Sparse & Redundant Representations and Their Applications in Signal and Image Processing

Image Priors and the *Sparseland* Model

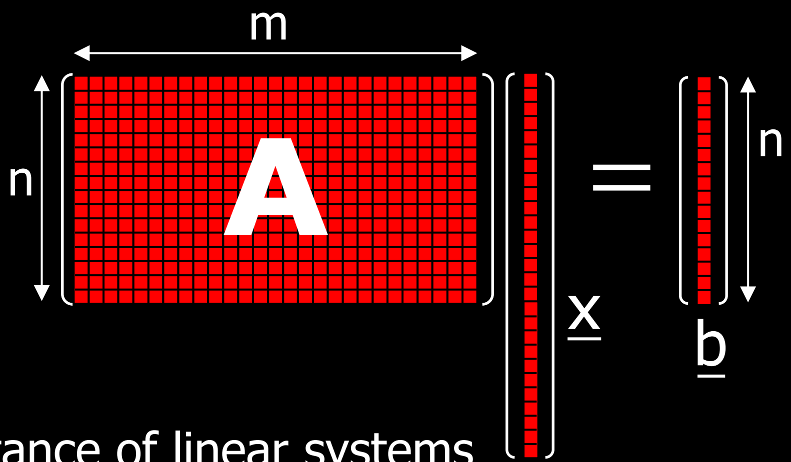
TECHNION
Israel Institute of Technology

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

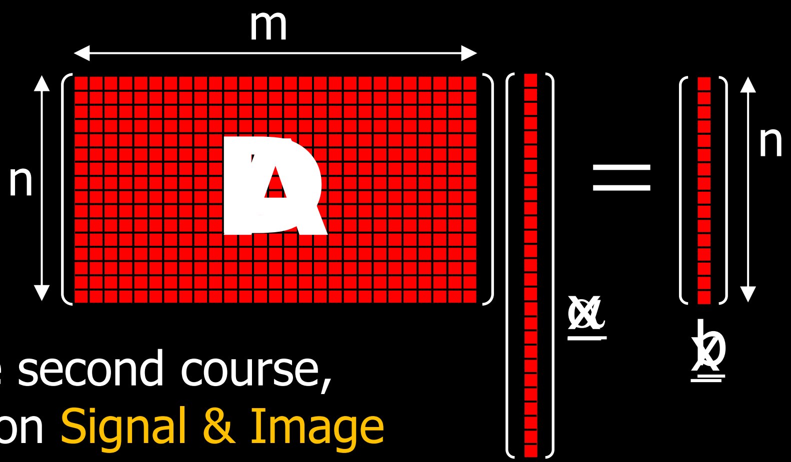
A Word About Notations

A Word About Notations

- This comment is meant for those of you who took the first course
- Obviously you recall the importance of linear systems of equations to our story
- Well, in the first course, we adopted a Linear Algebra point of view, and thus our notation for the linear system was **A**x=b

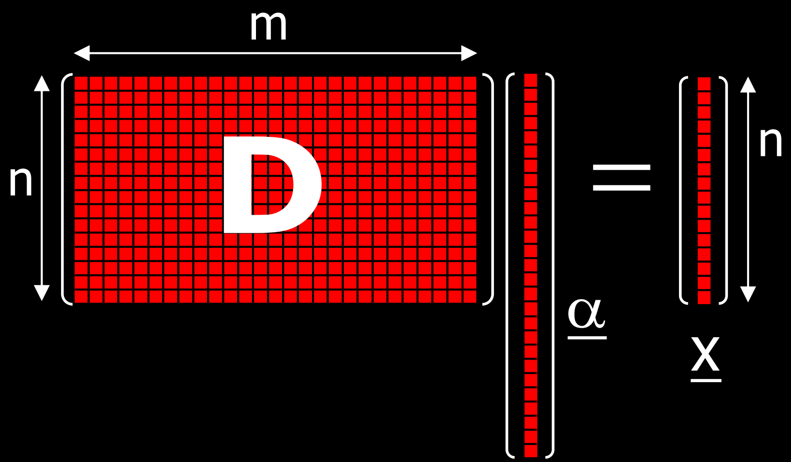


A Word About Notations



- As we enter the second course, which focuses on Signal & Image Processing, this notation will necessarily change to $\mathbf{D}\underline{\alpha}=\underline{x}$
- Now $\underline{x}$ will serve as a signal of interest, $\mathbf{D}$ is the dictionary, and $\underline{\alpha}$ is the signal's representation

A Word About Notations



I hope that you will not find this too confusing

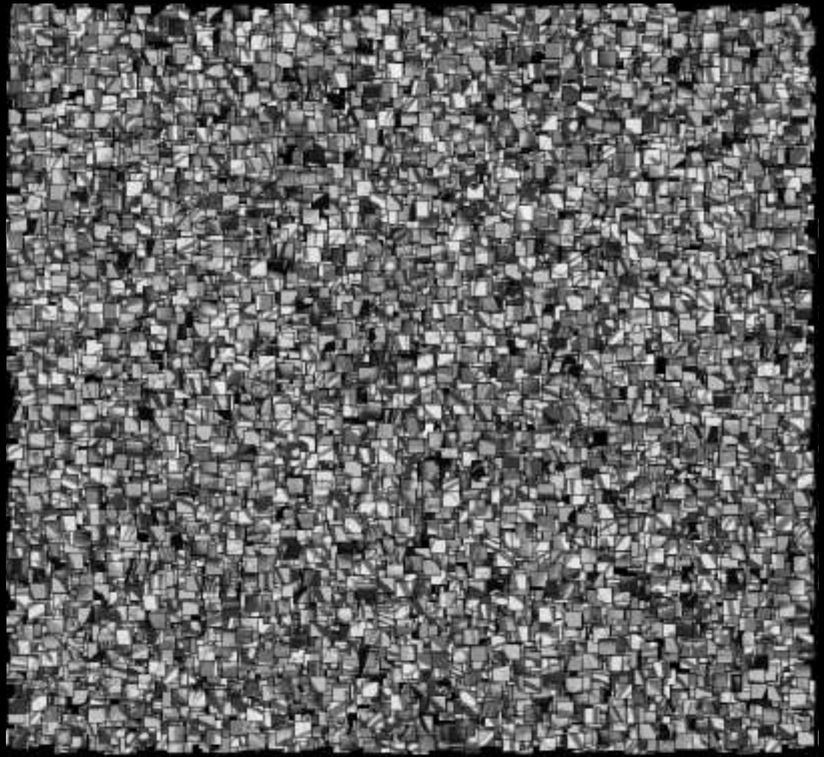
A Prior for Images:

How and Why?

A Virtual Experiment

- Suppose we accumulate many millions of image patches, each of size 20×20 pixels
- Clearly, every such image is a point in $\mathbb{R}^{400}$
- Let's put these points in this 400-dim. Euclidean space, in the cube $[0,1]^{400}$
- Now, LET'S STEP INTO THIS SPACE and look at the cloud of points we just generated

What are we expected to see?



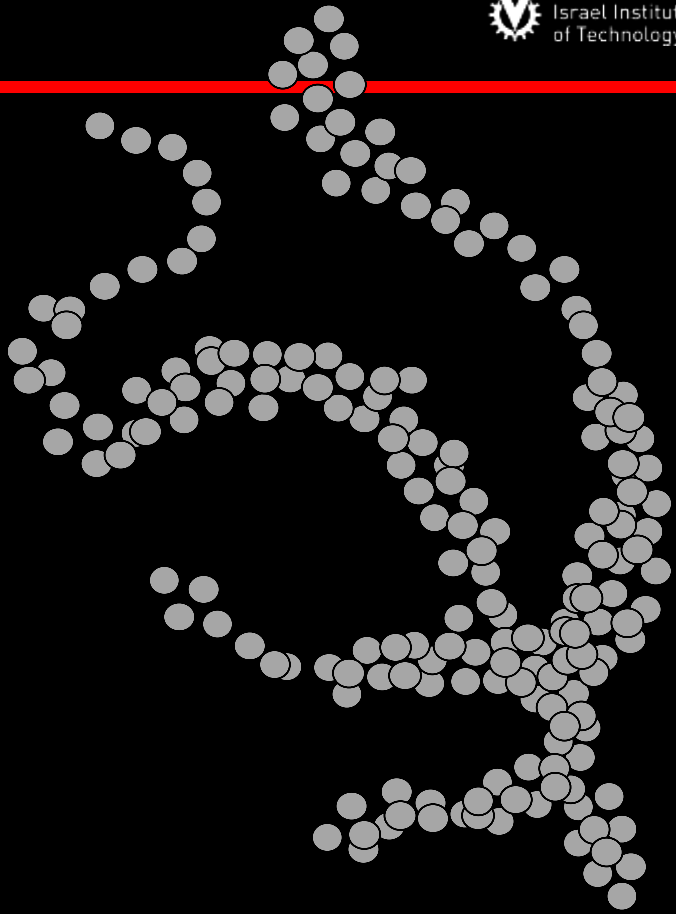
A Virtual Experiment

What are we expected to see?

1. Deserts! Vast emptiness!
2. Concentration of points in some regions
3. Filaments, manifold structure …
4. Different densities from one place to another

In this experiment we have actually created an empirical estimate of the Probability Density Function (PDF) of … image patches

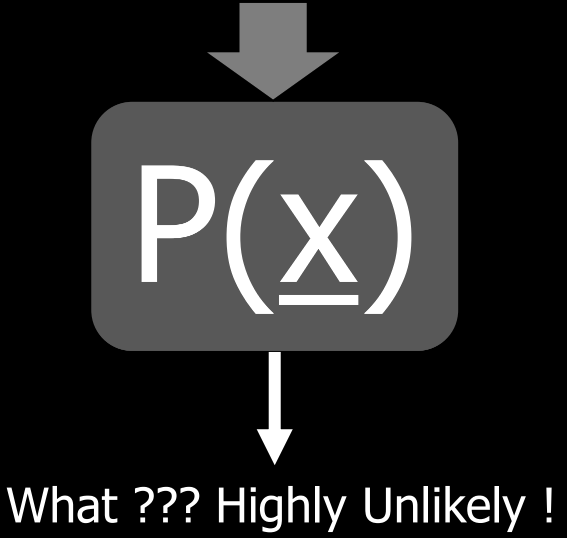
Call it $P(\underline{x})$



So, Lets Talk About … P($\underline{x}$)

- We "experimented" with small images, but the same phenomena will be found in audio, seismic data, financial data, text-files, … and practically any source of information you are familiar with
- Nevertheless, we stick to images in this course
- Imagine this: a function that can be given an image and returns its chances to exist! This is amazing, don't you think?
- What could you do with such a function?

Answer: EVERYTHING



Everything ? Can you Remove Noise ?

- In the denoising problem, the measurement is

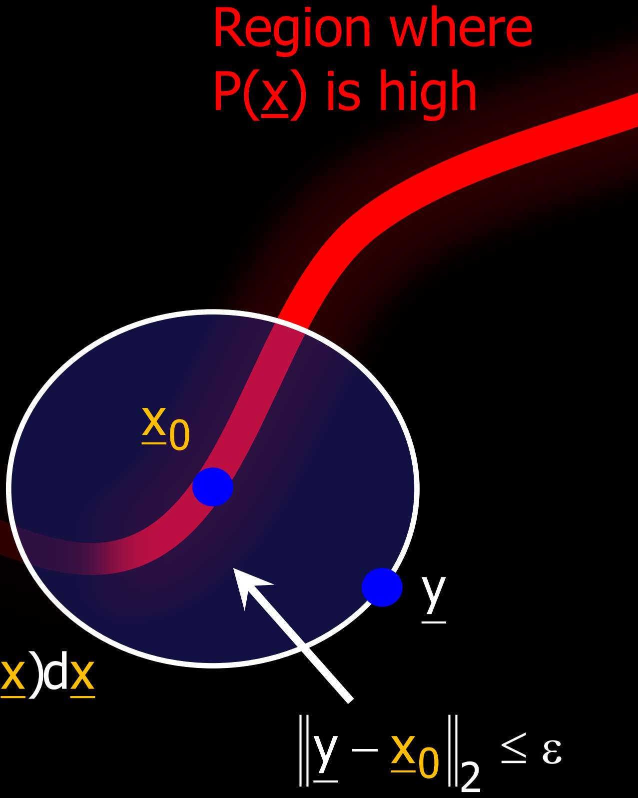
$$\underline{y} = \underline{x}_0 + \underline{v}, \quad \left\|\underline{v}\right\|_2 \le \varepsilon$$

- Our goal: Recover $\underline{x}_0$ from $\underline{y}$
- Given $P(\underline{x})$, we can suggest a recovery of $\underline{x}_0$ by
  - Option 1 (MAP):

$$\hat{\underline{x}} = \underset{\underline{x}}{\mathrm{ArgMax}}\, P(\underline{x}) \quad \text{s.t.} \quad \left\|\underline{y} - \underline{x}\right\|_2 \le \varepsilon$$

  - Option 2 (MMSE):

$$\hat{\underline{x}} = E\left\{\underline{x} \;\middle|\; \left\|\underline{y} - \underline{x}\right\|_2 \le \varepsilon\right\} = \int_{\left\|\underline{y}-\underline{x}\right\|_2 \le \varepsilon} \underline{x}\, P(\underline{x})\, d\underline{x}$$



What About General Inverse Problems ?

Region where $P(\underline{x})$ is high

- In a general inverse problem, the measurement is

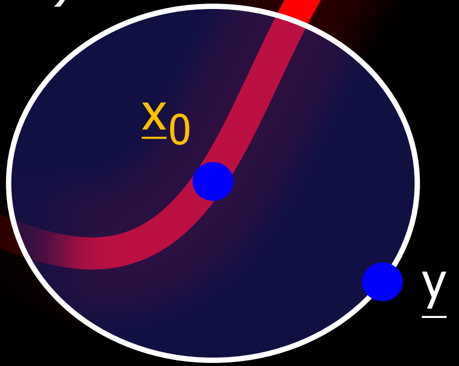
$$\underline{y} = \mathbf{C}\underline{x}_0 + \underline{v}, \quad \left\|\underline{v}\right\|_2 \le \varepsilon$$

where **C** is a general linear degradation operator (blur, projection, downscaling, subsampling, holes, …)

- Our goal: Recover $\underline{x}_0$ from $\underline{y}$
- Given $P(\underline{x})$, we can suggest a recovery of $\underline{x}_0$ by

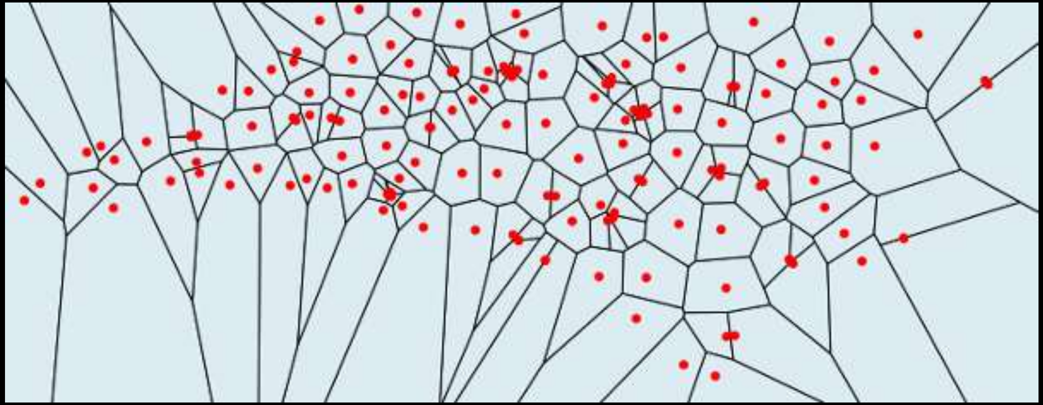
$$\hat{\underline{x}} = \underset{\underline{x}}{\mathrm{ArgMax}}\, P(\underline{x}) \quad \text{s.t.} \quad \left\|\underline{y} - \mathbf{C}\underline{x}\right\|_2 \le \varepsilon$$

[An MMSE version also exists, naturally]



Can it Help in Compression ?

