Denoising and Beyond via Learned Dictionaries and Sparse Representations*

Michael Elad

The Computer Science Department
The Technion – Israel Institute of technology
Haifa 32000, Israel

המרכז הבינתחומי הרצלייה

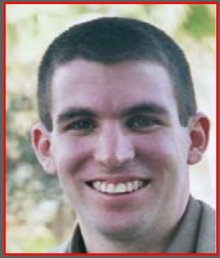
Israel Vision Day
Inter-Disciplinary Center
Herzlia, Israel
December 17, 2006

* Joint work with

Michal Aharon

Guillermo Sapiro

Julien Mairal



Matan Protter

Noise Removal ?

Our story starts with image denoising …

- 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, Example-based techniques, Sparse representations, …

Part I:

Sparse and Redundant Representations?

Denoising By Energy Minimization

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

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

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-Aposteriori Probability (MAP) estimation.
- Clearly, the wisdom in such an approach is within the choice of the prior – **modeling the images** of interest.

Thomas Bayes
1702 - 1761

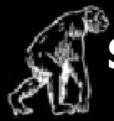
The Evolution Of $\Pr(\underline{x})$

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

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

Energy

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



Smoothness

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

Adapt+ Smooth

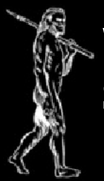
$$\Pr(\underline{x}) = \lambda \rho\{\mathbf{L}\underline{x}\}$$

