

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

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

\underline{y} : Given measurements
 \underline{x} : Unknown to be recovered

Relation to measurements

Prior or regularization

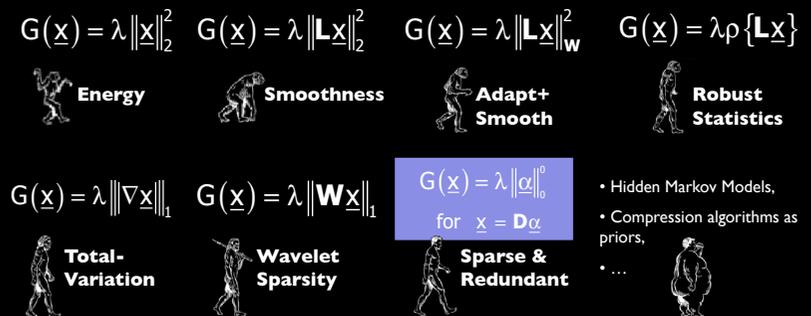
- This is in-fact a Bayesian point of view, adopting the Maximum-A-posteriori Probability (MAP) estimation.
- Clearly, the wisdom in such an approach is within the choice of the prior – **modeling the images** of interest.



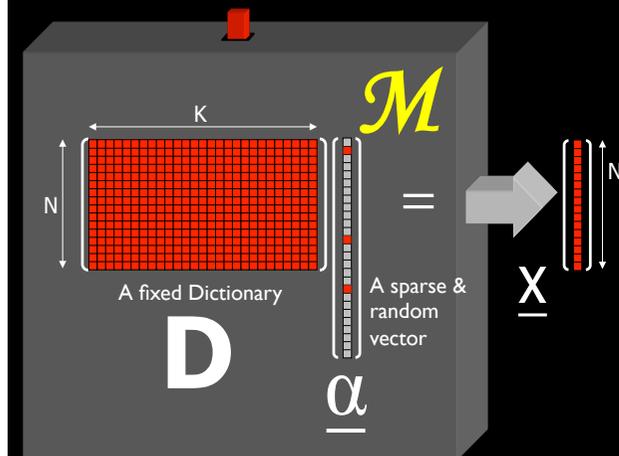
Thomas Bayes
1702 - 1761

The Evolution of $G(\underline{x})$

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

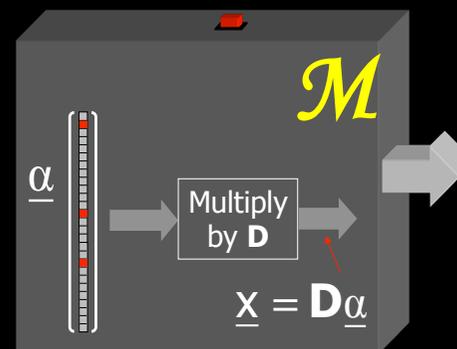


Sparse Modeling of Signals



- Every column in \mathbf{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



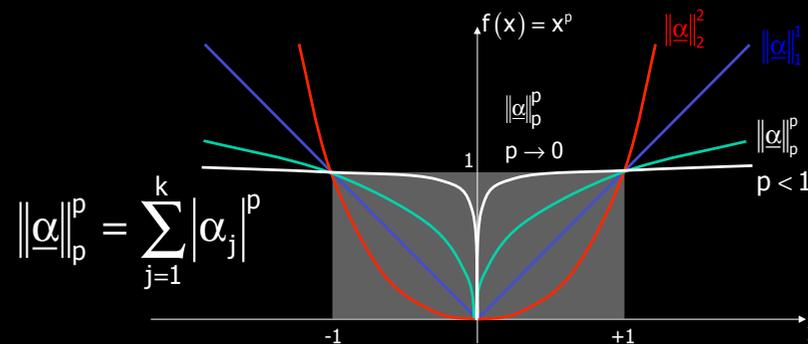
Interesting Model:

- **Simple:** Every generated signal is built as a linear combination of **few atoms** from our **dictionary \mathbf{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?