- We are given $\underline{x}$ from an ensemble of images, along with its distribution PDF, $P(\underline{x})$



- We are also given a budget of B bits to represent $\underline{x}$, where our goal is to get the best possible compression (i.e. minimize the error)
- The approach we take is to divide the whole space into 2^B disjoint sets (Voronoi) and minimize the error w.r.t. the representation vectors (VQ):

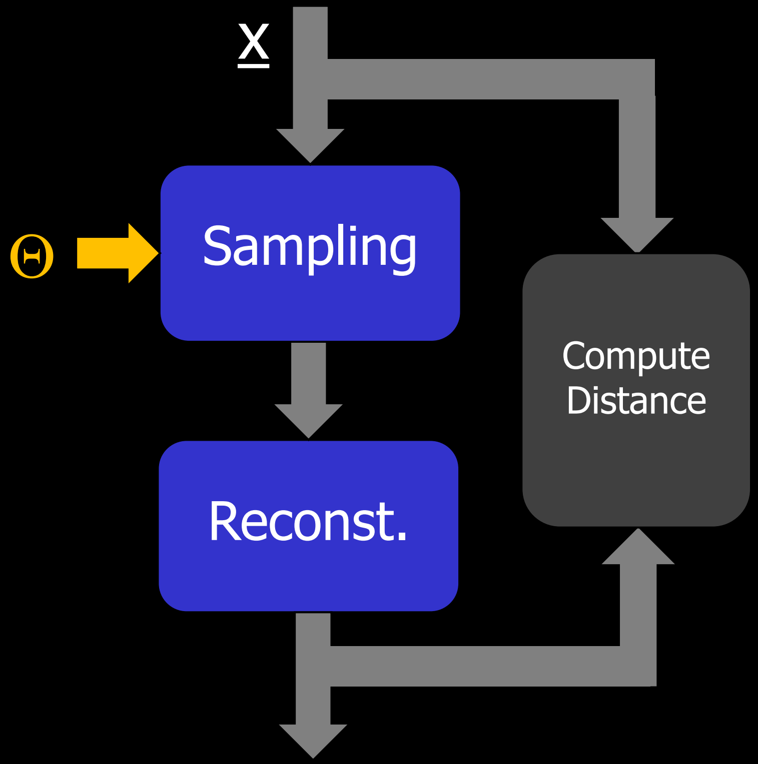
$$\underset{\{\underline{x}_k\}_k}{\text{Min}} \sum_{k=1}^{2^B} \int_{\underline{x} \in S_k} \left\| \underline{x} - \underline{x}_k \right\|_2^2 P(\underline{x}) d\underline{x}$$

Putting aside entropy coding, Vector Quantization is the best you could do in this case

Sampling ?

- The sampling operation relies on some chosen parameters, such as the basis functions to project upon, and their quantity
- Our goal is to propose sampling and reconstruction strategies, each (or just the first) is parameterized, and optimize the parameters for the smallest possible error:

$$\underset{\Theta}{\text{Min}} \int \left\| \underline{x} - \text{Reconst}\left\{ \text{Sample}_{\Theta}\left\{ \underline{x} \right\} \right\} \right\|_2^2 P(\underline{x}) d\underline{x}$$



Separation ?

- We are given a noisy mixture of the form:

$$\underline{y} = \underline{x}_1 + \underline{x}_2 + \underline{v}, \quad \left\|\underline{v}\right\|_2 \le \varepsilon$$

where $\underline{x}_1$ and $\underline{x}_2$ are two different signals from two different distributions, P_1 and P_2

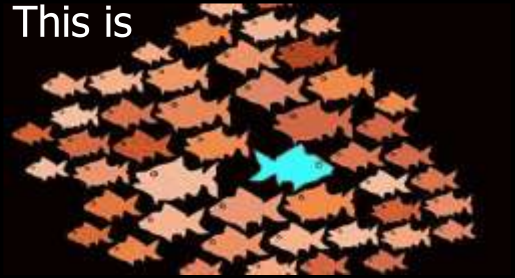
- Our goal is to separate the signal into its ingredients:

$$\underset{\underline{x}_1, \underline{x}_2}{\text{ArgMax}} \; P_1(\underline{x}_1) + P_2(\underline{x}_2) \quad \text{s.t.} \quad \left\|\underline{y} - \underline{x}_1 - \underline{x}_2\right\|_2 \le \varepsilon$$

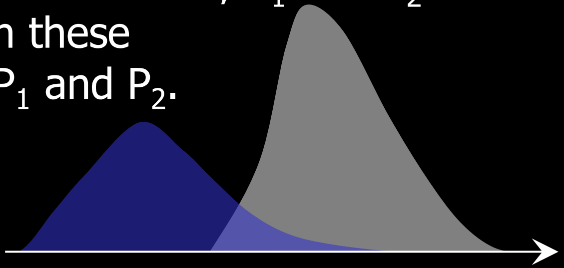
What Else ?

TECHNION
Israel Institute
of Technology

Anomaly Detection: We are given $\underline{x}$ and we are supposed to say if it is an anomaly. This is done by testing $P(\underline{x})<T$



Recognition: We are given a signal $\underline{x}$ that may belong to one of two possible sets, S_1 and S_2. The distributions within these two sets are given by P_1 and P_2. The decision will be made by $P_1(\underline{x})>P_2(\underline{x}) \rightarrow \underline{x}\in S_1$



Synthesis or Hallucinations: Given the PDF $P(\underline{x})$, we can synthesize artificial signals from it that obey the original distribution

● ● ●

Bottom Line

Question: What $P(\underline{x})$ is good for?

Answer: Great many things

Denoising, Interpolation, Prediction, Compression, Inference, Separation, Anomaly detection, Clustering, Summarizing, Segmentation, Style-changing, Conversion, Matching, Recognition, Indexing, Semi-supervised learning, Identification, Classification, Synthesis, Detection, …

So, … how do we find

$P(\underline{x})$?

We shall focus our efforts on $P(\underline{x})$ for images

The Evolution of Priors in Image Processing

Image Priors P(x)

- Here is an untold secret:

The vast literature in image processing over the past 4-5 decades is

ALMOST NOTHING BUT

an evolution of ideas on the identity of P(x), and ways to use it in actual tasks

- By the way, the same is true for many other data sources and signals …

So, Who is P($\underline{x}$) for Images ?

TECHNION
Israel Institute of Technology

- The very first attempts concentrated on the L_2-smoothness assumption – "*images are more likely if they are smooth*"
- Forcing smoothness is equivalent to penalizing the image derivatives (**L** – the Laplacian)

smoothness

$$\sim e^{-c \cdot \|\mathbf{L}\underline{x}\|_2^2}$$

70's 80's 90's 00's 10's time

So, Who is P($\underline{x}$) for Images ?

TECHNION Israel Institute of Technology

- The very first attempts concentrated on the L_2-smoothness assumption – "*images are more likely if they are smooth*"
- Forcing smoothness is equivalent to penalizing the image derivatives (**L** – the Laplacian)
- This led to the first instance of the Wiener filter for image denoising and deblurring:
- Benefits: L_2 is easy to handle, leading to a closed form solution
- Drawback: Wiener filter results suck!

$$\hat{\underline{x}} = \underset{\underline{x}}{\text{ArgMax}}\, P(\underline{x}) \quad \text{s.t.} \quad \|\underline{y} - \mathbf{C}\underline{x}\|_2 \le \varepsilon$$

$$\hat{\underline{x}} = \underset{\underline{x}}{\text{ArgMin}}\, \|\mathbf{L}\underline{x}\|_2^2 \quad \text{s.t.} \quad \|\underline{y} - \mathbf{C}\underline{x}\|_2 \le \varepsilon$$

$$\hat{\underline{x}} = \underset{\underline{x}}{\text{ArgMin}}\, \lambda\|\mathbf{L}\underline{x}\|_2^2 + \|\underline{y} - \mathbf{C}\underline{x}\|_2^2$$

$$\hat{\underline{x}} = \left[\lambda\mathbf{L}^T\mathbf{L} + \mathbf{C}^T\mathbf{C}\right]^{-1}\mathbf{C}^T\underline{y}$$

70's 80's 90's 00's 10's time

Using Transforms for Building P($\underline{x}$)

- Almost in parallel, transforms were used to construct P($\underline{x}$)
- Here **T** is some chosen transform (DCT, Fourier, …), and Λ is a diagonal non-negative matrix
- This is the prior that the JPEG algorithm relies upon so well

smoothness

$$\sim e^{-c\cdot\|\mathbf{L}\underline{x}\|_2^2}$$

Transform

$$\sim e^{-\|\Lambda\mathbf{T}\underline{x}\|_2^2}$$

- Observe that both these options effectively assume a multivariate Gaussian distribution*
- In fact, the two can be made equivalent if

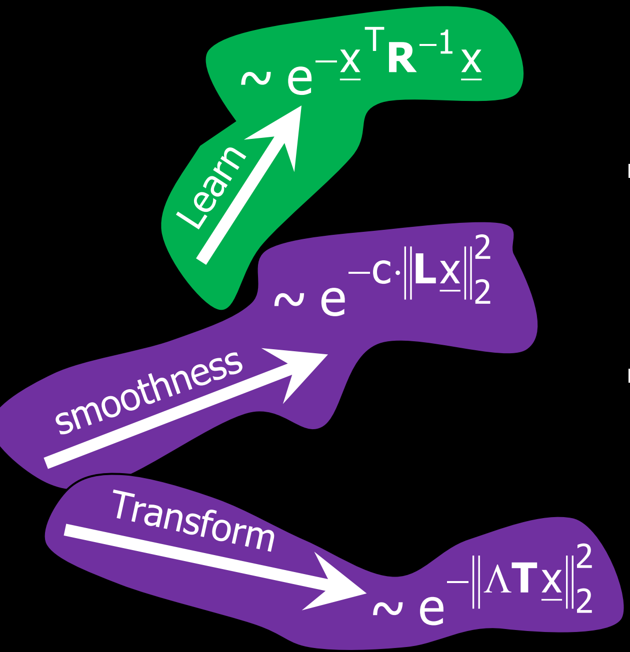
$$\mathbf{T}^T\Lambda^2\mathbf{T} = c\mathbf{L}^T\mathbf{L}$$

* If L_2 is so poorly performing, how come JPEG is so successful? We will say something about this later

70's 80's 90's 00's 10's time

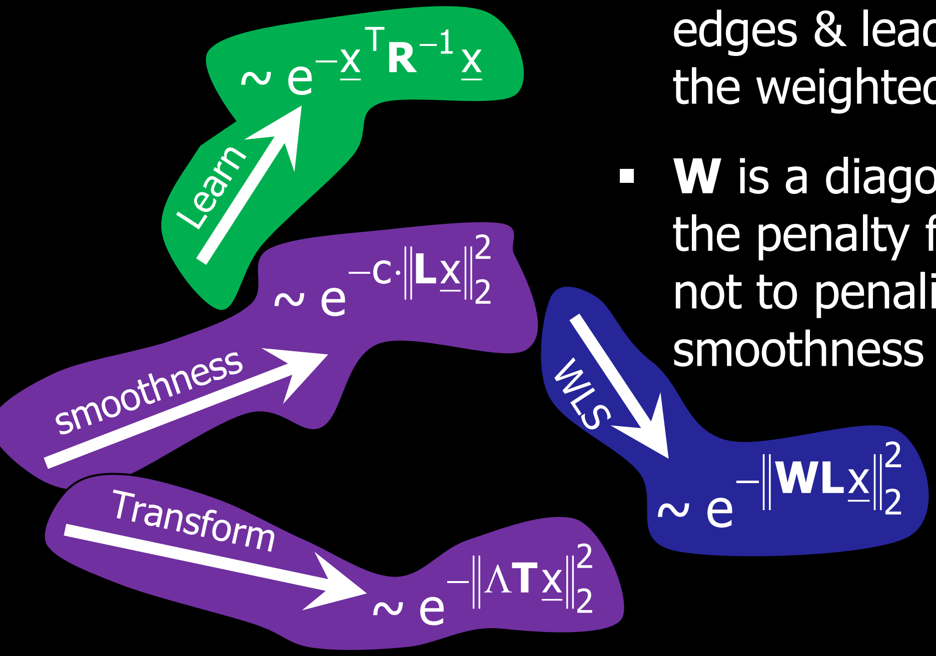
Adapting the Model to Actual Images

- Still under the Gaussian regime, came the KLT, which is the same as PCA
- The idea: learn the autocorrelation matrix instead of "guessing" it, as we have done before
- At least for small image patches, this was shown to be almost the same as 2D-DCT



What about the Over-Smoothing ?

