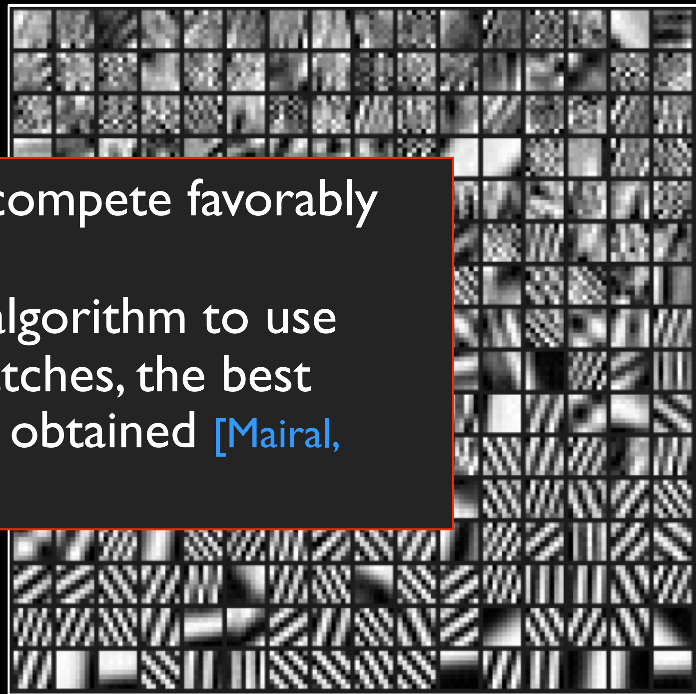


Result 30.829dB



Noisy image

$\sigma = 20$

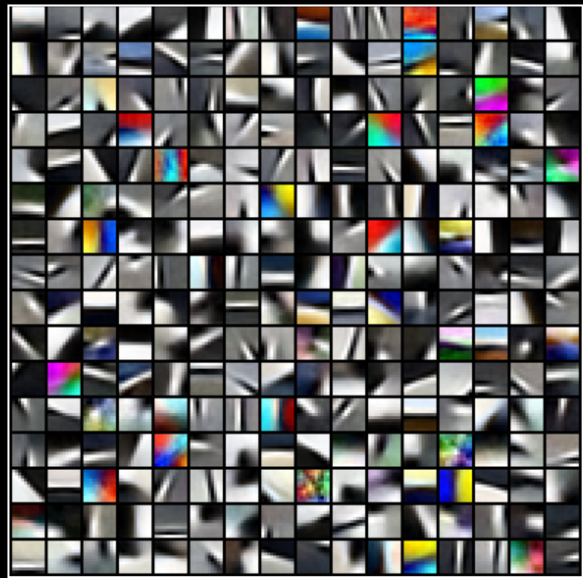


The obtained dictionary after 10 iterations

- The results of the K-SVD algorithm compete favorably with the state-of-the-art.
- In a recent work that extended this algorithm to use joint sparse representation on the patches, the best published denoising performance are obtained [Mairal, Bach, Ponce, Sapiro & Zisserman ('09)].

Denoising (Color) [Mairal, E. & Sapiro ('08)]

- When turning to handle color images, the main difficulty is in defining the relation between the color layers – R, G, and B.
- The solution with the above algorithm is simple – consider 3D patches or 8-by-8 with the 3 color layers, and the dictionary will detect the proper relations.



Denoising (Color) [Mairal, E. & Sapiro ('08)]

Original

Noisy (20.43dB)

Result (30.75dB)

Denoising (Color) [Mairal, E. & Sapiro ('08)]

The K-SVD algorithm leads to state-of-the-art denoising results, giving ~1dB better results compared to [Mcauley et. al. ('06)] which implements a learned MRF model (Field-of-Experts)

Original

Noisy (12.77dB)

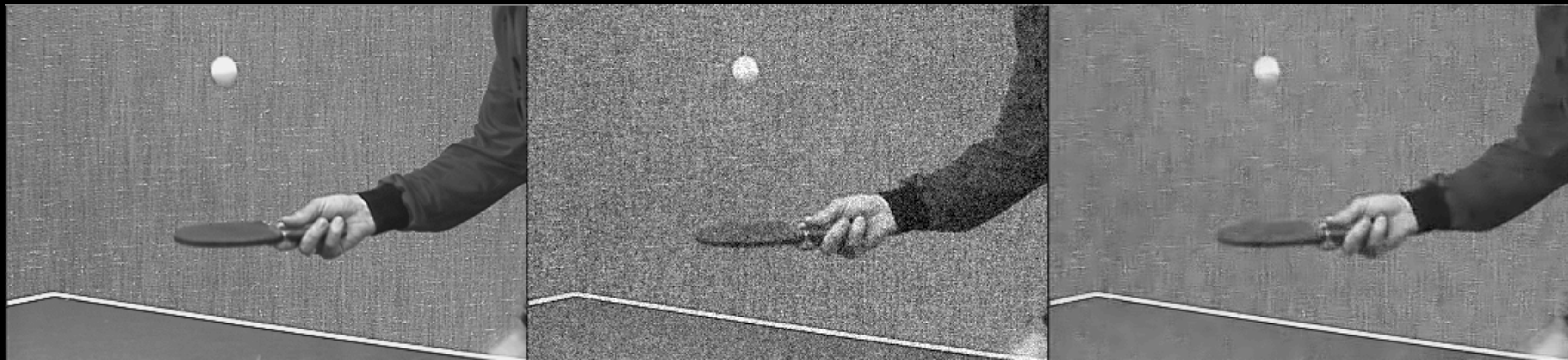
Result (29.87dB)

Video Denoising [Protter & E. ('09)]

When turning to handle video, one could improve over the previous scheme in three important ways:

1. Propagate the dictionary from one frame to another, and thus reduce the number of iterations;
2. Use 3D patches that handle the motion implicitly; and
3. Motion estimation and compensation can and should be avoided [Buades, Col, and Morel ('06)].