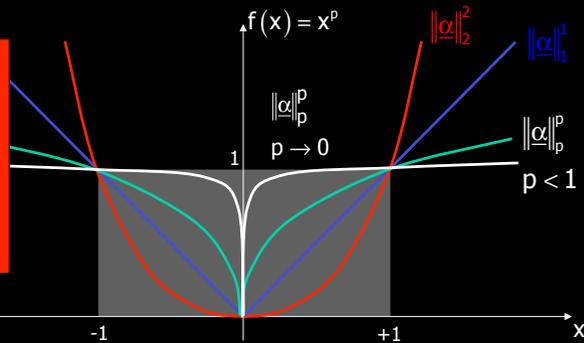


Our signal model is thus: $\underline{x} = \mathbf{D}\underline{\alpha}$ where $\underline{\alpha}$ is sparse

Sparse & Redundant Rep. Modeling?

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

→ $\|\underline{\alpha}\|_0$



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

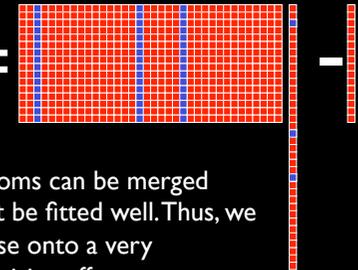
Back to Our MAP Energy Function

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

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

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

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



- 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

$$\min_{\underline{\alpha}} \|\mathbf{D}\underline{\alpha} - \underline{y}\|_2^2 \text{ s.t. } \|\underline{\alpha}\|_0 \leq L \quad \text{or} \quad \min_{\underline{\alpha}} \|\underline{\alpha}\|_0 \text{ s.t. } \|\mathbf{D}\underline{\alpha} - \underline{y}\|_2^2 \leq \epsilon^2$$

or $\min_{\underline{\alpha}} \lambda \|\underline{\alpha}\|_0 + \|\mathbf{D}\underline{\alpha} - \underline{y}\|_2^2$

- 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 \mathbf{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

There are some issues:

- Theoretical
- How to approximate?
- What about \mathbf{D} ?



Great! No?

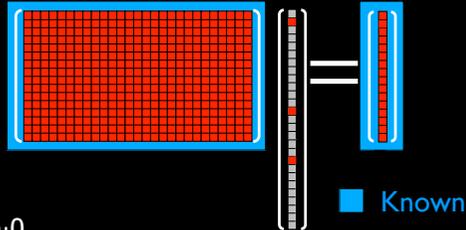
Lets Start with the Noiseless Problem

Suppose we build a signal by the relation

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

We aim to find the signal's representation:

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



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 \mathbf{D} , $\sigma = \text{Spark}\{\mathbf{D}\}$ 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

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

Uniqueness

Donoho & E. ('02)

If we found a representation that satisfy

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

Then necessarily it is unique (the sparsest).

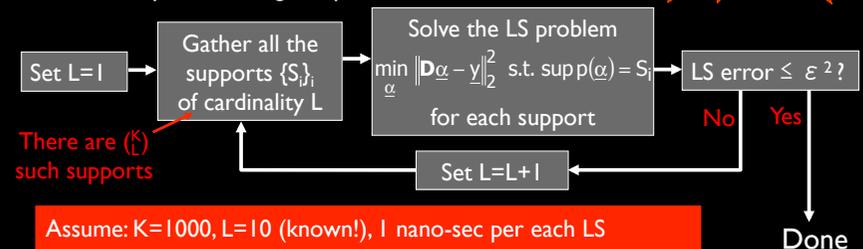
This result implies that if \mathcal{M} generates signals using "sparse enough" $\underline{\alpha}$, the solution of the above will find it exactly.

Our Goal

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

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 $\sim 8e+6$ years to solve this problem !!!!!