TECHNION
Israel Institute
of Technology



- Realizing the L_2 is unforgiving to edges & leading to over-smoothing, the weighted L_2 prior was proposed
- **W** is a diagonal matrix that reduces the penalty for "edge-pixels", so as not to penalize their lack of smoothness
- This is still following the Gaussian regime of distributions

A New Era in Image Processing: L_1

- In the late 80's, it became clear that L_2-based priors (and the linear filtering they lead to) cannot deliver the desired quality
- The alternative came in various forms: Robust-Statistics for handling outliers, Partial Differential Equations, and even Wavelets sparsity
- Common to all: assume a heavy-tailed distribution

PDE $\rightarrow \sim e^{-\lambda TV(\underline{x})}$

Robust stat. $\rightarrow \sim e^{-\lambda \|\mathbf{L}\underline{x}\|_1}$

Wavelets $\rightarrow \sim e^{-\lambda \|\mathbf{T}\underline{x}\|_1}$

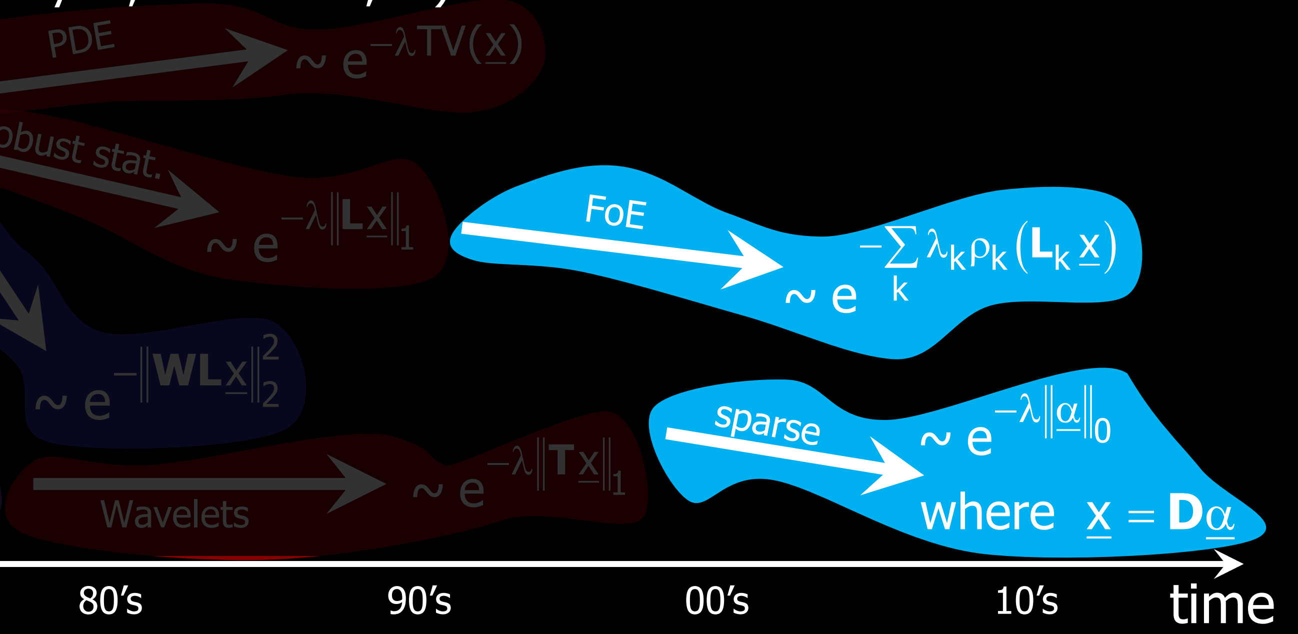
- Observation 1: All rely on L_1 !!
- Observation 2: All these led to a systematic way to design non-linear filtering algorithms

70's | 80's | 90's | 00's | 10's | time

The Most Recent Priors

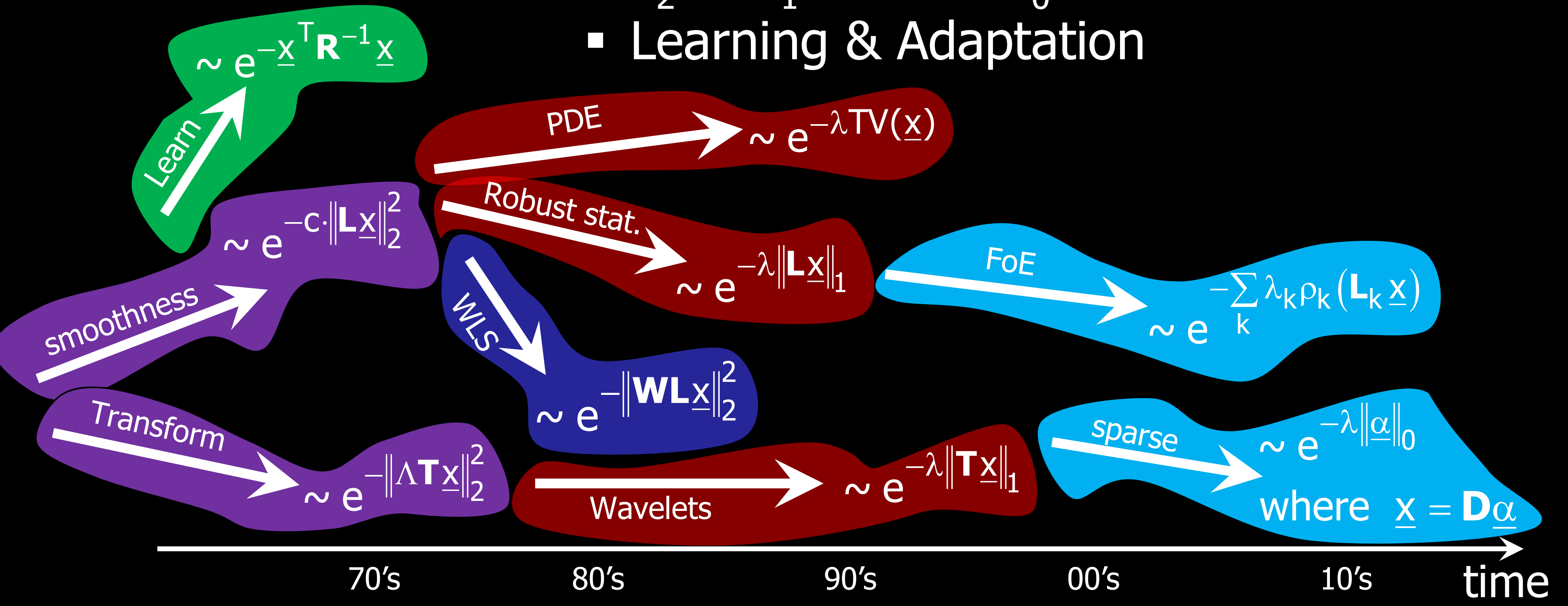
- The more recent and more effective comers to this game of building $P(\underline{x})$ are *Sparseland* using L_0, the Field-of-Expert, and more (GMM, Co-sparse Analysis, Low-Rank, …)

- Common to these:
  - Adapt the prior to the data by learning the parameters, very similar to the approach taken by PCA
  - Sparsity is key in forming the model, either explicitly or implicitly



The Main Themes in this Evolution

- $L_2 \rightarrow L_1$ or even L_0
- Learning & Adaptation



The Evolution of Image Priors

Observe that all the expressions we proposed for $P(\underline{x})$ have a Gibbs distribution form $P(\underline{x})=C\cdot\exp\{-G(\underline{x})\}$:

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


Energy


Smoothness


Adapt+ Smooth


Robust Statistics

$$G(\underline{x}) = \lambda \|\nabla\underline{x}\|_1 \qquad G(\underline{x}) = \lambda \|\mathbf{W}\underline{x}\|_1 \qquad G(\underline{x}) = \lambda \|\underline{\alpha}\|_0 \text{ for } \underline{x} = \mathbf{D}\underline{\alpha}$$


Total-Variation


Wavelets Sparsity


Sparseland

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

Linear vs. Non-Linear Approximation

$L_2 \rightarrow L_1$ (or even L_0)

- There are many ways to interpret the migration from L_2 to $L_{1/0}$:
  - Moving to heavy-tailed distributions
  - Handling better outliers (edge-pixels)
  - Getting to a non-linear estimation algorithm, or
  - Migration from a linear to a non-linear approximation
- Let us expand on the last interpretation as it is key in our story

Starting with the L_2 Option

- Suppose that our prior is the following (the matrix **T** is unitary, e.g. DCT):

$$P(\underline{x}) \sim e^{-\|\Lambda \mathbf{T} \underline{x}\|_2^2}$$

- The matrix Λ contains the weights of the transform elements:

$$\Lambda = \begin{bmatrix} \lambda_1 & & & & & \\ & \ddots & & & 0 & \\ & & \lambda_k & & & \\ & & & \lambda_{1+1} & & \\ & 0 & & & \ddots & \\ & & & & & \lambda_n \end{bmatrix} = \begin{bmatrix} \varepsilon & & & & & \\ & \ddots & & & 0 & \\ & & \varepsilon & & & \\ & & & 100 & & \\ & 0 & & & \ddots & \\ & & & & & 100 \end{bmatrix}$$

Starting with the L_2 Option

- Our goal: Denoising a signal with this $P(\underline{x})$

$$\hat{\underline{x}} = \underset{\underline{x}}{\text{ArgMax}}\, P(\underline{x}) \quad \text{s.t.} \quad \left\| \underline{y} - \underline{x} \right\|_2 \leq \varepsilon$$

- Or, more conveniently, by

$$\hat{\underline{x}} = \underset{\underline{x}}{\text{ArgMin}} \left\| \underline{y} - \underline{x} \right\|_2^2 - \log\left\{ P(\underline{x}) \right\}$$

- This leads us to

$$P(\underline{x}) \sim e^{-\left\| \Lambda \mathbf{T} \underline{x} \right\|_2^2} \Rightarrow \min_{\underline{x}} \frac{1}{2} \left\| \underline{x} - \underline{y} \right\|_2^2 + \frac{1}{2} \left\| \Lambda \mathbf{T} \underline{x} \right\|_2^2$$

(the factor 1/2 is there for mathematical convenience)

The L_2 Solution

$$\min_{\underline{x}} \frac{1}{2}\left\|\underline{x}-\underline{y}\right\|_2^2 + \frac{1}{2}\left\|\Lambda \mathbf{T}\underline{x}\right\|_2^2$$

$$\hat{\underline{x}}_{opt} = \left[I + \mathbf{T}^T\Lambda^T\Lambda\mathbf{T}\right]^{-1}\underline{y} = \mathbf{T}^T\left[I+\Lambda^2\right]^{-1}\mathbf{T}\underline{y}$$

$$= \mathbf{T}^T\begin{bmatrix} \frac{1}{1+\lambda_1^2} & & \\ & \ddots & \\ & & \frac{1}{1+\lambda_n^2}\end{bmatrix}\mathbf{T}\underline{y} = \mathbf{T}^T\begin{bmatrix} \frac{1}{1+\varepsilon^2} & & \\ & \ddots & \\ & & \frac{1}{101}\end{bmatrix}\mathbf{T}\underline{y}$$

- Implication: do not touch the leading transform coefficients and remove the rest
- The decision who survives is fixed by Λ
- This is Linear Approximation

Moving to the $L_{1/0}$ Option

- Suppose that our prior is the following (**T** is unitary, as before), where p=0 or 1:

$$P(\underline{x}) \sim e^{-\lambda \|\mathbf{T}\underline{x}\|_p}$$

- Observe that we do not have a weights matrix Λ, and a simple scalar λ is sufficient here

- Denoising this time: $\min_{\underline{x}} \frac{1}{2}\|\underline{x}-\underline{y}\|_2^2 + \lambda \|\mathbf{T}\underline{x}\|_p$

- Surprisingly, this has a closed-form solution due to the orthogonality of **T** – let's show this

The $L_{1/0}$ Solution

$$\min_{\underline{x}} \frac{1}{2}\left\|\underline{x}-\underline{y}\right\|_2^2 + \lambda\left\|\mathbf{T}\underline{x}\right\|_p$$

Define $\underline{z}=\mathbf{T}\underline{x}$

$$\frac{1}{2}\left\|\mathbf{T}^T\underline{z}-\underline{y}\right\|_2^2 + \lambda\left\|\underline{z}\right\|_p$$

$$\underset{\substack{\uparrow \\ \mathbf{T}^T\mathbf{T}=\mathbf{I}}}{=} \frac{1}{2}\left\|\mathbf{T}^T\left(\underline{z}-\mathbf{T}\underline{y}\right)\right\|_2^2 + \lambda\left\|\underline{z}\right\|_p$$

$$\underset{\substack{\uparrow \\ \left\|\mathbf{T}^T\underline{v}\right\|_2=\left\|\underline{v}\right\|_2}}{=} \frac{1}{2}\left\|\underline{z}-\mathbf{T}\underline{y}\right\|_2^2 + \lambda\left\|\underline{z}\right\|_p$$

The $L_{1/0}$ Solution

- Our goal is equivalent to ($\mathbf{T}y=\underline{z}_0$)

$$\min_{\underline{z}} \frac{1}{2}\left\|\underline{z}-\underline{z}_0\right\|_2^2+\lambda\left\|\underline{z}\right\|_p$$

$$=\min_{\underline{z}} \sum_{k=1}^{n}\left\{\frac{1}{2}\left(z_k-z_0^k\right)^2+\lambda\left|z_k\right|^p\right\}$$

- The problem has decomposed into n separate 1D-optimization tasks of the form

$$\min_{z} \frac{1}{2}\left(z-z_0\right)^2+\lambda\left|z\right|^p$$

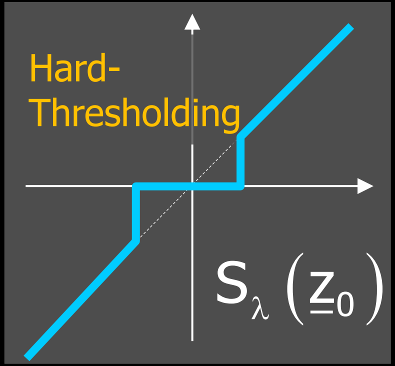
- Let's assume that p=0 (i.e., the L_0-norm), as it is simpler to analyze

The $L_{1/0}$ Solution

$$\underset{z}{\text{Min}} \ \frac{1}{2}\left(z-z_0\right)^2 + \lambda |z|^0$$

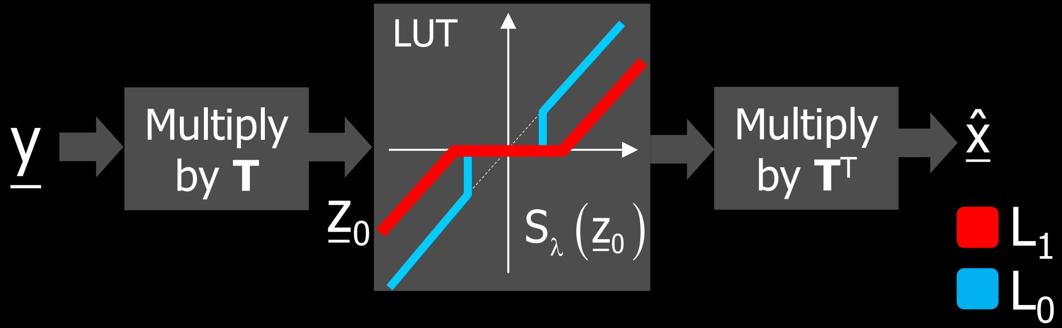
- The unknown, z, could be either =0 or $\neq 0$
  - If z=0, the penalty is $0.5z_0^2$
  - If $z \neq 0$, then choose $z=z_0$ and then the penalty is … λ

$$z_{opt} = \begin{cases} 0 & |z_0| \leq \sqrt{2\lambda} \\ z_0 & \text{Otherwise} \end{cases}$$