Video Denoising [Protter & E. ('09)]

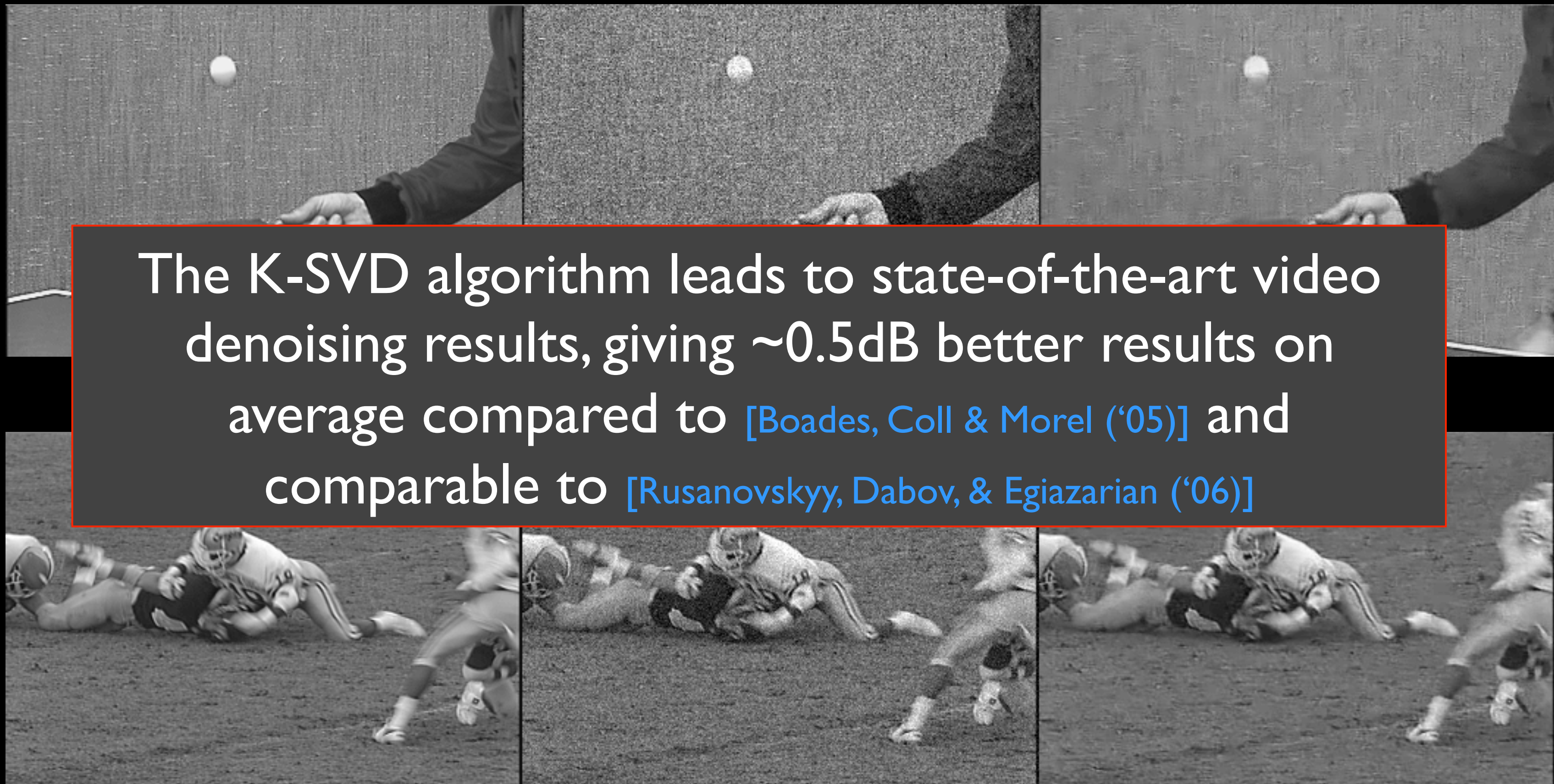


Original   Noisy (σ =25)   Denoised (PSNR=27.62)

Original   Noisy (σ =15)   Denoised (PSNR=29.98)

Video Denoising [Protter & E. ('09)]

The K-SVD algorithm leads to state-of-the-art video denoising results, giving ~0.5dB better results on average compared to [Boades, Coll & Morel ('05)] and comparable to [Rusanovskyy, Dabov, & Egiazarian ('06)]



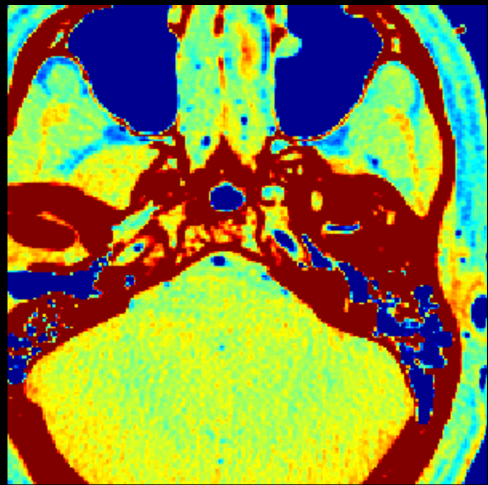
Original   Noisy (σ=15)   Denoised (PSNR=29.98)

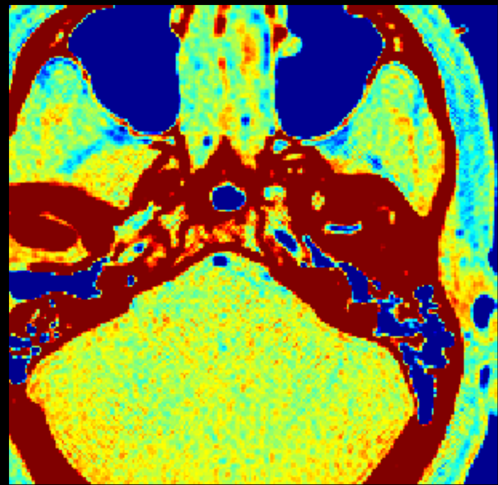
Low-Dosage Tomography [Shtok, Zibulevsky & E. ('10)]

- In Computer-Tomography (CT) reconstruction, an image is recovered from a set of its projections.
- In medicine, CT projections are obtained by X-ray, and it typically requires a high dosage of radiation in order to obtain a good quality reconstruction.
- A lower-dosage projection implies a stronger noise (Poisson distributed) in data to work with.
- Armed with sparse and redundant representation modeling, we can denoise the data and the final reconstruction … enabling CT with lower dosage.