Lets Approximate

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



Relaxation methods

Smooth the L_0 and use continuous optimization techniques

Greedy methods

Build the solution one non-zero element at a time

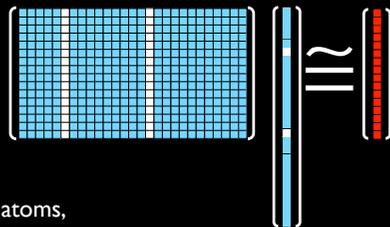
Relaxation – The Basis Pursuit (BP)

Instead of solving $\min_{\underline{\alpha}} \|\underline{\alpha}\|_0 \quad \text{s.t.} \quad \|\mathbf{D}\underline{\alpha} - \underline{y}\|_2 \leq \epsilon$  Solve Instead $\min_{\underline{\alpha}} \|\underline{\alpha}\|_1 \quad \text{s.t.} \quad \|\mathbf{D}\underline{\alpha} - \underline{y}\|_2 \leq \epsilon$

- 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 $\|\mathbf{D}\underline{\alpha} - \underline{y}\|_2$ 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

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

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

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

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

- Greedy Algorithms [e.g., Least-Squares, Orthogonal Matching Pursuit (OMP), etc.]
- Relaxation methods [e.g., Iterative Shrinkage and Thresholding (ISTA), etc.]
- Hybrid Algorithms [e.g., Iterative Shrinkage and Thresholding with Soft-Thresholding (ISTAT), etc.]
- ...

Why should they work?

(MP),
1993-

The Mutual Coherence

• Compute $\mathbf{D}^T \mathbf{D}$ (Assume normalized columns)

- 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)

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

BP and MP Equivalence (No Noise)

Equivalence

Given a signal \underline{x} with a representation $\underline{x} = \mathbf{D}\underline{\alpha}$, assuming that $\|\underline{\alpha}\|_0 < 0.5(1 + 1/\mu)$, 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

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

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 $\underline{v} \sim \mathcal{N}(0, \sigma^2 \mathbf{I})$, BP will show* stability, i.e.,

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

Ben-Haim, Eldar & E. ('09)

*With very high probability

- For $\sigma=0$ we get a weaker version of the previous result.
- This result is the oracle's error, multiplied 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?

The Dictionary \mathbf{D} should be found somehow !!!

What next?

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

What Should \mathbf{D} Be?

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

Our Assumption: Good-behaved Images have a sparse representation

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

One approach to choose \mathbf{D} is from a known set of transforms (Steerable wavelet, Curvelet, Contourlets, Bandlets, Shearlets ...)

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

Measure of Quality for D

$$\begin{bmatrix} \mathbf{x} & \dots & \mathbf{x} \end{bmatrix} \approx \begin{bmatrix} \mathbf{D} \end{bmatrix} \begin{bmatrix} \mathbf{A} & \dots & \mathbf{A} \end{bmatrix}$$

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

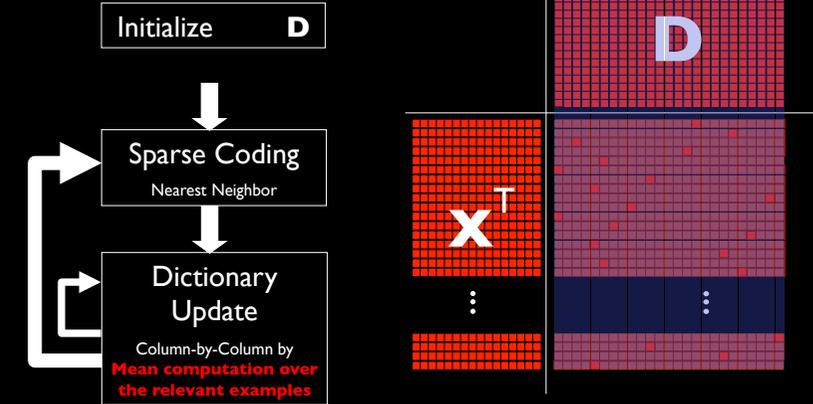
Each example is a linear combination of atoms from \mathbf{D}

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)]