The $L_{1/0}$ Solution

$$\min_{\underline{x}} \frac{1}{2}\left\|\underline{x}-\underline{y}\right\|_2^2 + \lambda \left\|\mathbf{T}\underline{x}\right\|_p \Rightarrow \hat{\underline{x}}_{opt} = \mathbf{T}^T S_\lambda\left\{\mathbf{T}\underline{y}\right\}$$



- Implication: Just like before, some transform coefficients are nulled while others stay "intact"
- However, the decision who survives is based on the coefficients' magnitude themselves
- This is Non-Linear Approximation

Linear vs. Nonlinear Approximation

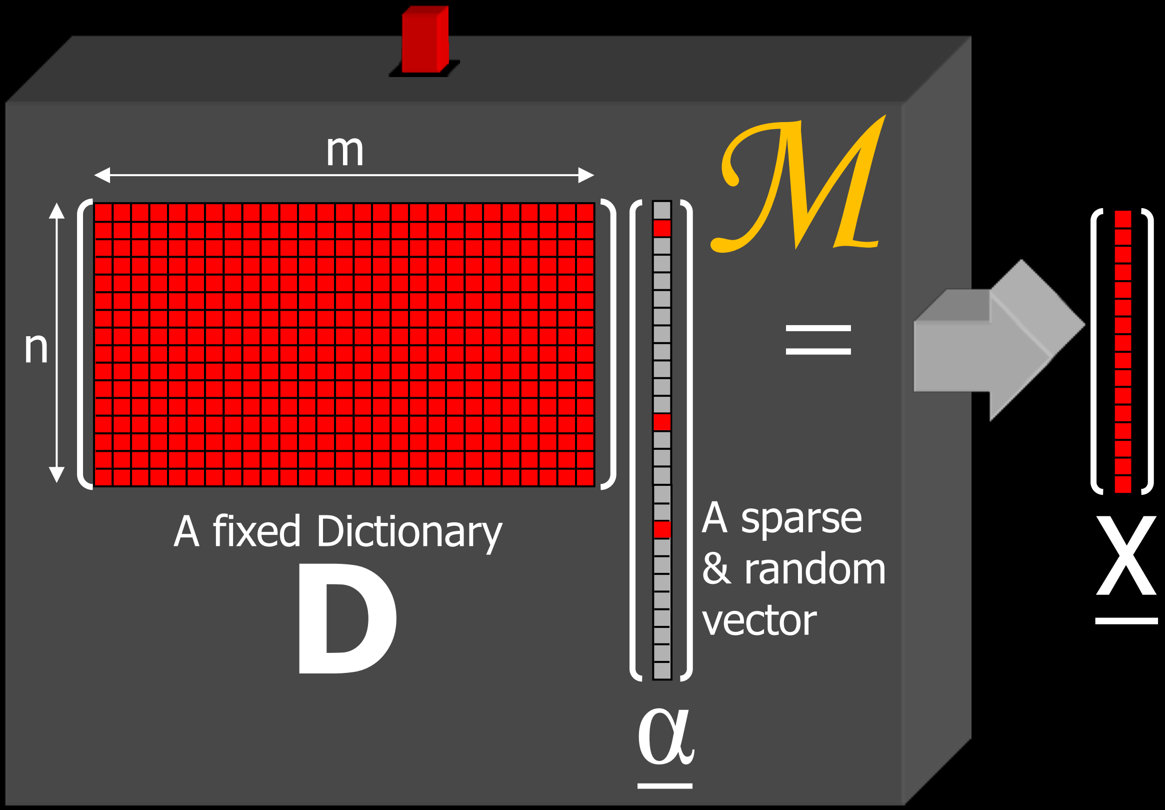
Linear	Non-Linear
$\min_{\underline{x}} \frac{1}{2}\left\|\underline{x}-\underline{y}\right\|_2^2 + \frac{1}{2}\left\|\Lambda \mathbf{T}\underline{x}\right\|_2^2$	$\min_{\underline{x}} \frac{1}{2}\left\|\underline{x}-\underline{y}\right\|_2^2 + \lambda\left\|\mathbf{T}\underline{x}\right\|_p$
$\hat{\underline{x}}_{opt} = \mathbf{T}^T\left[I+\Lambda^2\right]^{-1}\mathbf{T}\underline{y}$	$\hat{\underline{x}}_{opt} = \mathbf{T}^T S_\lambda\left\{\mathbf{T}\underline{y}\right\}$

Back to JPEG: Is it really a pure linear approximation based scheme?

The *Sparseland* Model

Sparseland: A Generative Model

TECHNION Israel Institute of Technology



Sparseland: A Generative Model

Draw k_0 – the cardinality of $\underline{\alpha}$

$P(k) \sim e^{-\lambda k}$

Draw k_0 non-zero values

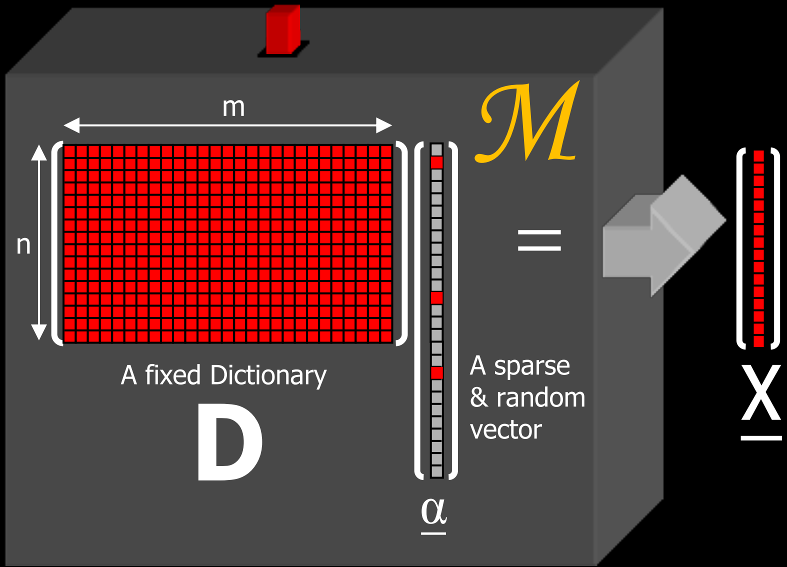
$P(\underline{v}) \sim e^{-\beta\|\underline{v}\|_2^2}$

Draw k_0 locations (e.g. uniformly)

Generate the representation

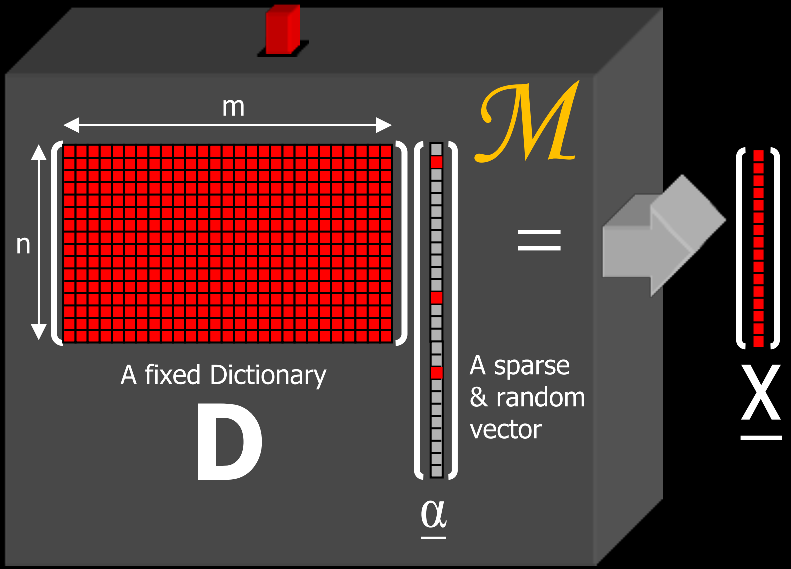
$\underline{\alpha}$

These steps define how $\underline{\alpha}$ could be created, and this leads to a complete definition of $P(\underline{x})$



Sparseland is an Interesting Model

- Simple: Every generated signal is built as a linear combination of few atoms from the dictionary **D**
- Rich: A general model in which the obtained signals are a union of many low-dimensional Gaussians
- Familiar: We have been using this model and variations thereof for a while, and now it is time to make it more precise



Relation to Transform-Based Priors

- Assume that **D** is square and invertible

$$P(\underline{x}) \sim e^{-\lambda\|\underline{\alpha}\|_0} \quad \text{where} \quad \underline{x} = \mathbf{D}\underline{\alpha}$$

$$\Downarrow$$

$$P(\underline{x}) \sim e^{-\lambda\left\|\mathbf{D}^{-1}\underline{x}\right\|_0} = e^{-\lambda\|\mathbf{T}\underline{x}\|_0}$$

- The *Sparseland* model generalizes previous transform-based methods by

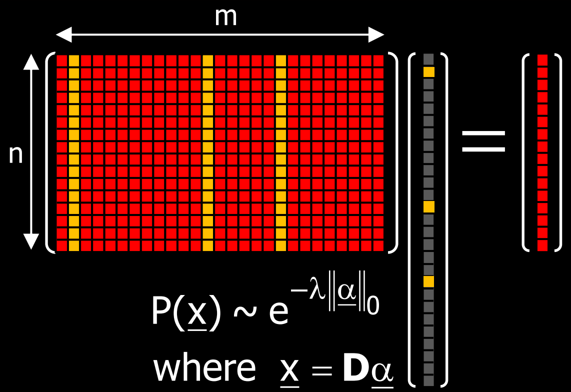
  (1) adopting over-completeness, &

  (2) daring to work directly with L_0

Union of Subspaces (UoS)

TECHNION
Israel Institute of Technology

- Consider all the signals $\underline{x}$ that emerge from the same k atoms in **D** – all of these reside in the same subspace, spanned by these atoms



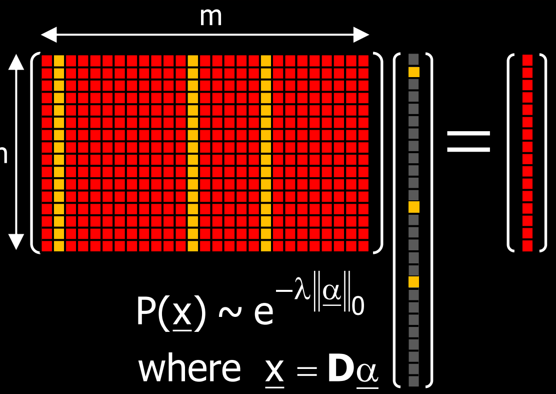
$$P(\underline{x}) \sim e^{-\lambda\|\underline{\alpha}\|_0}$$

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

- Thus, every possible support (there are m-choose-k of them) represents one such subspace which the modelled signals could belong to
- *Sparseland*: A Union of Subspaces model

The Pursuit Task

- Given an ε-noisy signal, we need to search the "closest subspace" and to project onto it
- This is the same as saying that we search the best-matching support
- This is hard due to the number of subspaces
- Pursuit = Projection onto our model



$$P(\underline{x}) \sim e^{-\lambda\|\underline{\alpha}\|_0}$$

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

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

Sparseland vs. GMM

- A closely related model: Gaussian-Mixture-Model

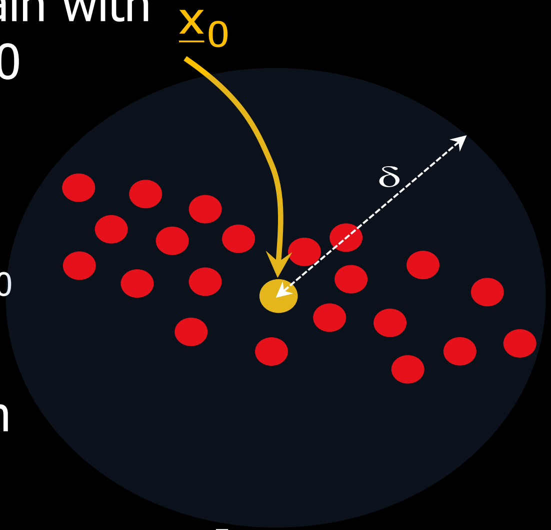
$$P(\underline{x}) \sim \sum_{k=1}^{N} \pi_k \exp\left\{-\lambda_k \underline{x}^T \mathbf{Q}_k \underline{x}\right\}$$

- In this model, there are N (assumed here as zero-mean) Gaussians, each characterized by its auto-correlation matrix $\mathbf{Q}_k$
- Typically, $\mathbf{Q}_k$ are of low-rank, to represent the fact that the Gaussians are low-dimensional
- *Sparseland* offers an exponential number of Gaussians, each obtained from a different support
- All of these Gaussians are encapsulated by **D**

The Geometry behind *Sparseland*

Another Virtual Experiment

- Suppose we experiment again with image patches of size 20×20 and we have a database with many millions of them
- Choose an arbitrary patch $\underline{x}_0$
- Find the δ-neighbors of this patch (N of them), and form the following matrix

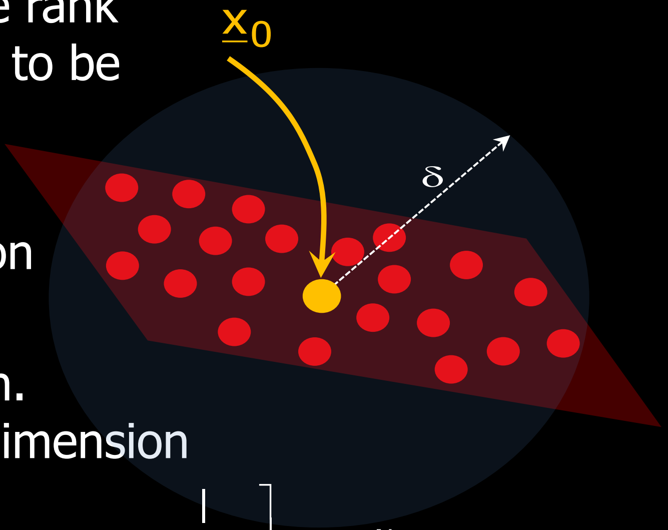


$$\mathbf{E} = \begin{bmatrix} | & | & | & & | \\ \underline{x}_1-\underline{x}_0 & \underline{x}_2-\underline{x}_0 & \underline{x}_3-\underline{x}_0 & \cdots & \underline{x}_N-\underline{x}_0 \\ | & | & | & & | \end{bmatrix} \in \mathbb{R}^{n\times N}$$

- Let's look more closely at the matrix **E** …

Another Virtual Experiment

- Observation: The effective rank of **E** (by SVD) is expected to be very low: rank(**E**)=d<<n
- This is universally true for most signals we operate on
- Why? because the local behavior is of a low-dimen. subspace, where d is its dimension