Low-Dosage Tomography [Shtok, Zibulevsky & E. ('10)]

Original



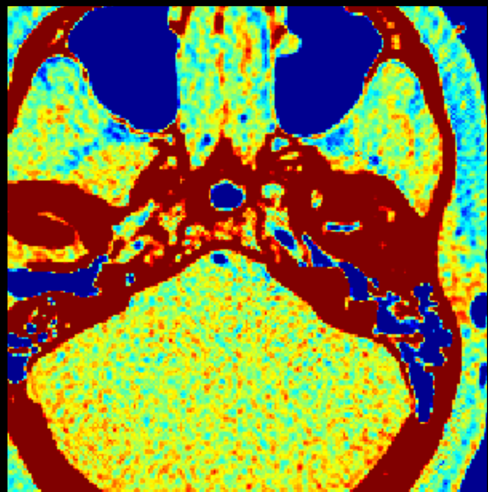


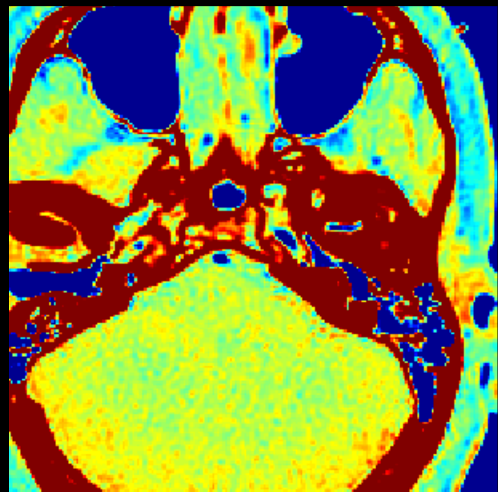
FBP result with high dosage

PSNR=24.63dB

FBP result with low dosage (one fifth)

PSNR=22.31dB





Denoising of the sinogram and post-processing (another denoising stage) of the reconstruction

PSNR=26.06dB

Image Inpainting – The Basics

- Assume: the signal $\underline{x}$ has been created by $\underline{x}=D\underline{\alpha}_0$ with very sparse $\underline{\alpha}_0$.
- Missing values in $\underline{x}$ imply missing rows in this linear system.
- By removing these rows, we get

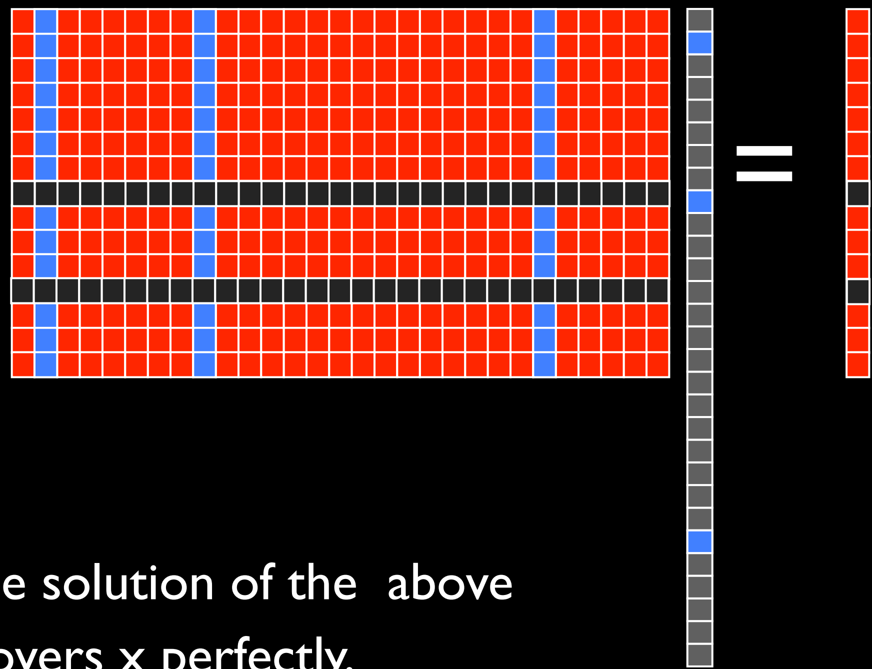
$$\tilde{\mathbf{D}}\underline{\alpha} = \underline{\tilde{x}}$$