Robust Statistics

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

Total-Variation

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



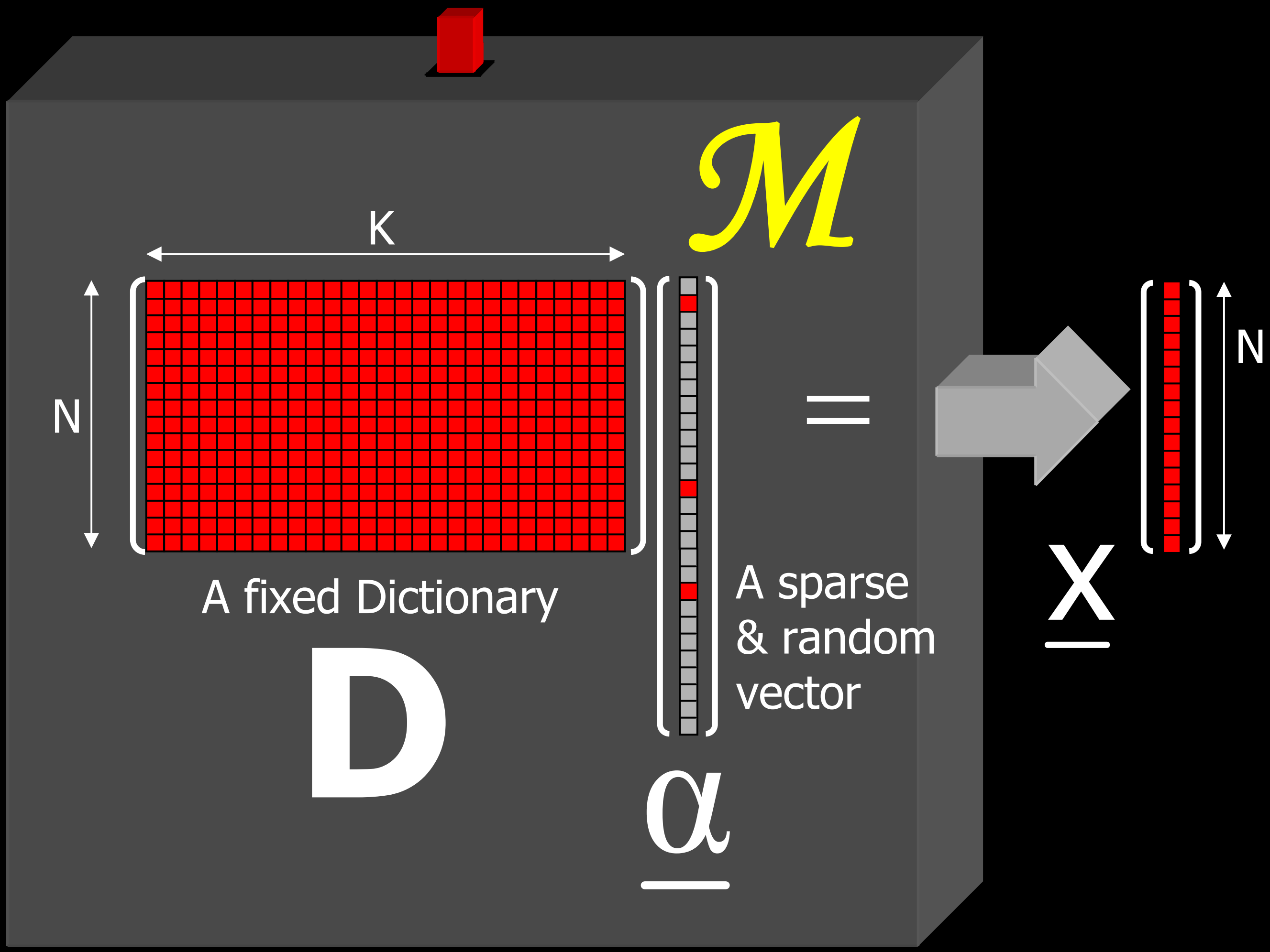
Wavelet Sparsity

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

Sparse & Redundant

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

The *Sparseland* Model for Images



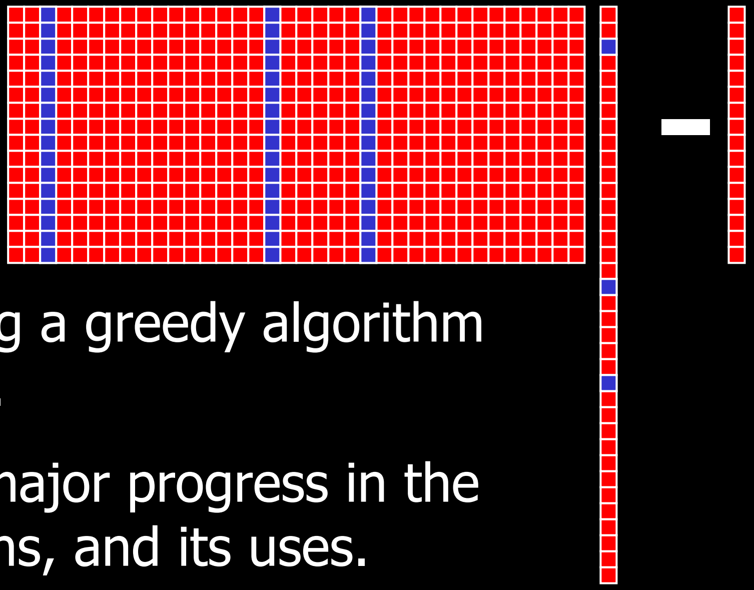
- Every column in **D** (dictionary) is a prototype signal (Atom).
- The vector $\underline{\alpha}$ is generated randomly with few (say L) non-zeros at random locations and with random values.

Our MAP Energy Function

- We L_0 norm is effectively counting the number of non-zeros in $\underline{\alpha}$.
- The vector $\underline{\alpha}$ is the representation (**sparse**/**redundant**).

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

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



- The above is solved (approximated!) using a greedy algorithm - the Matching Pursuit [Mallat & Zhang (`93)].
- In the past 5-10 years there has been a major progress in the field of sparse & redundant representations, and its uses.

What Should D Be?