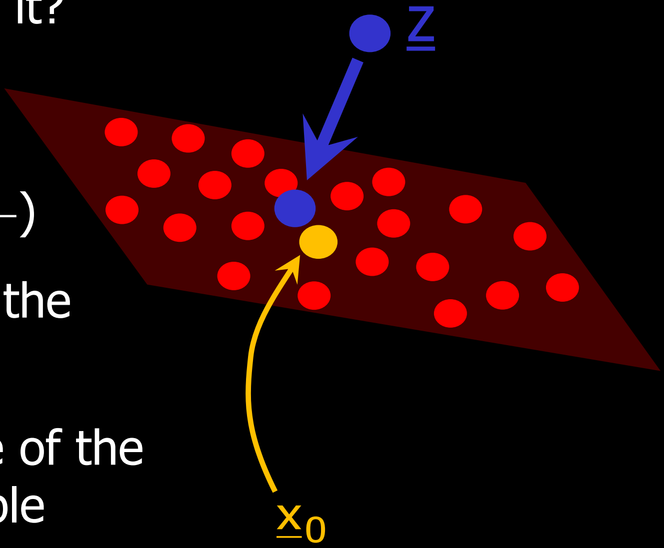


$$\mathbf{E} = \begin{bmatrix} | & | & | & & | \\ \underline{x}_1 - \underline{x}_0 & \underline{x}_2 - \underline{x}_0 & \underline{x}_3 - \underline{x}_0 & \cdots & \underline{x}_N - \underline{x}_0 \\ | & | & | & & | \end{bmatrix} \in \mathbb{R}^{n \times N}$$

- The orientation and dimension of this subspace may (and will) change from one point to another

Implications

- Given a noisy version of $\underline{x}_0$: $\underline{z}=\underline{x}_0+\underline{v}$ $[\underline{v}\sim\mathbb{N}(0,\sigma_2\mathbf{I})]$ how could we denoise it?
- By projecting to the subspace around $\underline{x}_0$ ($\rightarrow$ chicken and egg $\leftarrow$)
- How come $\underline{z}$ is not on the subspace itself?
  - The relative volume of the subspace is negligible
  - Recall that $E\{(\underline{z}-\underline{x}_0)^T\mathbf{E}\}=0$, and this implies that $\underline{z}-\underline{x}_0$ is very likely to be orthogonal to the above subspace



Implications

- Given a noisy version of $\underline{x}_0$: $\underline{z}=\underline{x}_0+\underline{v}$ $[\underline{v}\sim\mathbb{N}(0,\sigma_2\mathbf{I})]$ how shall we denoise it?
- Here are several options:
  - Non-Parametric: Nearest Neighbor (NN), or K-NN
  - Local-Parametric: Group neighbors, estimate the subspace and project
  - Parametric: Cluster the DB into K subgroups, and estimate a subspace per each. When a signal is to be denoised, assign it to the closest subgroup, and then project on the corresponding subspace (K=1: PCA)
  - *Sparseland:* one dictionary encapsulates many such clusters, and thus the pursuit applies this projection

Processing *Sparseland*'s Signals

So, Lets Work with *Sparseland*

TECHNION Israel Institute of Technology

- We have just seen how *Sparsland* generalizes some of the best-known models
- This new model offers a powerful union-of-subspaces to describe practically any source of data
- This parallels a specific and very rich Gaussian-Mixture-Model structure
- It is time to deploy it to actual signal processing tasks and the question is how should this be done

Signal Transform in *Sparseland*

- We are given a *Sparseland* signal $\underline{x}=\mathbf{D}\underline{\alpha}$ (where $\underline{\alpha}$ is very sparse) and need the most effective transform
- Effective? In what sense? We want the coefficients to
  - … expose interesting knowledge about the signal
  - … be independent of each other, so that operating on them separately is optimal
  - … concentrate the energy in as fewest elements
- How about this?

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

- The sparsest representation is the ideal transform, satisfying all the above, and we do have theoretical results guaranteeing finding it

Signal Denoising in *Sparseland*

- We are given $\underline{z}$, an ε-noisy version of a *Sparseland* signal $\underline{x}_0 = \mathbf{D}\underline{\alpha}_0$ and our goal is to clean it up
- Since $\underline{\alpha}_0$ is very sparse, this implies that $\underline{x}_0$ resides in a low-dim. subspace spanned by a small set of atoms from **D**
- How about this as a denoising procedure:

$$\hat{\underline{\alpha}} = \min_{\underline{\alpha}} \|\underline{\alpha}\|_0 \quad \text{s.t.} \quad \|\underline{z} - \mathbf{D}\underline{\alpha}\|_2 \leq \varepsilon \quad \Rightarrow \quad \hat{\underline{x}} = \mathbf{D}\hat{\underline{\alpha}}$$

- If $\hat{\underline{\alpha}}$ is close to $\underline{\alpha}_0$ (e.g., in support) this leads to a strong denoising effect
- Theoretical claims supporting this hope exist !!

Inverse Problems in *Sparseland*

- We are given $\underline{z}=\mathbf{H}\underline{x}_0+\underline{v}$, an ε-noisy corrupted measurement of a *Sparseland* signal $\underline{x}_0=\mathbf{D}\underline{\alpha}_0$ and our goal is to restore $\underline{x}_0$
- Our strategy – recover $\underline{\alpha}_0$ and then build our estimate:

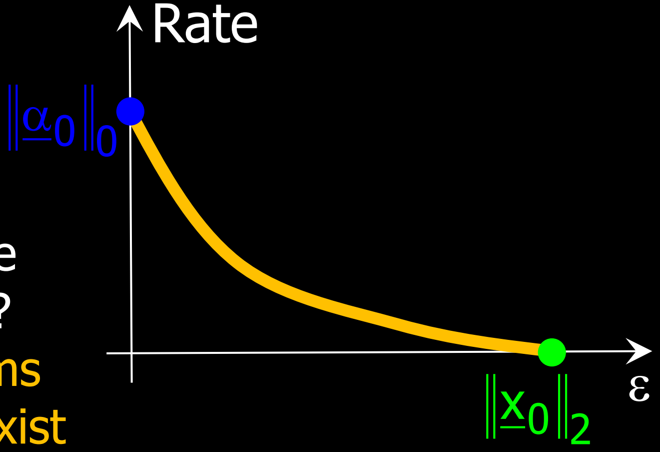
$$\hat{\underline{\alpha}} = \min_{\underline{\alpha}} \|\underline{\alpha}\|_0 \quad \text{s.t.} \quad \|\underline{z} - \mathbf{HD}\underline{\alpha}\|_2 \le \varepsilon \quad \Rightarrow \quad \hat{\underline{x}} = \mathbf{D}\hat{\underline{\alpha}}$$

- Here again we are equipped with theoretical guarantees that finding a solution close to $\underline{\alpha}_0$ is within reach, and practical algorithms to do this are available

Signal Compression in *Sparseland*

- We are given $\underline{x}_0$, a *Sparseland* signal $\underline{x}_0=\mathbf{D}\underline{\alpha}_0$ and our goal is to compress it
- Solving the following for varying values of ε could lead to an ideal Rate-Distortion curve

$$\min_{\underline{\alpha}} \|\underline{\alpha}\|_0 \quad \text{s.t.} \quad \|\underline{x}_0 - \mathbf{D}\underline{\alpha}\|_2 \le \varepsilon$$



- Could we really solve this set of problems? Yes! theoretical claims supporting this do exist

Signal Separation in *Sparseland*

- We are given $\underline{z}=\underline{x}_1+\underline{x}_2+\underline{v}$, an ε-noisy mixture of two *Sparseland* signals $\underline{x}_1=\mathbf{D}_1\underline{\alpha}_1$ and $\underline{x}_2=\mathbf{D}_2\underline{\alpha}_2$ and our goal is to break z into its ingredients

- Our strategy – recover $\underline{\alpha}_1$ and $\underline{\alpha}_2$ by:

$$\hat{\underline{\alpha}}_1,\hat{\underline{\alpha}}_2 = \min_{\underline{\alpha}_1,\underline{\alpha}_2} \|\underline{\alpha}_1\|_0 + \|\underline{\alpha}_2\|_0 \quad \text{s.t.} \quad \|\underline{z} - \mathbf{D}_1\underline{\alpha}_1 - \mathbf{D}_2\underline{\alpha}_2\|_2 \le \varepsilon \quad \Rightarrow \quad \hat{\underline{x}}_1 = \mathbf{D}_1\hat{\underline{\alpha}}_1, \quad \hat{\underline{x}}_2 = \mathbf{D}_2\hat{\underline{\alpha}}_2$$

- The above can be re-written as

$$\hat{\underline{\alpha}}_T = \min_{\underline{\alpha}_T} \|\underline{\alpha}_T\|_0 \quad \text{s.t.} \quad \|\underline{z} - \mathbf{D}_T\underline{\alpha}_T\|_2 \le \varepsilon$$

where $\underline{\alpha}_T = \begin{bmatrix} \underline{\alpha}_1 \\ \underline{\alpha}_2 \end{bmatrix}, \ \mathbf{D}_T = \begin{bmatrix} \mathbf{D}_1 & \mathbf{D}_2 \end{bmatrix}$

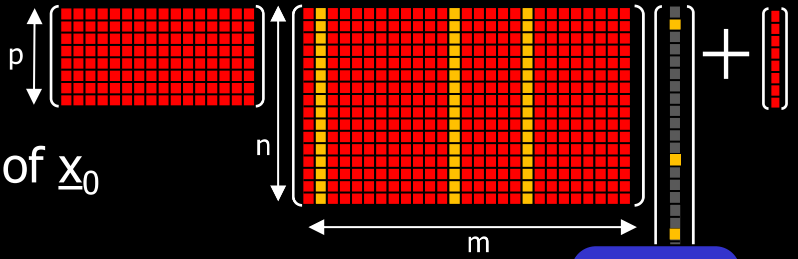
Compressed-Sensing in *Sparseland*

TECHNION
Israel Institute of Technology

- Suppose that $\underline{x}_0=\mathbf{D}\underline{\alpha}_0$ is a *Sparseland* signal of length n that we aim to measure

- Instead, we get an ε-noisy projected version of it, $\underline{z}=\mathbf{P}\underline{x}_0+\underline{v}$. **P** is a well-chosen measurement operator

- Given $\underline{z}$, our goal is the recovery of $\underline{x}_0$



$$\hat{\underline{\alpha}} = \min_{\underline{\alpha}} \|\underline{\alpha}\|_0 \quad \text{s.t.} \quad \|\underline{z} - \mathbf{PD}\underline{\alpha}\|_2 \le \varepsilon \quad \Rightarrow \quad \hat{\underline{x}} = \mathbf{D}\hat{\underline{\alpha}}$$

- This resembles the inverse problems mentioned above with one major difference: We can design **P**

Processing *Sparseland* Signals

All these (and many other) processing tasks boil down to the solution of

$$(P_0^{\varepsilon}) \quad \underline{\hat{\alpha}} = \min_{\underline{\alpha}} \left\|\underline{\alpha}\right\|_0 \quad \text{s.t.} \quad \left\|\underline{z} - \mathbf{D}\underline{\alpha}\right\|_2 \le \varepsilon$$

for which we know that

1. It is theoretically sensible, and
2. There are numerical ways to handle it

Bottom line: *Sparseland* is rooted on well-established modeling ideas, and accompanied by solid mathematical foundations

A Word of Caution

- At this stage you might get the impression that bringing *Sparseland* to actual image processing tasks is very simple – All that is needed is to form and solve (P_0^ε)
- Reality is very different !
- As we will see, in the migration from theory to practice, there are many different ways to turn *Sparseland* to actual algorithms
- This leaves much room for ingenuity, originality, flexibility and creativity, in designing novel image processing algorithms

Sparse & Redundant Representations and Their Applications in Signal and Image Processing

Iterative Shrinkage and Image Deblurring

TECHNION
Israel Institute of Technology

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

Image-Deblurring via *Sparseland*: Problem Formulation

The Deblurring Experiment

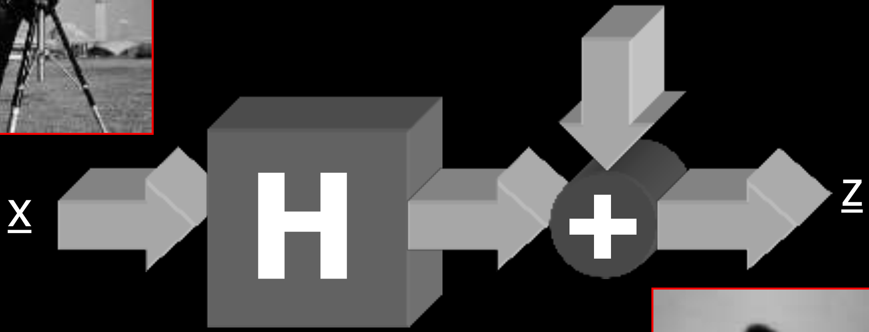
TECHNION
Israel Institute of Technology

- We have just been convinced about the importance and relevance of *Sparseland* to actual image processing needs
- We are eager to demonstrate this to a specific task: We choose to address image deblurring
- Our task: Recover an image $\underline{x}$ from its blurry & noisy version $\underline{z}=\mathbf{H}\underline{x}+\underline{v}$, where $\underline{v}\sim\mathbb{N}(0,\sigma^2\mathbf{I})$ & $\mathbf{H}$ is assumed known
- Recall: we said that this would be done by

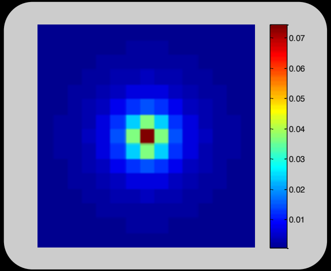
$$\hat{\underline{\alpha}} = \min_{\underline{\alpha}} \|\underline{\alpha}\|_0 \quad \text{s.t.} \quad \|\underline{z}-\mathbf{HD}\underline{\alpha}\|_2 \le \varepsilon \;\Rightarrow\; \hat{\underline{x}} = \mathbf{D}\hat{\underline{\alpha}}$$

More Specifically

$\underline{v}$: White Gaussian noise $\sigma^2=2$



H: 15×15 kernel



$$\frac{c}{n^2+m^2+1}$$
for
$$-7 \le n,m \le 7$$

The Restoration Algorithm

$$\underline{\hat{\alpha}} = \min_{\underline{\alpha}} \|\underline{\alpha}\|_0 \quad \text{s.t.} \quad \|\underline{z} - \mathbf{HD}\underline{\alpha}\|_2 \le \varepsilon \;\Rightarrow\; \underline{\hat{x}} = \mathbf{D}\underline{\hat{\alpha}}$$