- Now solve

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

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

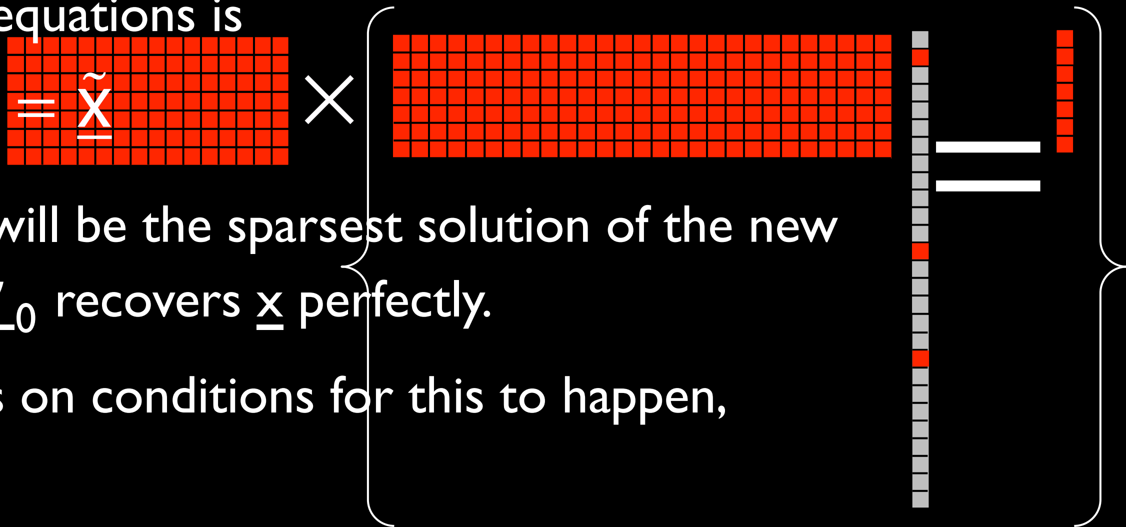
$$\mathbf{D}\,\underline{\alpha}_0 = \underline{x}$$



Side Note: Compressed-Sensing

- Compressed Sensing is leaning on the very same principal, leading to alternative sampling theorems.
- Assume: the signal $\underline{x}$ has been created by $\underline{x}=D\underline{\alpha}_0$ with very sparse $\underline{\alpha}_0$.
- Multiply this set of equations by the matrix Q which reduces the number of rows.
- The new, smaller, system of equations is

$$\mathbf{QD}\underline{\alpha} = \mathbf{Q}\underline{x} \;\; \Rightarrow \;\; \tilde{\mathbf{D}}\underline{\alpha} = \tilde{\underline{x}}$$



- If $\underline{\alpha}_0$ was sparse enough, it will be the sparsest solution of the new system, thus, computing $D\underline{\alpha}_0$ recovers $\underline{x}$ perfectly.
- Compressed sensing focuses on conditions for this to happen, guaranteeing such recovery.

Inpainting [Mairal, E. & Sapiro (‘08)]

Experiments lead to state-of-the-art inpainting results.

Original

80% missing

Result

Inpainting [Mairal, E. & Sapiro ('08)]

Experiments lead to state-of-the-art inpainting results.

Original

80% missing

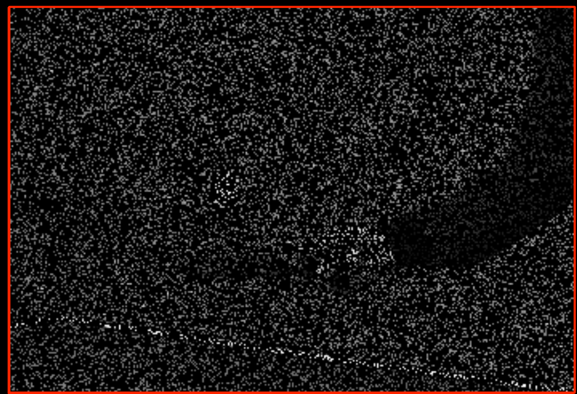
Result

Inpainting [Mairal, E. & Sapiro ('08)]

Experiments lead to state-of-the-art inpainting results.

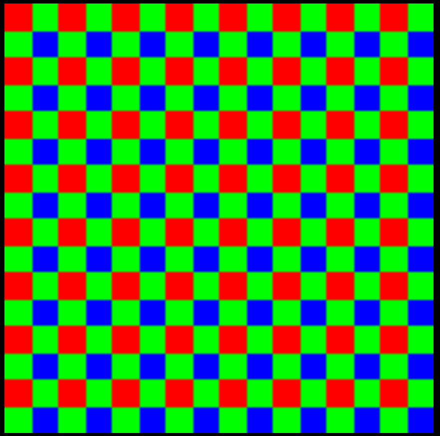
Inpainting [Mairal, E. & Sapiro ('08)]

The same can be done for video, very much like the denoising treatment: (i) 3D patches, (ii) no need to compute the dictionary from scratch for each frame, and (iii) no need for explicit motion estimation



Original 80% missing Result

Demosaicing [Mairal, E. & Sapiro ('08)]



- Today's cameras are sensing only one color per pixel, leaving the rest for interpolated.
- Generalizing the inpainting scheme to handle demosaicing is tricky because of the possibility to learn the mosaic pattern within the dictionary.
- In order to avoid "over-fitting", we handle the demosaicing problem while forcing strong sparsity and applying only few iterations.

Demosaicing [Mairal, E. & Sapiro ('08)]

Experiments lead to state-of-the-art demosaicing results, giving ~0.2dB better results on average, compared to [Chang & Chan ('06)]








Image Compression [Bryt and E. ('08)]

- 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 well.
- Geometric alignment of the image is very helpful and should be done [Goldenberg, Kimmel, & E. ('05)].

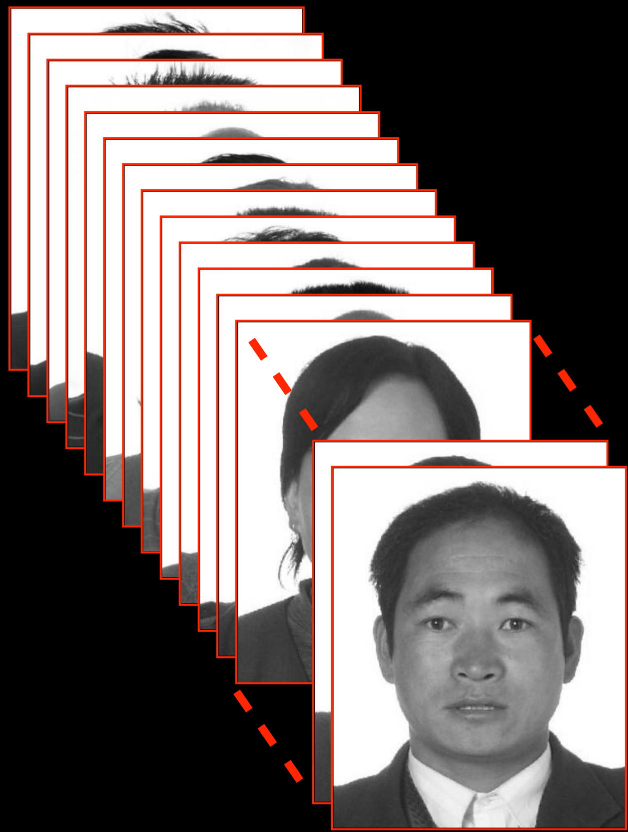


Image Compression

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, apply Q, and dewarp

On the training set

On the test image

Training set (2500 images)