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

Our Assumption: Good-behaved Images have a sparse representation

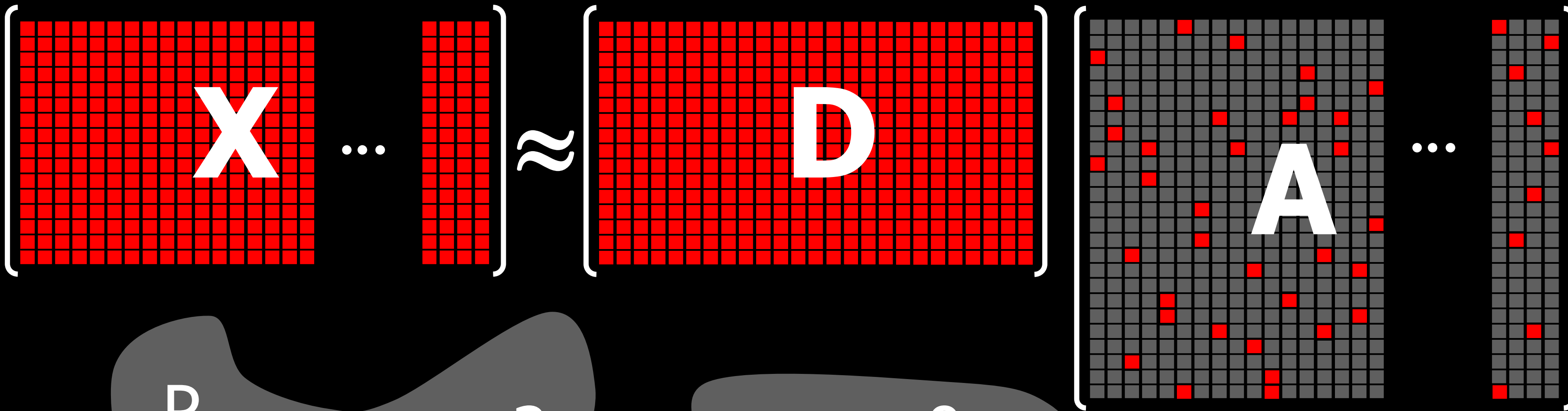
D should be chosen such that it sparsifies the representations

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

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

Part II:
Dictionary Learning: The K-SVD Algorithm

Measure of Quality for D



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

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

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

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

The K–SVD Algorithm – General

Aharon, Elad, & Bruckstein (`04,`05)