We turn to the Lagrangian form of this optimization, so as to manage the constraint more conveniently

$$\underline{\hat{\alpha}} = \min_{\underline{\alpha}} \lambda \|\underline{\alpha}\|_0 + \frac{1}{2}\|\underline{z} - \mathbf{HD}\underline{\alpha}\|_2^2$$

and this implies that we will have a parameter λ to tune

The Restoration Algorithm

$$\underline{\hat{\alpha}} = \min_{\underline{\alpha}} \lambda \left\| \underline{\alpha} \right\|_0 + \frac{1}{2} \left\| \underline{z} - \mathbf{HD}\underline{\alpha} \right\|_2^2$$

We relax the L_0 and replace it with an L_1

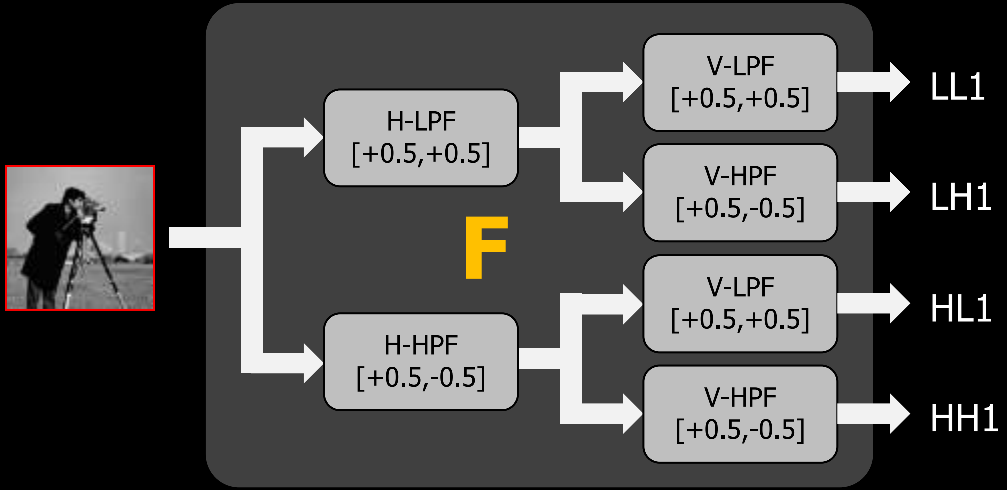
$$\underline{\hat{\alpha}} = \min_{\underline{\alpha}} \lambda \left\| \underline{\alpha} \right\|_1 + \frac{1}{2} \left\| \underline{z} - \mathbf{HD}\underline{\alpha} \right\|_2^2$$

Main Questions to Address:

- Who is **D** ? We'll answer this immediately
- How shall we minimize this function ? We'll address this next

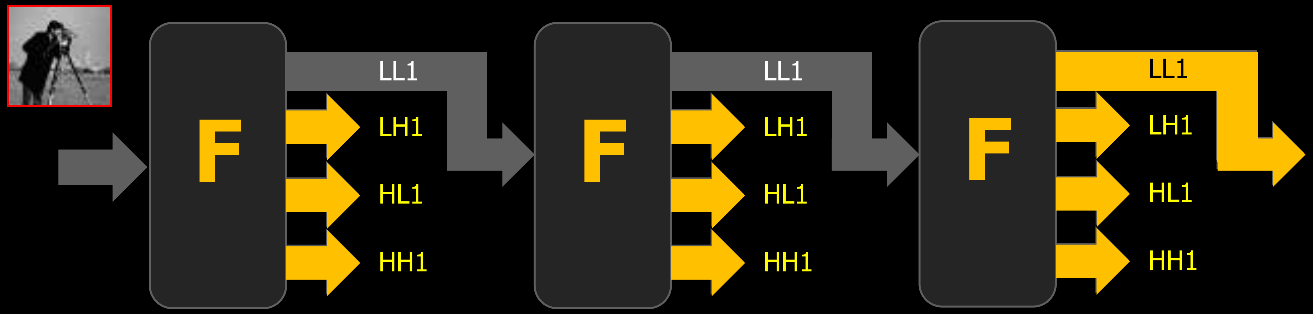
The Dictionary D

- We choose to use the un-decimated Haar Wavelet as the dictionary
- It is best described by the operation $\mathbf{D}^T\underline{x}$
  - Part 1: We apply this pair of separable filters (low-pass and high-pass)



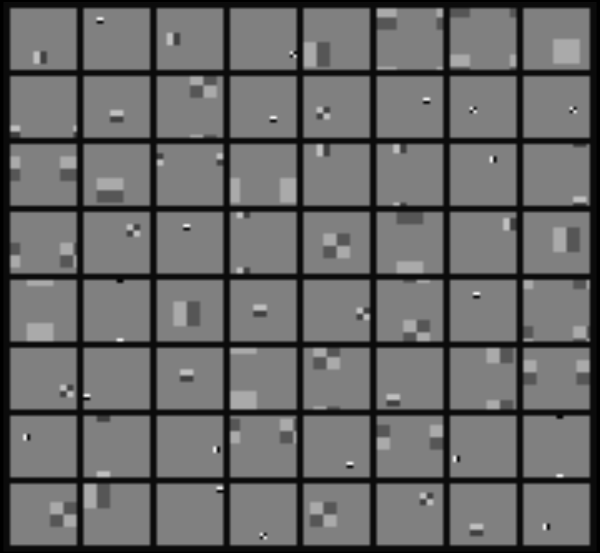
The Dictionary **D**

- We choose to use the un-decimated Haar Wavelet as the dictionary
- It is best described by the operation $\mathbf{D}^T\underline{x}$
  - Part 1: We apply this pair of separable filters (low-pass and high-pass)
  - Part 2: We repeat this filtering in 3 layers, getting a redundancy of 10:1 in **D**



The Dictionary **D**: The Atoms

- Here are a few atoms from **D**, demonstrated for an image of size 20×20 pixels
- Observe that there are three scales in these atoms
- The atoms' content: horizontal vertical and diagonal edges or a constant value
- Note: these atoms ARE NOT normalized

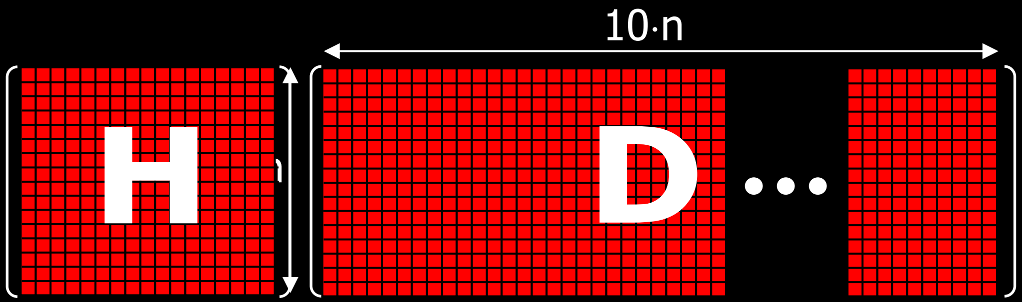


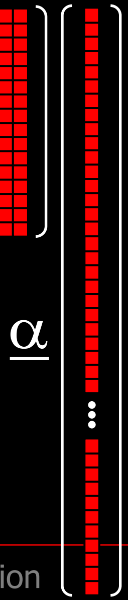
Starting with Classical Optimization

Our Optimization Task

$$\underline{\hat{\alpha}} = \min_{\underline{\alpha}} \lambda \left\|\underline{\alpha}\right\|_1 + \frac{1}{2}\left\|\underline{z} - \mathbf{HD}\underline{\alpha}\right\|_2^2$$

Let's talk about the dimensions involved





- We will process an image of size $n=65536=256^2$ pixels
- The Haar dictionary is 10-times redundant
- Thus, the unknown ($\underline{\alpha}$) length is 655,360

So, How do we Optimize ?

TECHNION
Israel Institute
of Technology

$$\underline{\hat{\alpha}} = \min_{\underline{\alpha}} \lambda \left\|\underline{\alpha}\right\|_1 + \frac{1}{2}\left\|\underline{z} - \mathbf{HD}\underline{\alpha}\right\|_2^2$$

The first thought that comes to mind:
With all the vast knowledge in optimization,
we could easily find a proper tool

Optional Algorithms

$$\hat{\underline{\alpha}} = \min_{\underline{\alpha}} \lambda \left\|\underline{\alpha}\right\|_1 + \frac{1}{2}\left\|\underline{z} - \mathbf{HD}\underline{\alpha}\right\|_2^2$$

- Methods to consider:
  - Steepest Descent (SD)
  - Conjugate Gradient
  - Pre-Conditioned SD
  - Truncated Newton
  - Interior-Point Algorithms
  - …

Let’s Focus on the SD

$$\underline{\hat{\alpha}} = \min_{\underline{\alpha}} \underbrace{\lambda \|\underline{\alpha}\|_1 + \frac{1}{2}\|\underline{z} - \mathbf{HD}\underline{\alpha}\|_2^2}_{f(\underline{\alpha})}$$

$$\nabla f(\underline{\alpha}) = \lambda \cdot \text{sign}(\underline{\alpha}) + \mathbf{D}^T\mathbf{H}^T(\mathbf{HD}\underline{\alpha} - \underline{z})$$

$$\begin{aligned}\underline{\alpha}_{k+1} &= \underline{\alpha}_k - \mu \cdot \nabla f(\underline{\alpha}_k) \\ &= \underline{\alpha}_k - \mu\lambda \cdot \text{sign}(\underline{\alpha}_k) - \mu \cdot \mathbf{D}^T\mathbf{H}^T(\mathbf{HD}\underline{\alpha}_k - \underline{z})\end{aligned}$$

- μ depends on the Hessian’s eigenvalues:

$$0 < \mu < \frac{2}{\lambda_{max}\left\{\nabla^2 f(\underline{\alpha})\right\}} \approx \frac{2}{\lambda_{max}\left\{\mathbf{D}^T\mathbf{H}^T\mathbf{HD}\right\}}$$

(assuming that λ is very small)

Momentum Acceleration

- The SD algorithm is known for its zigzag path of solution (especially so when μ is optimized)

- A possible remedy: Momentum Acceleration

k

$$\underline{\alpha}_{k+1} = \underline{\alpha}_k + \underline{e}_k$$

k

$$\underline{d}_k = \underline{\alpha}_k + \underline{e}_k$$

$$\underline{\alpha}_{k+1} = \underline{d}_k + m \cdot \left(\underline{d}_k - \underline{d}_{k-1}\right)$$

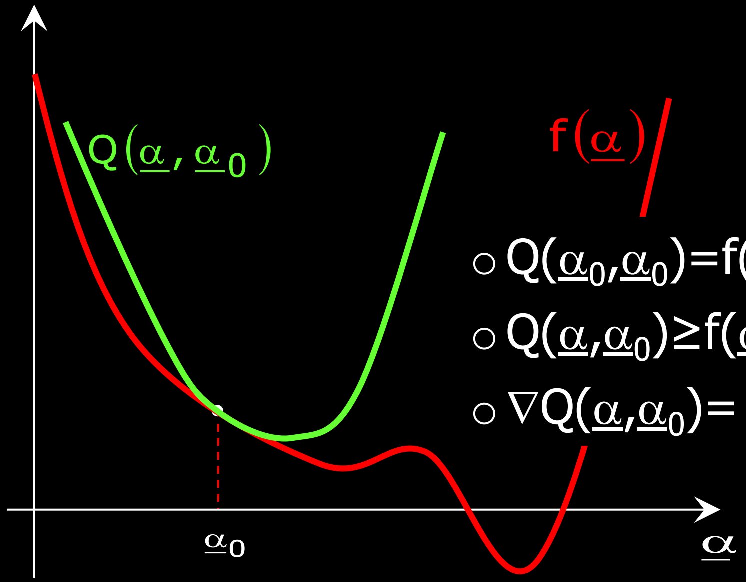
- The parameter m can be optimized for best performance (typically m≈0.9-1)

- This method has close ties with the Conjugate Gradient (CG) method

Iterative Shrinkage -Thresholding Algorithm (ISTA)

The Majorization-Minimization Idea

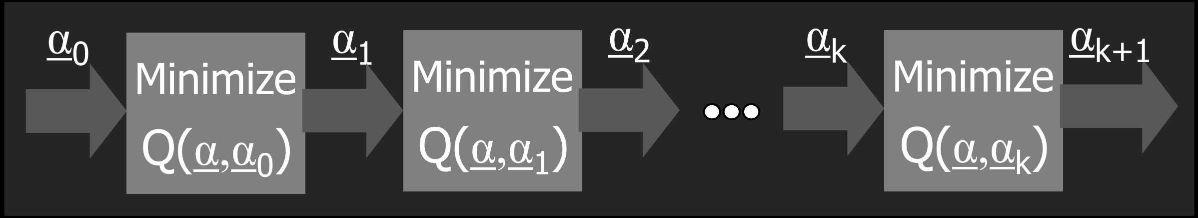
- Aim: minimize $f(\underline{\alpha})$ – Suppose it is too hard
- Define a function $Q(\underline{\alpha},\underline{\alpha}_0)$ that satisfies:



- $Q(\underline{\alpha}_0,\underline{\alpha}_0)=f(\underline{\alpha}_0)$
- $Q(\underline{\alpha},\underline{\alpha}_0)\geq f(\underline{\alpha})$ for all $\underline{\alpha}$ &
- $\nabla Q(\underline{\alpha},\underline{\alpha}_0)= \nabla f(\underline{\alpha})$ at $\underline{\alpha}_0$

The Majorization-Minimization Idea

- Then, the following algorithm necessarily converges to a local (global if $f(\underline{\alpha})$ is convex) minima of $f(\underline{\alpha})$ [Hunter & Lange (04)]



- We have replaced one optimization task by a series of them. This makes sense only if the minimization of $Q(\underline{\alpha},\underline{\alpha}_0)$ is much easier
- This implies that we need to build $Q(\underline{\alpha},\underline{\alpha}_0)$ wisely. How?