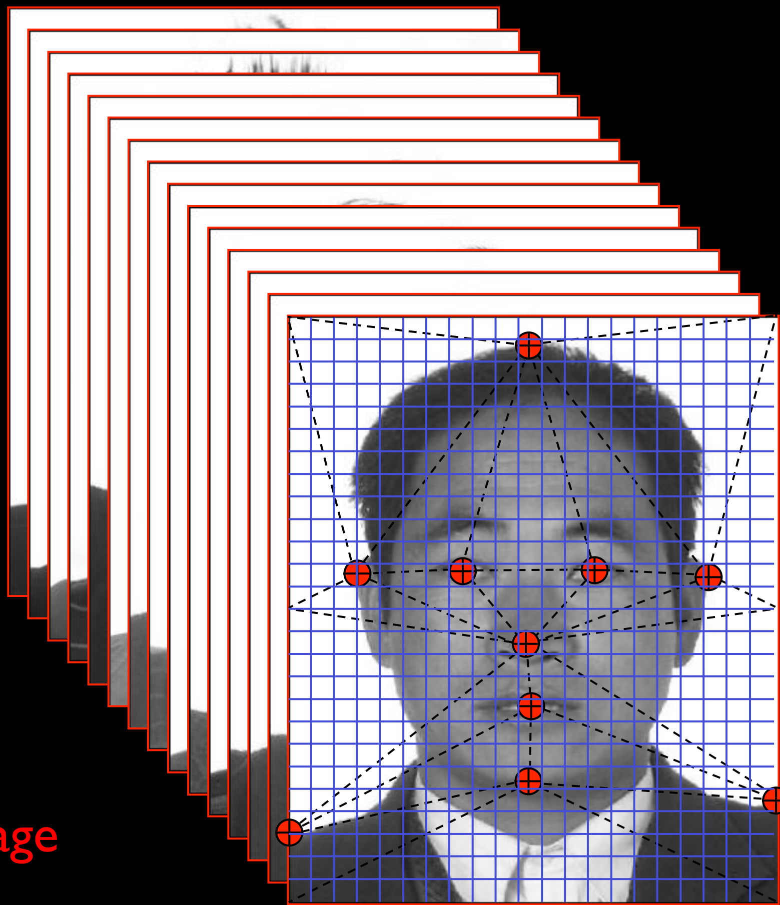


Image Compression Results

Original
JPEG
JPEG-2000
Local-PCA
K-SVD

Results for **820** Bytes per each file

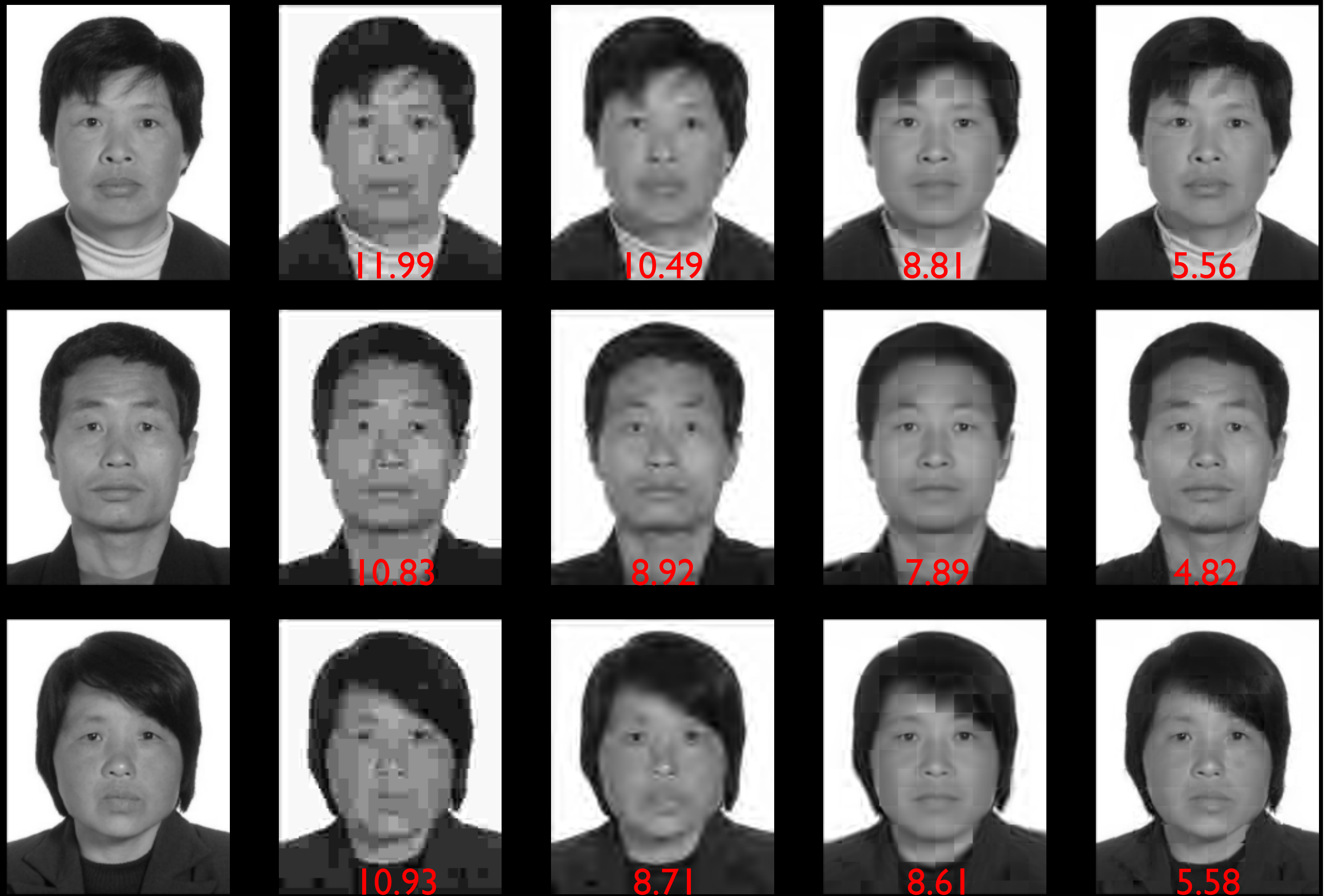


Image Compression Results

Original

JPEG

JPEG-2000

Local-PCA

K-SVD

Results for **550** Bytes per each file

Image Compression Results

Original

JPEG

JPEG-2000

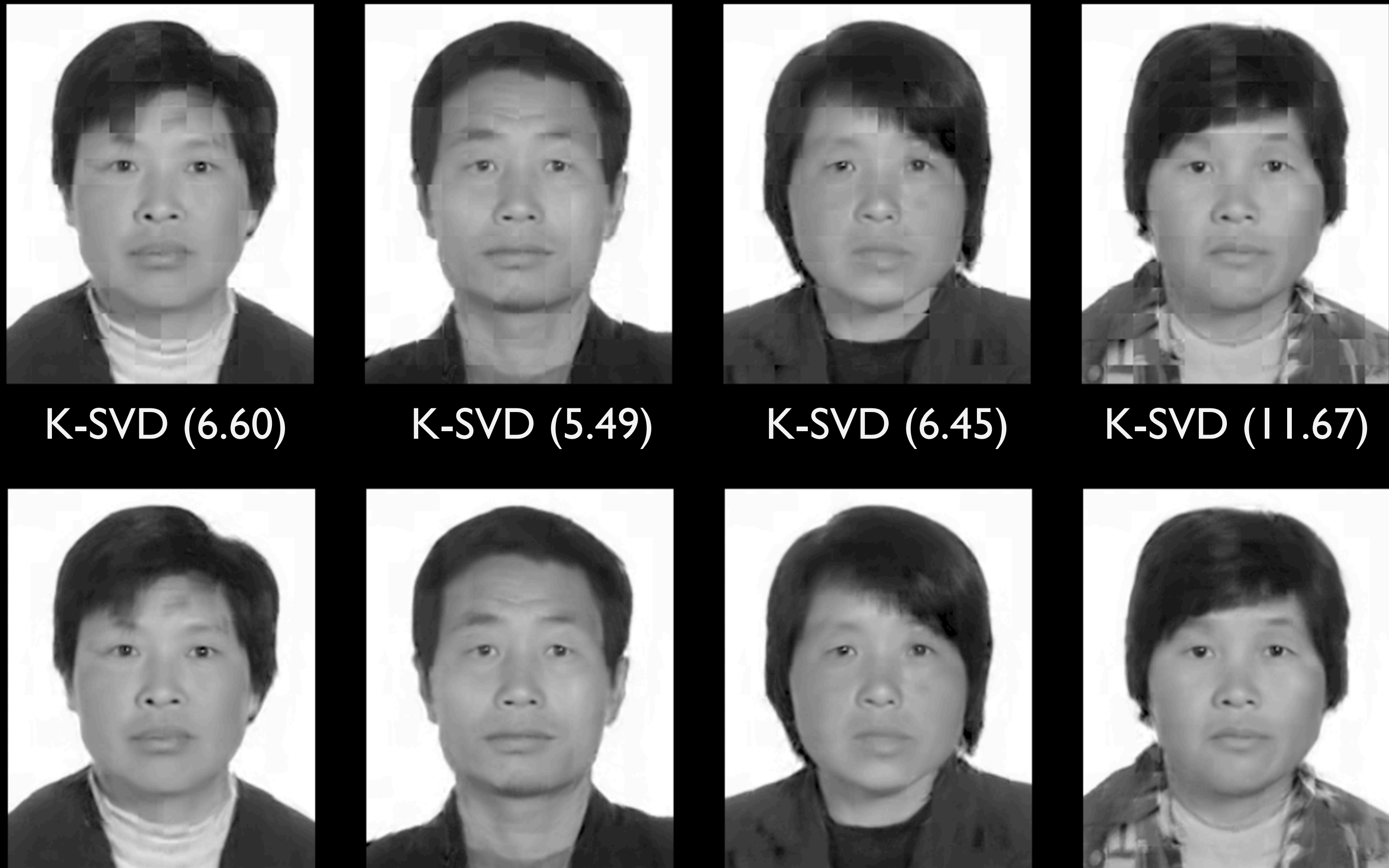
Local-PCA

K-SVD

Results for **400** Bytes per each file

Deblocking the Results [Bryt and E. (`09)]

550 bytes K-SVD results with and without deblocking



K-SVD (6.60) K-SVD (5.49) K-SVD (6.45) K-SVD (11.67)

Deblock (6.24) Deblock (5.27) Deblock (6.03) Deblock (11.32)

Super-Resolution [Zeyde, Protter, & E. ('11)]

- Given a low-resolution image, we desire to enlarge it while producing a sharp looking result. This problem is referred to as "Single-Image Super-Resolution".
- Image scale-up using bicubic interpolation is far from being satisfactory for this task.
- Recently, a sparse and redundant representation technique was proposed [Yang, Wright, Huang, and Ma ('08)] for solving this problem, by training a coupled-dictionaries for the low- and high res. images.
- We extended and improved their algorithms and results.