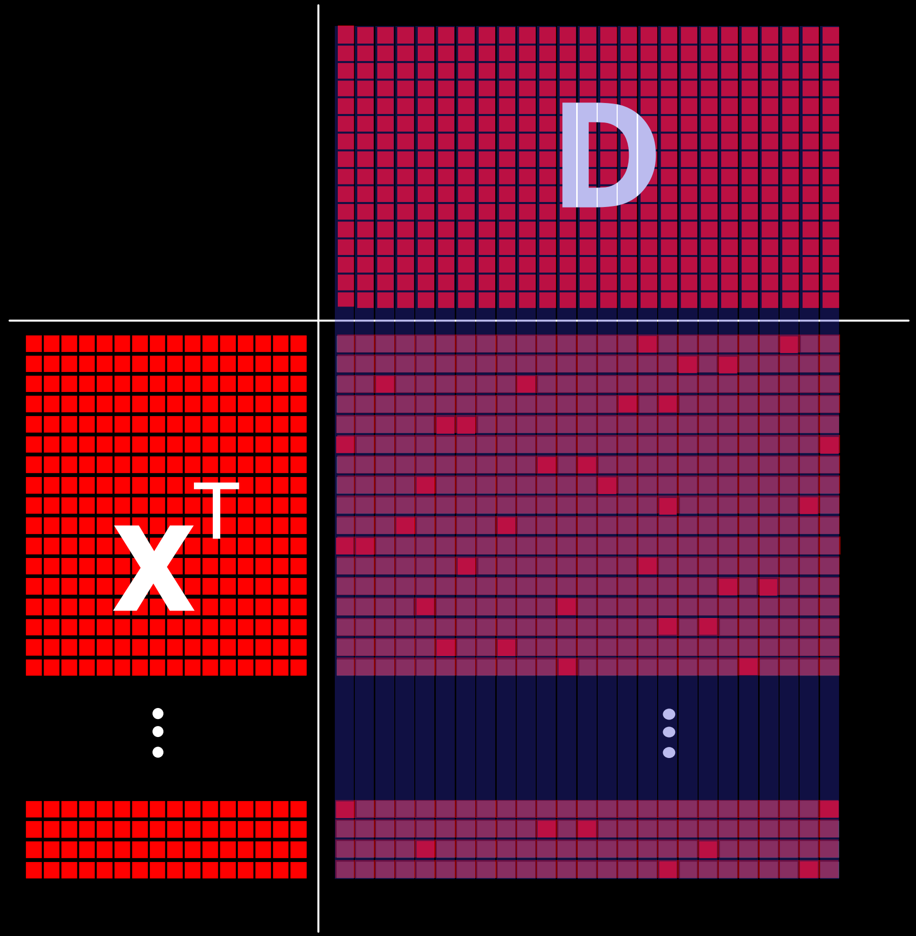
Initialize D

Sparse Coding

Use Matching Pursuit

Dictionary Update

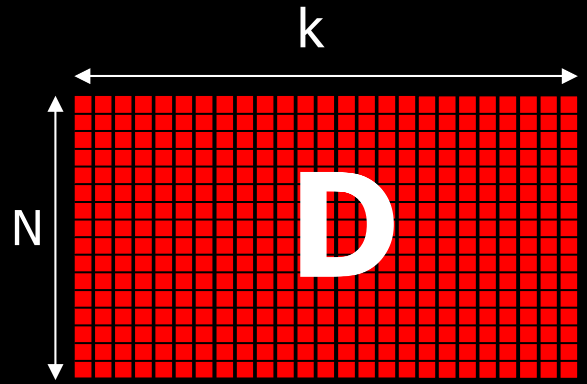
Column-by-Column by **SVD computation over the relevant examples**



Part III: Combining It All

From Local to Global Treatment

- The K-SVD algorithm is reasonable for low-dimension signals (N in the range 10-400). As N grows, the complexity and the memory requirements of the K-SVD become prohibitive.
- So, how should large images be handled?
- The solution: Force shift-invariant sparsity - on each patch of size N-by-N (N=8) in the image, including overlaps [Roth & Black (`05)].



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

Extracts a patch in the ij location

Our prior

What Data to Train On?

Option 1:

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

Option 2:

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

Application 2: Image Denoising

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

- The dictionary (and thus the image prior) is trained on the corrupted itself!
- This leads to an elegant fusion of the K-SVD and the denoising tasks.

Application 2: Image Denoising

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

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

Compute α_{ij} per patch

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

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

using the matching pursuit

$\underline{x}$ and α_{ij} known

Compute **D** to minimize

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

using SVD, updating one column at a time

K-SVD

D and α_{ij} known

Compute $\underline{x}$ by

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

which is a simple averaging of shifted patches

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

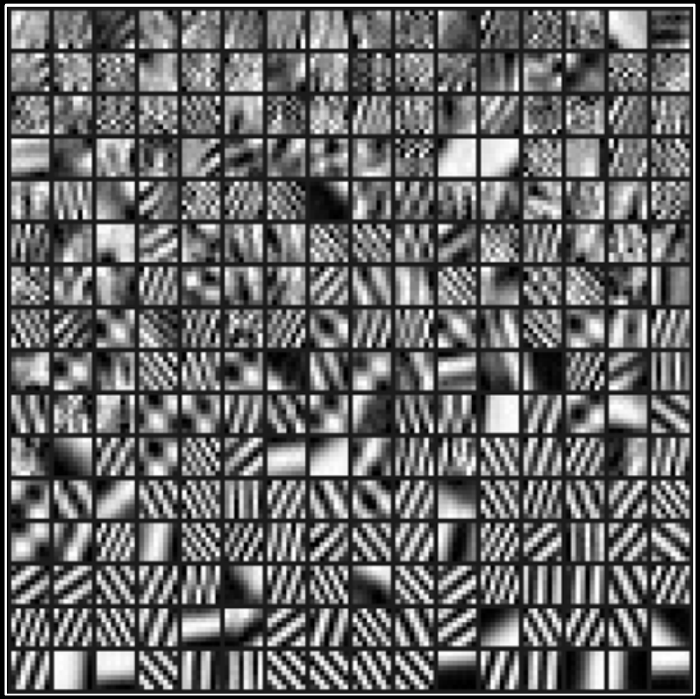
Image Denoising (Gray) [Elad & Aharon (`06)]

Source

Result 30.829dB

Noisy image

$\sigma = 20$



The obtained dictionary after 10 iterations

Image Denoising (Gray) [Elad & Aharon (`06)]