Constructing $Q(\underline{\alpha},\underline{\alpha}_0)$ for our Case

$$\underline{\hat{\alpha}} = \min_{\underline{\alpha}} \underbrace{\lambda \left\|\underline{\alpha}\right\|_1 + \frac{1}{2}\left\|\underline{z} - \mathbf{HD}\underline{\alpha}\right\|_2^2}_{f(\underline{\alpha})}$$

$$Q\left(\underline{\alpha},\underline{\alpha}_0\right) = f\left(\underline{\alpha}\right) + \frac{c}{2}\left\|\underline{\alpha} - \underline{\alpha}_0\right\|_2^2 - \frac{1}{2}\left\|\mathbf{HD}\left(\underline{\alpha} - \underline{\alpha}_0\right)\right\|_2^2$$

Let's check:

- $Q(\underline{\alpha}_0,\underline{\alpha}_0)=f(\underline{\alpha}_0)$? Definitely
- $Q(\underline{\alpha},\underline{\alpha}_0)\geq f(\underline{\alpha})$ for all $\underline{\alpha}$? Yes, as long as

  $$c\mathbf{I} - \left(\mathbf{HD}\right)^T\left(\mathbf{HD}\right) \succ 0 \Longrightarrow c > \lambda_{max}\left\{\left(\mathbf{HD}\right)^T\left(\mathbf{HD}\right)\right\}$$

- $\nabla Q(\underline{\alpha},\underline{\alpha}_0)= \nabla f(\underline{\alpha})$ at $\underline{\alpha}_0$? Yes, since the addition is quadratic with a minimum at $\underline{\alpha}=\underline{\alpha}_0$

Is $Q(\underline{\alpha},\underline{\alpha}_0)$ Easy to Minimize ?

$$Q(\underline{\alpha},\underline{\alpha}_0) = \lambda\|\underline{\alpha}\|_1 + \frac{1}{2}\|\underline{z} - \mathbf{HD}\underline{\alpha}\|_2^2$$

$$+\frac{c}{2}\|\underline{\alpha} - \underline{\alpha}_0\|_2^2 - \frac{1}{2}\|\mathbf{HD}(\underline{\alpha} - \underline{\alpha}_0)\|_2^2$$

Little bit of algebra (please check), and the above can be shown to be equal to

$$Q(\underline{\alpha},\underline{\alpha}_0) = \lambda\|\underline{\alpha}\|_1 +$$

$$+\frac{c}{2}\left\|\underline{\alpha} - \underbrace{\left\{\underline{\alpha}_0 + \frac{1}{c}(\mathbf{HD})^T(\underline{z} - \mathbf{HD}\underline{\alpha}_0)\right\}}_{\text{This expression can be computed – let's denote it as } \underline{v}_0}\right\|_2^2 + \text{Const.}$$

Is $Q(\underline{\alpha},\underline{\alpha}_0)$ Easy to Minimize ?

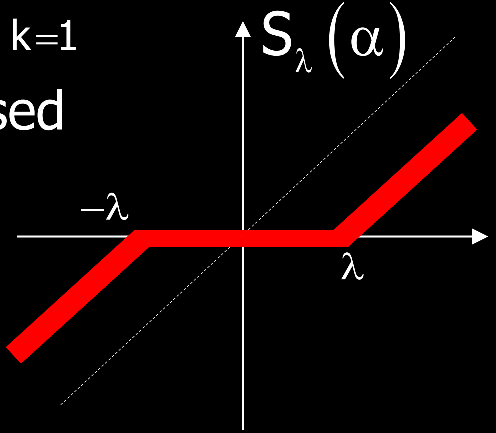
$$\min_{\underline{\alpha}} Q(\underline{\alpha},\underline{\alpha}_0) = \min_{\underline{\alpha}} \left\{ \lambda \|\underline{\alpha}\|_1 + \frac{c}{2}\|\underline{\alpha} - \underline{v}_0\|_2^2 + \text{Const.} \right\}$$

- This minimization is easy. It can be broken into m scalar tasks of the form (assume c=1)

$$\left\{ \min_{\alpha_k} \lambda|\alpha_k| + \frac{1}{2}\|\alpha_k - \beta_k\|_2^2 \right\}_{k=1}^{m}$$

- These problems have a closed form solution known as soft-thresholding

$$S_\lambda(\beta_k) = \begin{cases} 0 & |\beta_k| \le \lambda \\ \beta_k - \lambda \operatorname{sign}(\beta_k) & |\beta_k| > \lambda \end{cases}$$



Is Q($\underline{\alpha}$,$\underline{\alpha}_0$) Easy to Minimize ?

$$\min_{\underline{\alpha}} Q\left(\underline{\alpha},\underline{\alpha}_0\right)=\min_{\underline{\alpha}}\left\{\lambda\left\|\underline{\alpha}\right\|_1+\frac{c}{2}\left\|\underline{\alpha}-\underline{v}_0\right\|_2^2+\text{Const.}\right\}$$

- Thus, the solution of the above problem is given by a simple soft-thresholding applied on the elements of $\underline{v}_0$

$$\hat{\underline{\alpha}}_{opt}=S_{\lambda/c}\left\{\underline{v}_0\right\}$$

- This is easy, and applying this sequentially is definitely an appealing algorithm
- A proof ? See a related video from Course 1
- A Demo of this closed form ? See next

Bottom Line: ISTA

- Our objective is $\hat{\underline{\alpha}} = \min_{\underline{\alpha}} \lambda \|\underline{\alpha}\|_1 + \frac{1}{2}\|\underline{z} - \mathbf{HD}\underline{\alpha}\|_2^2$
- We apply this algorithm:

$$\underline{\alpha}_0 \rightarrow \boxed{\text{Minimize } Q(\underline{\alpha},\underline{\alpha}_0)} \xrightarrow{\underline{\alpha}_1} \boxed{\text{Minimize } Q(\underline{\alpha},\underline{\alpha}_1)} \xrightarrow{\underline{\alpha}_2} \cdots \xrightarrow{\underline{\alpha}_k} \boxed{\text{Minimize } Q(\underline{\alpha},\underline{\alpha}_k)} \xrightarrow{\underline{\alpha}_{k+1}}$$

$$\Rightarrow \quad \underline{\alpha}_{k+1} = S_{\lambda/c}\left\{\underline{\alpha}_k + \frac{1}{c}(\mathbf{HD})^T\left(\underline{z} - \mathbf{HD}\underline{\alpha}_k\right)\right\}$$

- This is the Iterative Shrinkage-Thresholding Algorithm (ISTA) [Figueiredo & Nowak, ‘03] [Daubechies, Defrise, De-Mol, ‘05] and it is guaranteed to get the global minimizer

Fast ISTA (FISTA)

The General Idea

k

$$\underline{\alpha}_{k+1} = g\left(\underline{\alpha}_k, \underline{e}_k\right)$$

k

$$\underline{d}_k = g\left(\underline{\alpha}_k, \underline{e}_k\right)$$

$$\underline{\alpha}_{k+1} = \underline{d}_k + m \cdot \left(\underline{d}_k - \underline{d}_{k-1}\right)$$

and in our case:

$$\underline{d}_k = S_{\lambda/c}\left\{\underline{\alpha}_k + \frac{1}{c}(\mathbf{HD})^T\left(\underline{z} - \mathbf{HD}\underline{\alpha}_k\right)\right\}$$

$$\underline{\alpha}_{k+1} = \underline{d}_k + m \cdot \left(\underline{d}_k - \underline{d}_{k-1}\right)$$

This is known as FISTA and it is proven to converge to the optimal solution [Beck & Teboul, '09]

ISTA – Summary

- We derived ISTA based on the Majorization-Minimization (MM) approach
- An alternative derivation relies on Proximal Regularization, a central concept in optimization theory
- Different methods of the same flavor exist:
  - Split-Bergman
  - ADMM based (presented in the first course)
  - Parallel Coordinate Descend
  - IRLS-based ISTA
- All share the same idea, of applying shrinkage and simple multiplications by **HD** and $\mathbf{D}^T\mathbf{H}^T$

ISTA – A Possible Generalization

- We can repeat all the above analysis for

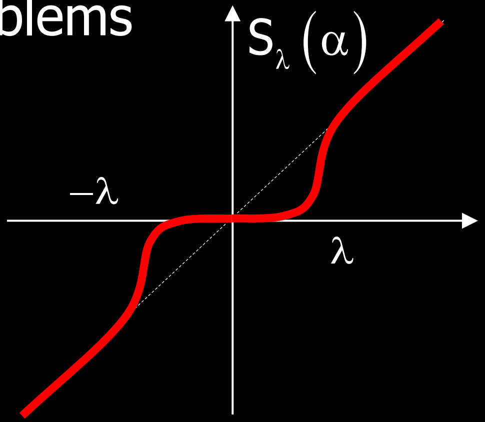
$$\hat{\underline{\alpha}} = \min_{\underline{\alpha}} \lambda\rho\left(\underline{\alpha}\right) + \frac{1}{2}\left\|\underline{z} - \mathbf{HD}\underline{\alpha}\right\|_2^2$$

where $\rho\left(\underline{\alpha}\right) = \sum_k \rho\left(\alpha_k\right)$ [ρ(α)=|α| for L_1]

- This leads to m scalar problems of the form

$$\min_{\alpha} \lambda\rho\left(\alpha\right) + \frac{1}{2}\left\|\alpha - \beta\right\|_2^2$$

- The solution is a ρ-depending shrinkage curve – see demo next



Shrinkage:
A Matlab Demo

Image Deblurring: Results & Discussion

Parameters

TECHNION
Israel Institute
of Technology

Found using the Power-Method

Found using the Power-Method

$$\underline{d}_k = \begin{cases} S_{\lambda/c}\left\{\underline{\alpha}_k + \frac{1}{c}(\mathbf{HD})^T\left(\underline{z} - \mathbf{HD}\underline{\alpha}_k\right)\right\} & \text{ISTA} \\ \underline{\alpha}_k + \mu\left[\lambda \mathrm{sign}\left(\underline{\alpha}\right) + (\mathbf{HD})^T\left(\underline{z} - \mathbf{HD}\underline{\alpha}_k\right)\right] & \text{SD} \end{cases}$$

$$\underline{\alpha}_{k+1} = \underline{d}_k + m \cdot \left(\underline{d}_k - \underline{d}_{k-1}\right)$$

m is either 0 (no acceleration) or m=0.9

λ is set experimentally to 0.06 for best performance

Evaluating c/μ by the Power-Method

- These two parameters are governed by

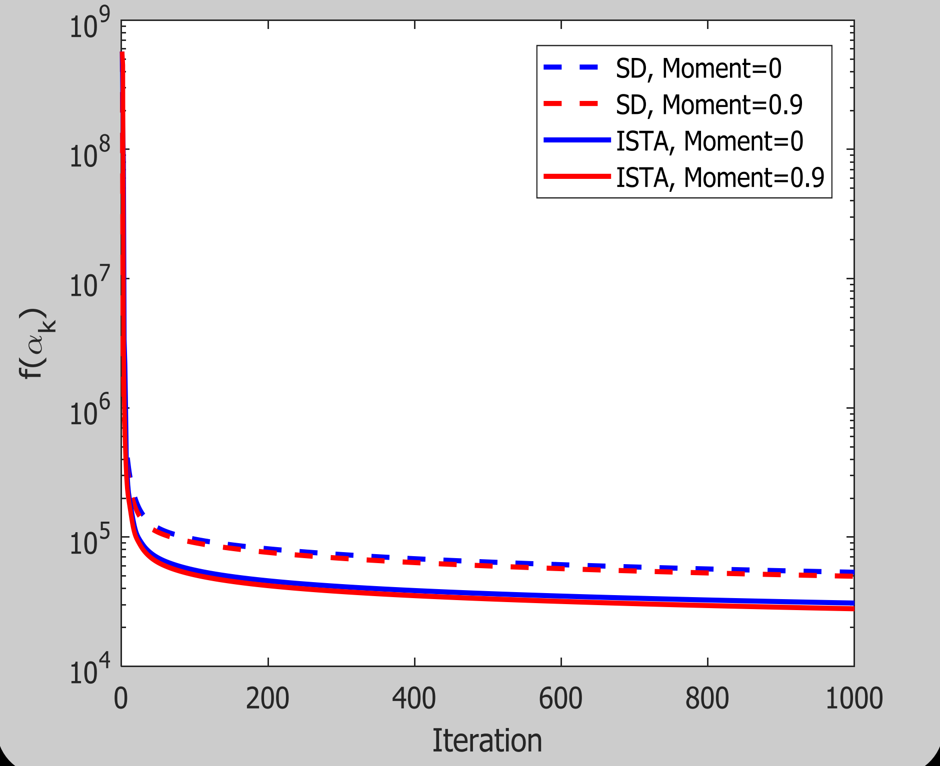
$$\lambda_{max}\left\{(\mathbf{HD})^T(\mathbf{HD})\right\}$$

$$c > \lambda_{max}\left\{(\mathbf{HD})^T(\mathbf{HD})\right\}$$

$$\mu \approx \frac{2}{\lambda_{max}\left\{\mathbf{D}^T\mathbf{H}^T\mathbf{HD}\right\}}$$

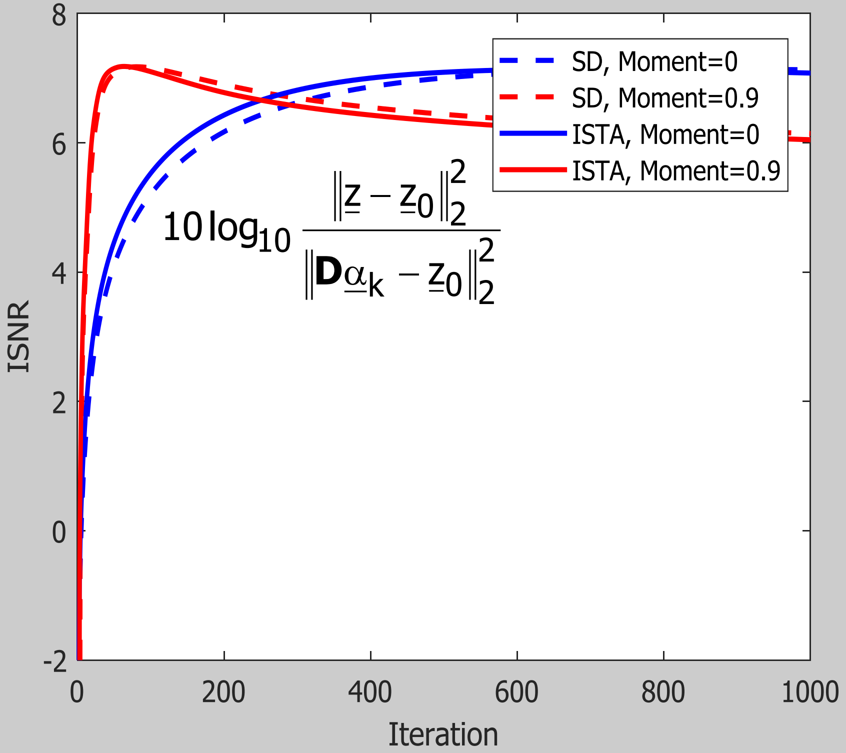
- We evaluate this value using the Power-Method:
  - Start with a random vector $\underline{v}_0$ of length m
  - Iterate k=0:1:N
    - Normalize $\underline{v}_k=\underline{v}_k/||\underline{v}_k||$
    - Compute $\underline{v}_k=\mathbf{D}^T\mathbf{H}^T(\mathbf{HD}\underline{v}_{k-1})$
  - The value $\underline{v}_k^T\underline{v}_{k-1}$ is the estimate for the maximal eigenvector λ_{max}

Results: $f(\underline{\alpha})$ $\left[\lambda\|\underline{\alpha}\|_1 + \frac{1}{2}\|\underline{z} - \mathbf{HD}\underline{\alpha}\|_2^2\right]$