Super-Resolution – Results (1)

This book is about *convex optimization*, a special class of mathematical optimization problems, which includes least-squares and linear programming problems. It is well known that least-squares and linear programming problems have a fairly complete theory, arise in a variety of applications, and can be solved numerically very efficiently. The basic point of this book is that the same can be said for the larger class of convex optimization problems.

While the mathematics of convex optimization has been studied for about a century, several related recent developments have stimulated new interest in the topic. The first is the recognition that interior-point methods, developed in the 1980s to solve linear programming problems, can be used to solve convex optimization problems as well. These new methods allow us to solve certain new classes of convex optimization problems, such as semidefinite programs and second-order cone programs, almost as easily as linear programs.

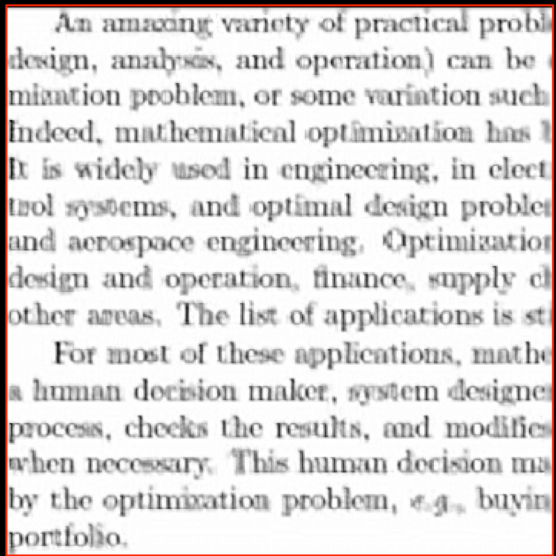
The second development is the discovery that convex optimization problems (beyond least-squares and linear programs) are more prevalent in practice than was previously thought. Since 1990 many applications have been discovered in areas such as automatic control systems, estimation and signal processing, communications and networks, electronic circuit design, data analysis and modeling, statistics, and finance. Convex optimization has also found wide application in combinatorial optimization and global optimization, where it is used to find bounds on the optimal value, as well as approximate solutions. We believe that many other applications of convex optimization are still waiting to be discovered.

There are great advantages to recognizing or formulating a problem as a convex optimization problem. The most basic advantage is that the problem can then be solved, very reliably and efficiently, using interior-point methods or other special methods for convex optimization. These solution methods are reliable enough to be embedded in a computer-aided design or analysis tool, or even a real-time reactive or automatic control system. There are also theoretical or conceptual advantages of formulating a problem as a convex optimization problem. The associated dual

PSNR=14.68dB

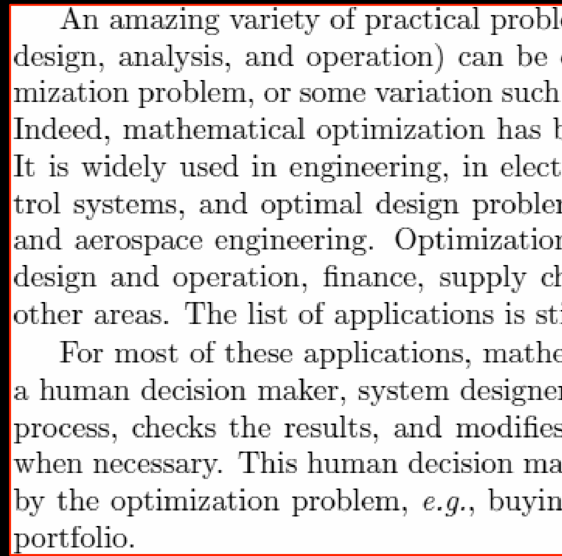
The training image: 717×717 pixels, providing a set of 54,289 training patch-pairs.

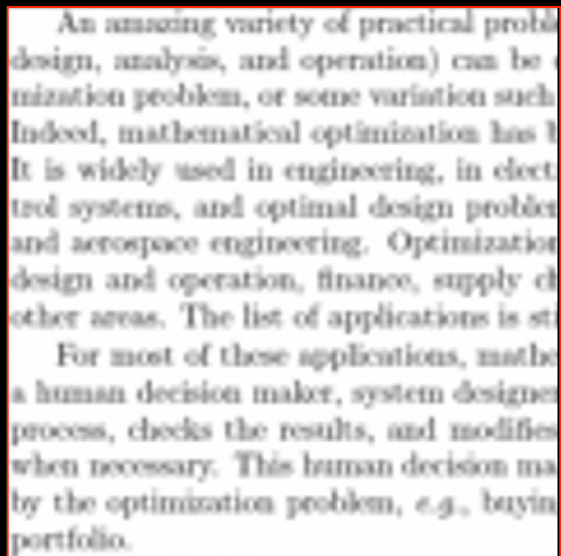
Super-Resolution – Results (1)


An amazing variety of practical probl
design, analysis, and operation) can be
mization problem, or some variation such
Indeed, mathematical optimization has
It is widely used in engineering, in elect
trol systems, and optimal design proble
and aerospace engineering. Optimizatio
design and operation, finance, supply c
other areas. The list of applications is st
For most of these applications, mathe
a human decision maker, system designe
process, checks the results, and modifies
when necessary. This human decision ma
by the optimization problem, *e.g.*, buyin
portfolio.

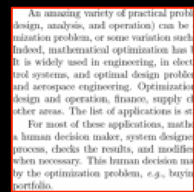
SR Result
PSNR=16.95dB

Ideal Image


An amazing variety of practical probl
design, analysis, and operation) can be
mization problem, or some variation such
Indeed, mathematical optimization has b
It is widely used in engineering, in elect
trol systems, and optimal design problem
and aerospace engineering. Optimization
design and operation, finance, supply ch
other areas. The list of applications is sti
For most of these applications, mathe
a human decision maker, system designer
process, checks the results, and modifies
when necessary. This human decision ma
by the optimization problem, *e.g.*, buyin
portfolio.