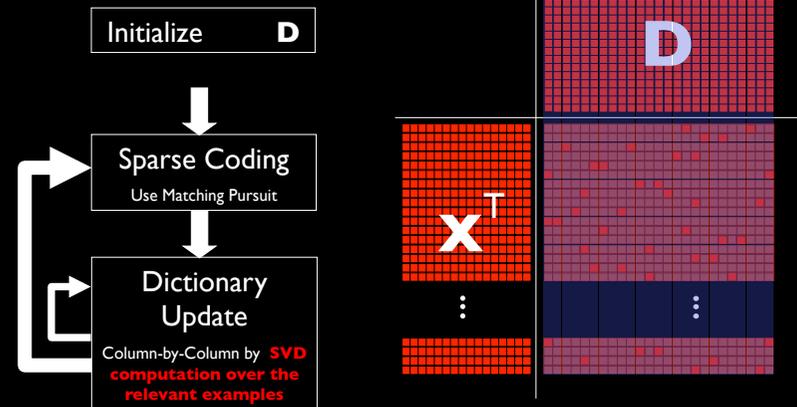
K-Means For Clustering

Clustering: An extreme sparse representation



The K-SVD Algorithm – General

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



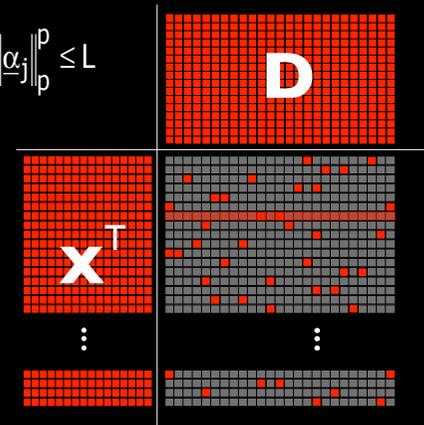
K-SVD: Sparse Coding Stage

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

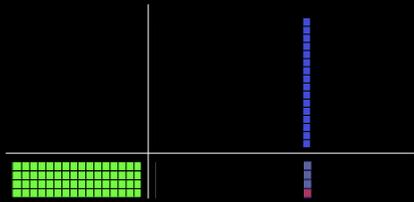
\mathbf{D} is known!
 For the j^{th} item
 we solve

$$\text{Min}_{\underline{\alpha}} \|\mathbf{D}\underline{\alpha} - \mathbf{x}_j\|_2^2 \quad \text{s.t. } \|\underline{\alpha}\|_p \leq L$$

Solved by
A Pursuit Algorithm



K-SVD: Dictionary Update Stage



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

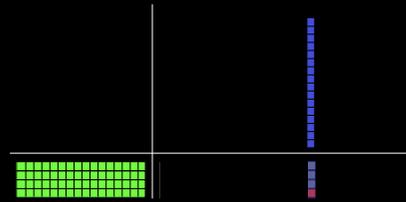


We should solve:

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

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

K-SVD: Dictionary Update Stage



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



We should solve:

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

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

To Summarize So Far ...

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



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



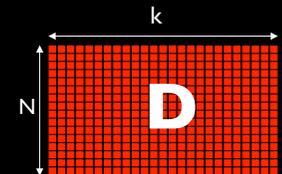
Will it all work in applications?



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

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.



$$\hat{\underline{x}} = \text{ArgMin}_{\underline{x}, \{\alpha_{ij}\}_{ij}} \frac{1}{2} \|\underline{x} - \underline{y}\|_2^2 + \mu \sum_{ij} \|\mathbf{R}_{ij} \underline{x} - \mathbf{D} \alpha_{ij}\|_2^2$$

s.t. $\|\alpha_{ij}\|_0 \leq L$

Extracts a patch in the ij location

Our prior

What Data to Train On?

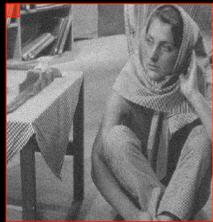
Option 1:

- 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 1000² pixels → ~10⁶ examples to use – more than enough.
- This works much better!