- It appears that (F)ISTA is more effective in minimizing the function
- You might get the feeling that the algorithm has not yet converged – you are right

Results: ISNR



- $\underline{z}_0$ is the ideal image: Thus, the ISNR quantifies the improvement over assuming that $\underline{z}$ is our solution
- Both boosted methods lead to ISNR≈7dB after ~70 iterations, and then deteriorate
- With a smart stopping condition, (which exists!) we could catch this peak-performance and stop
- λ was tuned in this case to get the highest value at the peak

Results: The Restored Image

Iteration=[illegible] ISNR=[illegible]

Image Deblurring: A Closer Look at the Results

Results: The Good and the Bad

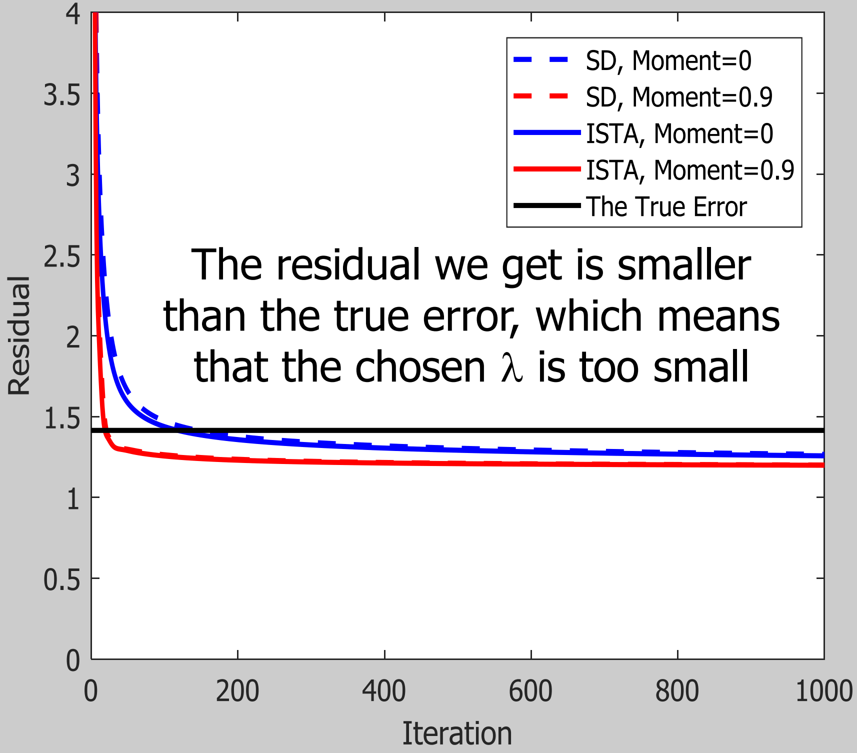
The results look great!
We get a strong deblurring effect
just as desired

However

This is not the result we expected !!

Let's explain why

Results: The Residual



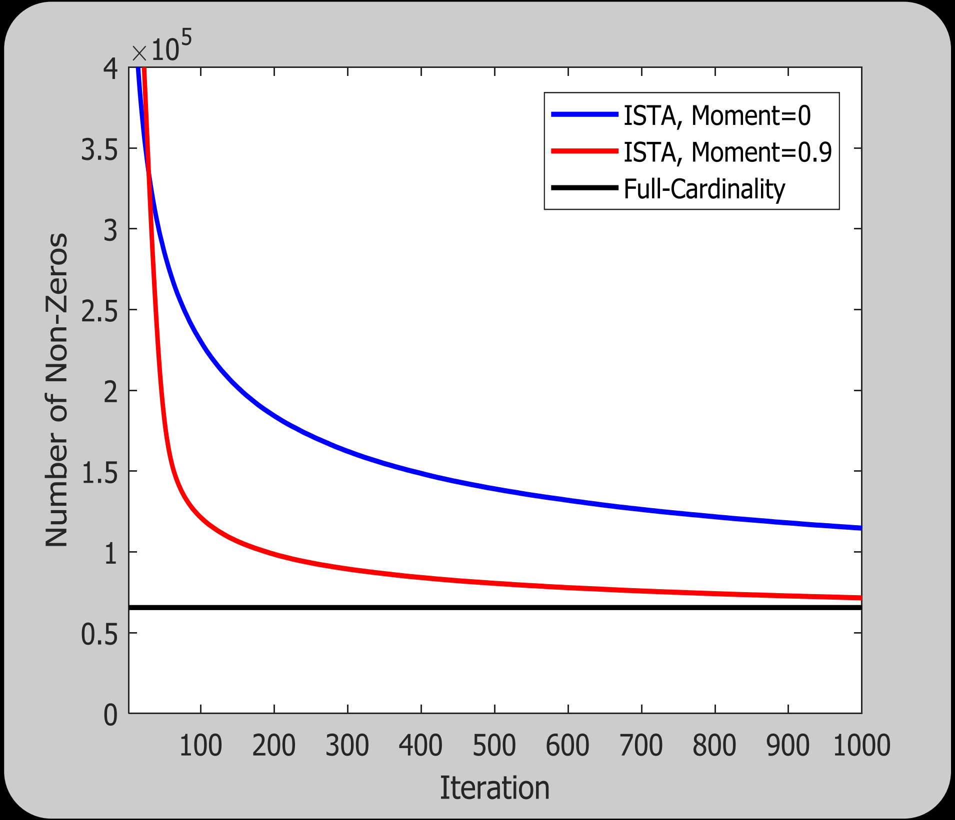
This is the function we are minimizing

$$\underline{\hat{\alpha}} = \min_{\underline{\alpha}} \lambda \|\underline{\alpha}\|_1 + \frac{1}{2}\|\underline{z} - \mathbf{HD}\underline{\alpha}\|_2^2$$

- If the residual does not match the noise energy (being smaller), we should choose a bigger λ
- This in turn means that we will lose on the high intermediate peak performance we saw

But this is not all …

Results: Sparsity ?



Here is the major difficulty:

- The solution we get is not sparse at all, and especially so around the first iterations where the peak was obtained (140,000NZ)
- Recall: the dimension of the signal is 256^2, so we expect the minimizer of our function to have 256^2 non-zeros at the most
- This comes back to the fact that the algorithm has not converged

Results: Sparsity ?

TECHNION Israel Institute of Technology

So, what shall we do?

- Run the FISTA for many more iterations in order to get the true optimal and sparse result, and then see what we get
- Do the above with a proper λ (0.17 was found to be suitable) so as to get the proper residual

Algorithm: FISTA

200,000 iterations, λ=0.17

Results: NNZ=18,460 (This is Sparse!)

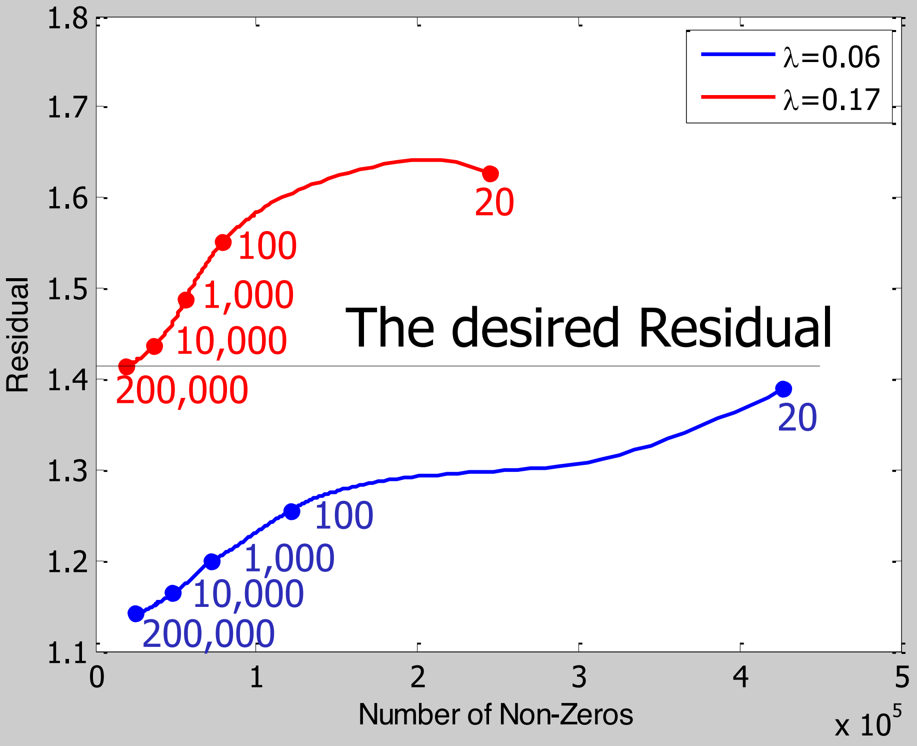
Residual=1.4144

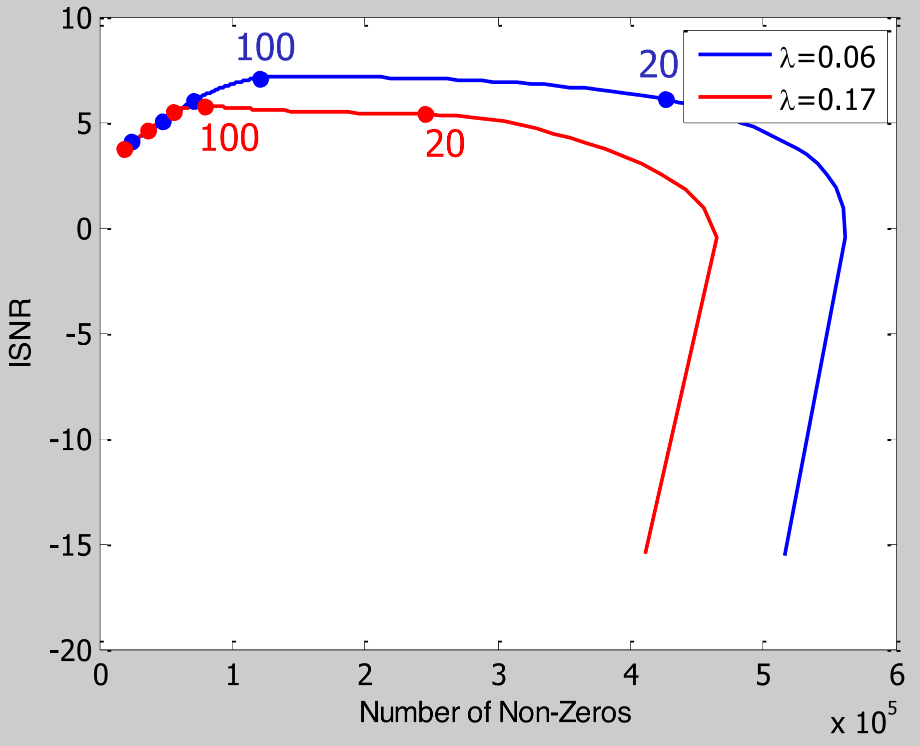
f(200,000)/f(1000)=0.985

ISNR=3.77dB **!!!**

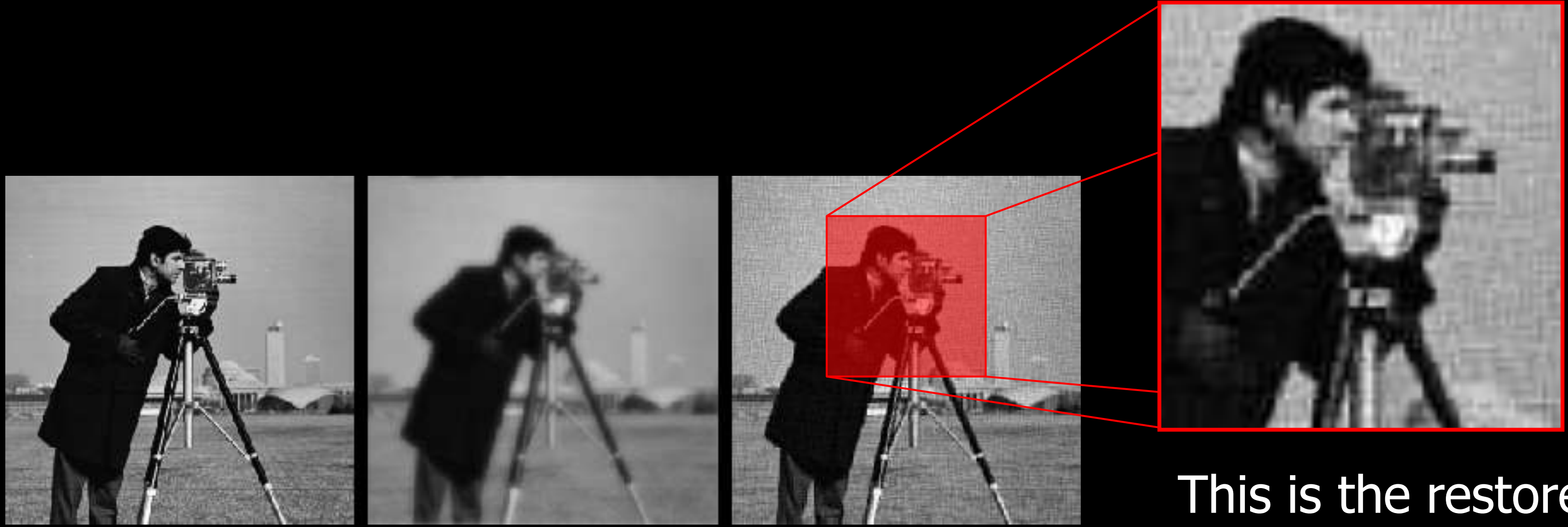
So, why have we gotten such a lovely deblurring with a dense solution ?

Results: Running till Convergence





Results: Running till Convergence



This is the restored image (3.77dB) – reasonably sharper but with some distortions

Explanations ?

Observation:

Sparsaland works

Harnessing *Sparseland* for image deblurring, we minimized

$$f\left(\underline{\alpha}\right)=\lambda\left\|\underline{\alpha}\right\|_1+\frac{1}{2}\left\|\underline{z}-\mathbf{HD}\underline{\alpha}\right\|_2^2$$

This led to a 3.77dB improvement over $\underline{z}$, & with a sparse representation (18,460 NZ)

However …

We observe a strange behavior

While minimizing this function, we encountered a MUCH better solution (7.18dB), obtained after only ~70 iterations, and having a very dense representation (140,000 NZ)

How Come ?

Answers: (1) MMSE Estimation
(2) Proper CSC Model
(3) Global vs. Local Modeling

Averaging!

So, What Next ?

We will certainly come back to the issue of getting a non-sparse solution, with an attempt to explain this phenomenon

But first, let’s discuss the choice of the dictionary, as this is a key step in deploying *Sparseland*