An amazing variety of practical probl
design, analysis, and operation) can be
mization problem, or some variation such
Indeed, mathematical optimization has
It is widely used in engineering, in elect
trol systems, and optimal design proble
and aerospace engineering. Optimizatio
design and operation, finance, supply c
other areas. The list of applications is st
For most of these applications, mathe
a human decision maker, system designe
process, checks the results, and modifies
when necessary. This human decision ma
by the optimization problem, *e.g.*, buyin
portfolio.

Bicubic interpolation
PSNR=14.68dB



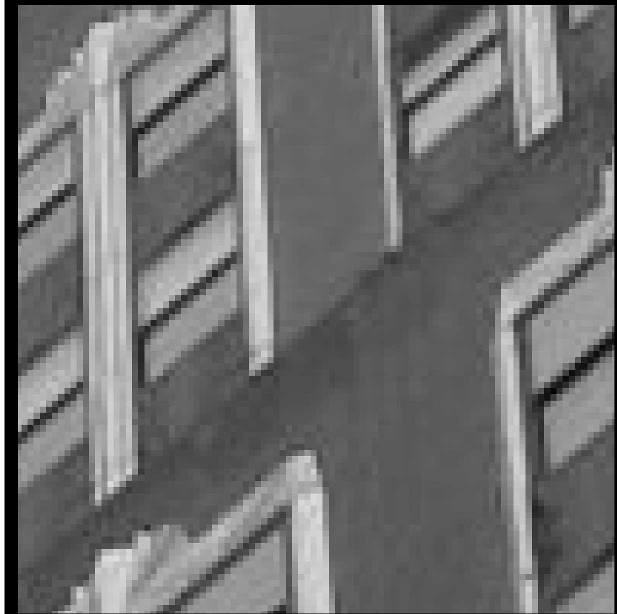
Given Image

Super-Resolution – Results (2)

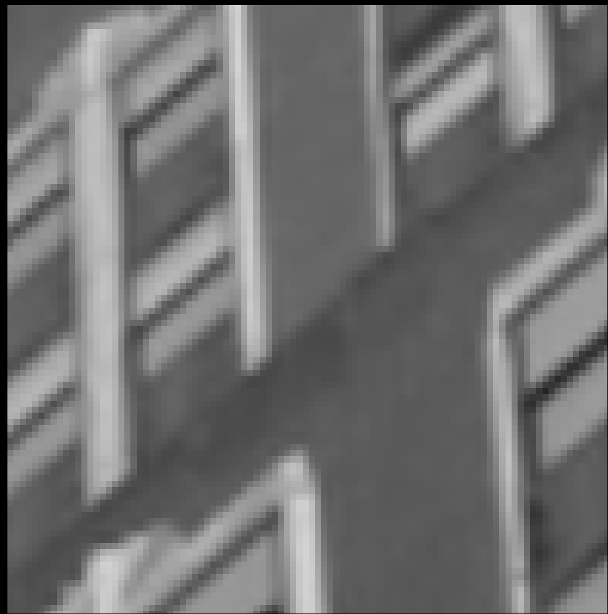
Given image

Scaled-Up (factor 2:1) using the proposed algorithm,
PSNR=29.32dB (3.32dB improvement over bicubic)

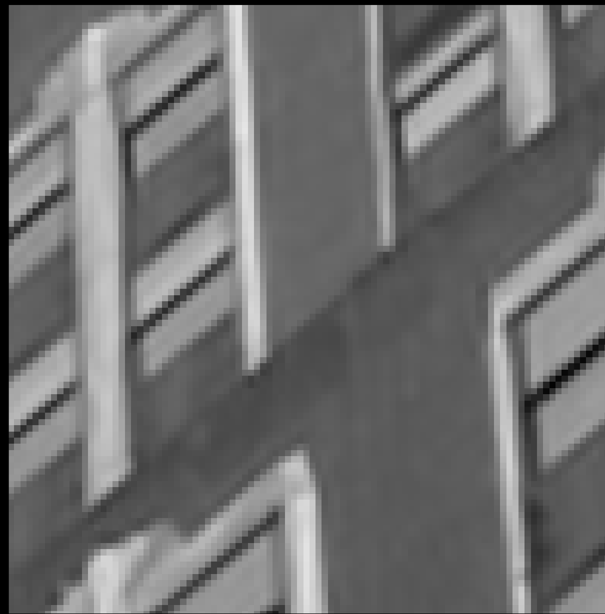
Super-Resolution – Results (2)



The Original



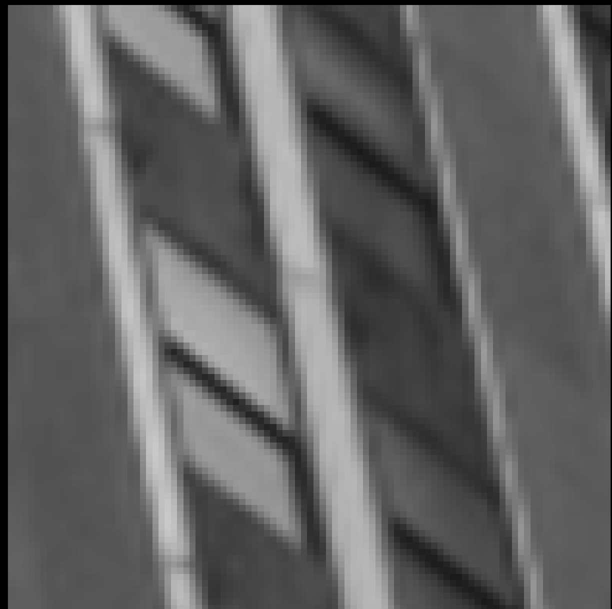
Bicubic Interpolation



SR result

Super-Resolution – Results (2)

The Original



Bicubic Interpolation

SR result