K-SVD Image Denoising

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

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

\mathbf{x} and α_{ij} known

\mathbf{D} and α_{ij} known

Compute α_{ij} per patch

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

s.t. $\|\alpha\|_0 \leq L$
using the matching pursuit

Compute \mathbf{D} to minimize

$$\underset{\mathbf{D}}{\text{Min}} \sum_{ij} \|\mathbf{R}_{ij}\mathbf{x} - \mathbf{D}\alpha_{ij}\|_2^2$$

using SVD, updating one column at a time

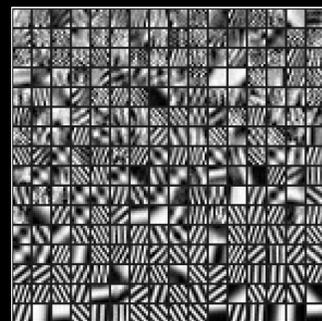
Compute \mathbf{x} by

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

which is a simple averaging of shifted patches

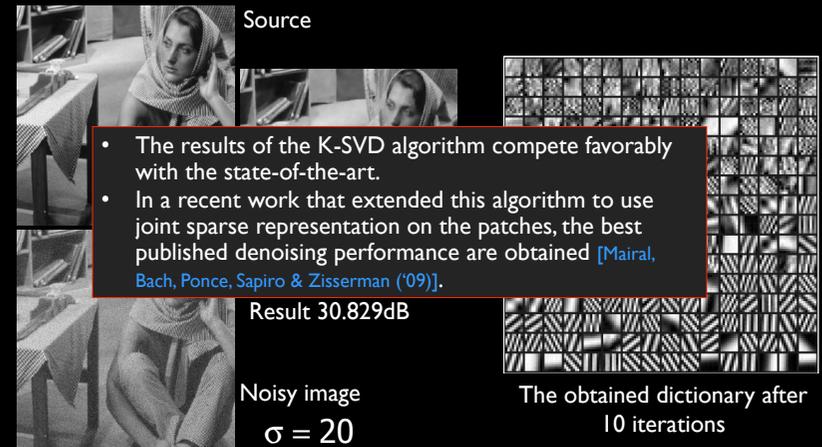
K-SVD

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

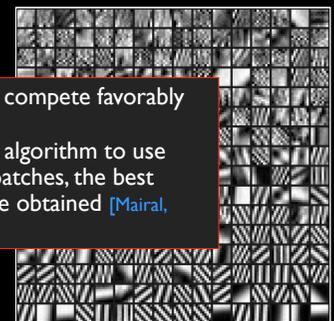


The obtained dictionary after 10 iterations

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



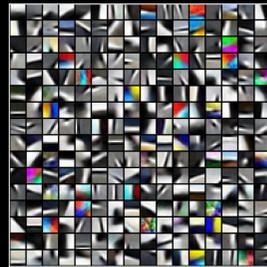
- The results of the K-SVD algorithm compete favorably with the state-of-the-art.
- In a recent work that extended this algorithm to use joint sparse representation on the patches, the best published denoising performance are obtained [Mairal, Bach, Ponce, Sapiro & Zisserman ('09)].



The obtained dictionary after 10 iterations

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)

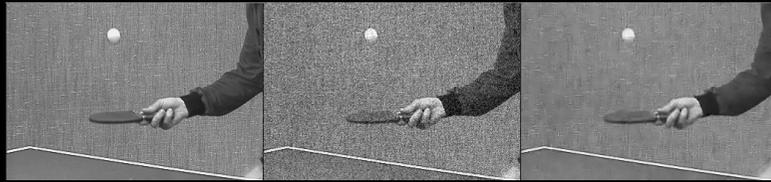
Video Denoising [Protter & E. ('09)]

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 ($\sigma=25$)

Denoised (PSNR=27.62)

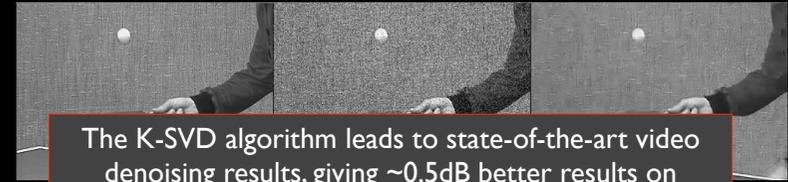


Original

Noisy ($\sigma=15$)