Source

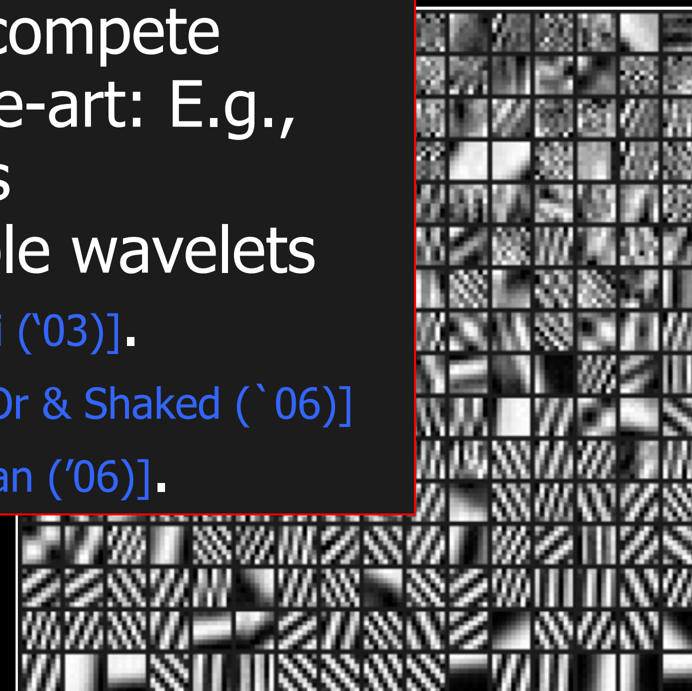
The results of this algorithm compete favorably with the state-of-the-art: E.g.,

- ❑ We get ~1dB better results compared to GSM+steerable wavelets [Portilla, Strela, Wainwright, & Simoncelli ('03)].
- ❑ Competitive works are [Hel-Or & Shaked (`06)] and [Rusanovskyy, Dabov, & Egiazarian ('06)].

Result 30.829dB

Noisy image

$\sigma = 20$



The obtained dictionary after 10 iterations

Denoising (Color) [Mairal, Elad & Sapiro, ('06)]

- When turning to handle color images, the direct generalization (working with R+G+B patches) leads to color artifacts.
- The solution was found to be a bias in the pursuit towards the color content.

Denoising (Color) [Mairal, Elad & Sapiro, ('06)]

Our experiments lead 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)

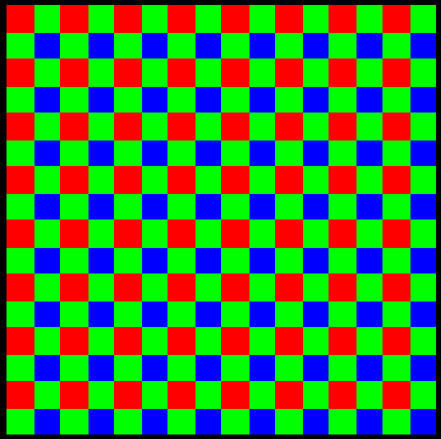
Original

Noisy (12.77dB)

Result (29.87dB)

Demosaicing [Mairal, Elad & Sapiro, ('06)]

- Today's cameras are sensing only one color per pixel, leaving the rest to be interpolated.
- Generalizing the previous scheme to handle demosaicing is tricky because of the possibility to learn the mosaic pattern within the dictionary.



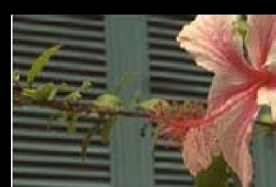
- In order to avoid "over-fitting", we have handled the demosaicing problem while forcing strong sparsity and only few iterations.
- The same concept can be deployed to inpainting.

Demosaicing [Mairal, Elad & Sapiro, ('06)]

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

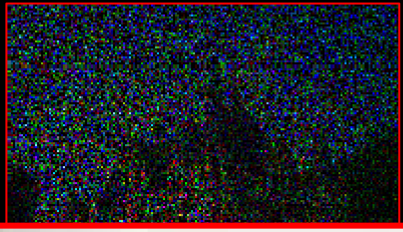




Inpainting [Mairal, Elad & Sapiro, ('06)]

Our experiments lead to state-of-the-art inpainting results.


Or

Video Denoising [Protter & Elad ('06)]

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

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

Video Denoising [Protter & Elad ('06)]

.62)

Our experiments lead 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)

Part IV:
To Conclude

Today We Have Seen that …

Sparsity, Redundancy, and the use of examples are important ideas, and can be used in designing better tools in signal/image processing

More specifically?

We have shown how these lead to state-of-the art results:

- K-SVD+Image denoising,
- Extension to color, and handling of missing values,
- Video denoising.

Michal Aharon

Guillermo Sapiro and Julien Mairal

Matan Protter

More on these (including the slides, the papers, and a Matlab toolbox) in http://www.cs.technion.ac.il/~elad