Denoised (PSNR=29.98)

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



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



Original

Noisy ($\sigma=15$)

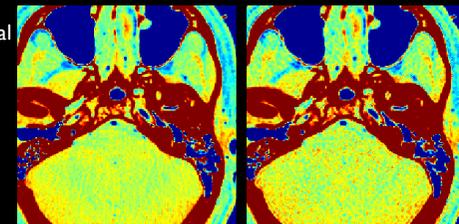
Denoised (PSNR=29.98)

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

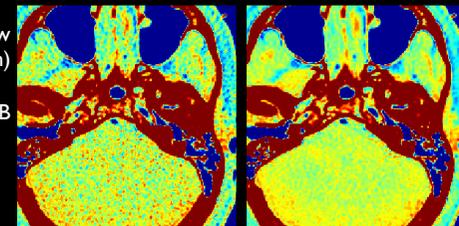


FBP result with high dosage

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

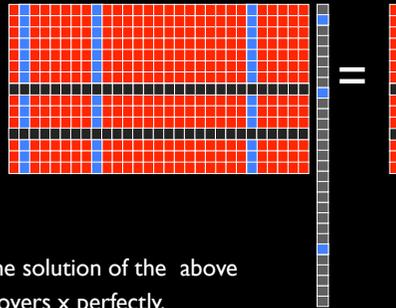
$$\tilde{D}\underline{\alpha} = \tilde{\underline{X}}$$

- Now solve

$$\text{Min}_{\underline{\alpha}} \|\underline{\alpha}\|_0 \quad \text{s.t.} \quad \tilde{\underline{X}} = \tilde{D}\underline{\alpha}$$

- 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.

$$D \underline{\alpha}_0 = \underline{X}$$



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

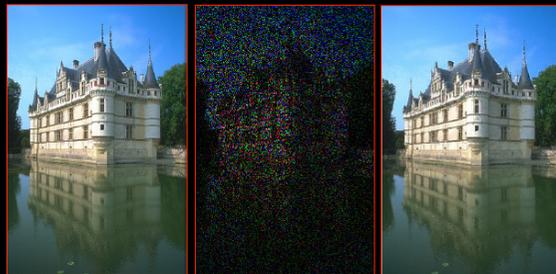
$$QD\underline{\alpha} = Q\underline{x} \rightarrow \tilde{D}\underline{\alpha} = \tilde{\underline{X}}$$

- 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.



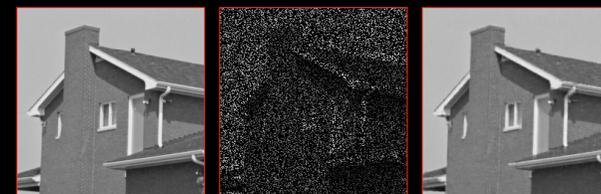
Original

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)]

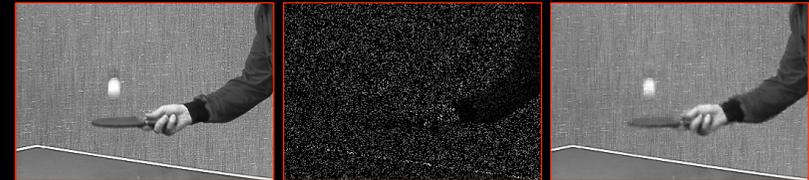
Experiments lead to state-of-the-art inpainting results.

Since 1699, when French explorers landed at the great bend of the Mississippi River and celebrated the first Mardi Gras in North America, New Orleans has brewed a fascinating melange of cultures. It was French (less Spanish, than French) agents that sold to the United States. Through all these years, and even into the 1900s, others arrived from everywhere: Acadians (Cajuns), African, indige-



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



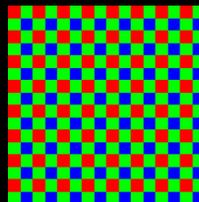
Original

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 ~ 0.2 dB 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)].

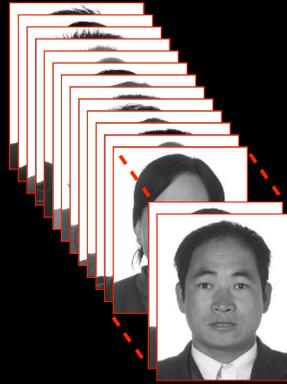


Image Compression

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

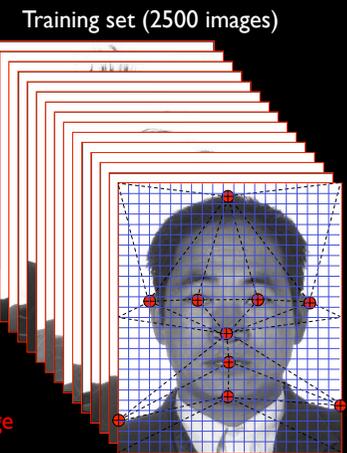


Image Compression Results

Original					
JPEG					
JPEG-2000					
Local-PCA					
K-SVD					

Results for 820 Bytes per each file

Local-PCA	11.99	10.49	8.81	5.56
K-SVD	10.83	9.92	7.89	4.82
Local-PCA	10.93	8.71	8.61	5.58

Image Compression Results

Original					
JPEG					
JPEG-2000					
Local-PCA					
K-SVD					

Results for 550 Bytes per each file

Local-PCA	15.81	13.89	10.66	6.60
K-SVD	14.67	12.41	9.44	5.49
Local-PCA	15.30	12.57	10.27	6.36

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 (I)

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 of 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

The training image: 717x717 pixels, providing a set of 54,289 training patch-pairs.

PSNR=17.00dB

Super-Resolution – Results (1)

An amazing variety of practical problems (design, analysis, and operation) can be modeled as optimization problems, or some variation thereof. Indeed, mathematical optimization has become a widely used tool in engineering, in electrical systems, and optimal design problems and aerospace engineering. Optimization is used in design and operation, finance, supply chain, and other areas. The list of applications is still growing.

For most of these applications, mathematical optimization is used as a human decision maker, system designer, process checker, and modifier when necessary. This human decision maker is typically the optimization problem, e.g., buying a portfolio.

SR Result
PSNR=16.95dB

Ideal Image

An amazing variety of practical problems (design, analysis, and operation) can be modeled as optimization problems, or some variation thereof. Indeed, mathematical optimization has become a widely used tool in engineering, in electrical systems, and optimal design problems and aerospace engineering. Optimization is used in design and operation, finance, supply chain, and other areas. The list of applications is still growing.

For most of these applications, mathematical optimization is used as a human decision maker, system designer, process checker, and modifier when necessary. This human decision maker is typically the optimization problem, e.g., buying a portfolio.

Bicubic interpolation
PSNR=14.68dB

Given Image

Super-Resolution – Results (2)



Given image

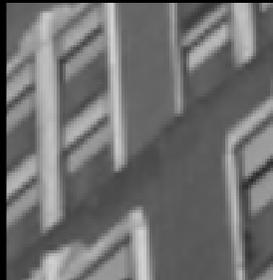


Scaled-Up (factor 2:1) using the proposed algorithm,
PSNR=29.32dB (3.32dB improvement over bicubic)

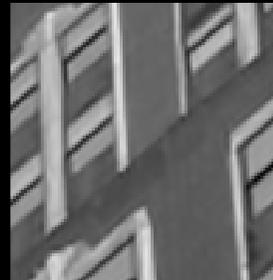
Super-Resolution – Results (2)



The Original



Bicubic Interpolation

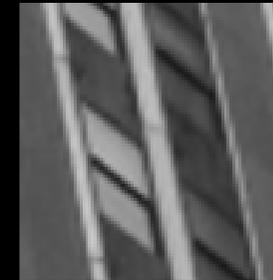


SR result

Super-Resolution – Results (2)



The Original



Bicubic Interpolation



SR result