Sparse & Redundant Representations by Iterated-Shrinkage Algorithms

Michael Elad*

The Computer Science Department

The Technion – Israel Institute of technology

Haifa 32000, Israel

26 - 30 August 2007

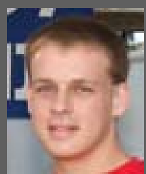
San Diego Convention Center

San Diego, CA USA

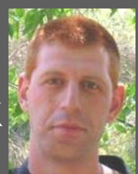
SPIE Optics+Photonics

Wavelet XII (6701)

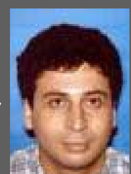
* Joint work with: Boaz Matalon



Joseph Shtok



Michael Zibulevsky



Today's Talk is About

the Minimization of the Function

$$f(\underline{\alpha}) = \frac{1}{2}\left\|\mathbf{D}\underline{\alpha} - \underline{x}\right\|_2^2 + \lambda\rho(\underline{\alpha})$$

by **Iterated-Shrinkage Algorithms**

Why This Topic?

- Key for turning theory to applications in *Sparse-land*,
- Shrinkage is intimately coupled with wavelet,
- The applications we target – fundamental signal processing ones.
- This is a hot topic in harmonic analysis.

Today we will discuss:

- **Why** this minimization task is important?
- **Which** applications could benefit from this minimization?
- **How** can it be minimized effectively?
- **What** iterated-shrinkage methods are out there? and

Agenda

1. **Motivating the Minimization of** $f(\underline{\alpha})$

   Describing various applications that need this minimization

2. **Some Motivating Facts**

   General purpose optin

3. **Iterated-Shrinkage**

   We describe five versi

4. **Some Results**

   Image deblurring results

5. **Conclusions**

$$\boxed{\hat{\underline{\alpha}} = \underset{\underline{\alpha}}{\text{ArgMin}} \ \frac{1}{2}\left\|\mathbf{D}\underline{\alpha} - \underline{x}\right\|_2^2 + \lambda\rho\left(\underline{\alpha}\right)}$$

Why?

Lets Start with Image Denoising

Remove Additive Noise ?

$$\underline{y} = \underline{x} + \underline{v}, \quad \underline{v} \sim N\left\{\underline{0}, \sigma^2 \mathbf{I}\right\}$$

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

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

Likelihood: Relation to measurements

Prior or regularization

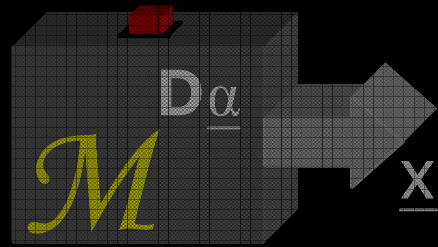
$\underline{y}$: Given measurements

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

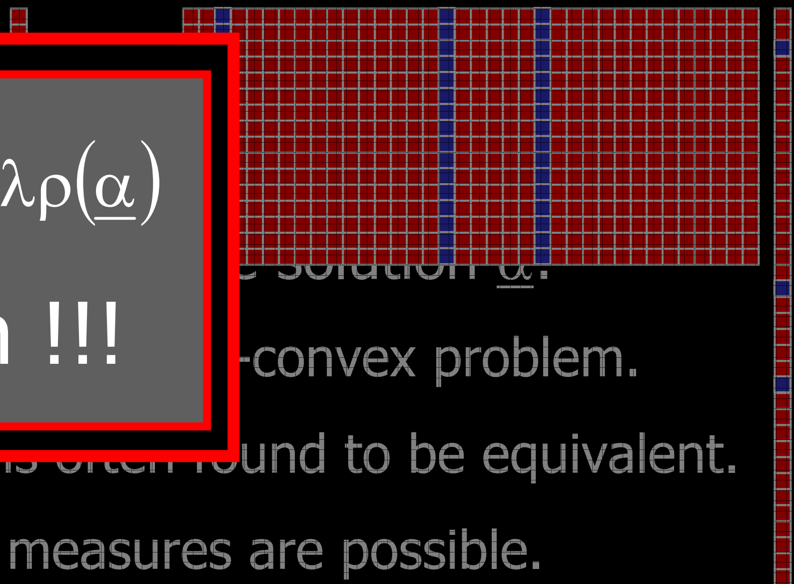
We will use a **Sparse & Redundant Representation** prior.

Our MAP Energy Function

- We assume that $\underline{x}$ is created by $\mathcal{M}$: where $\underline{\alpha}$ is a **sparse** & **redundant** representation and **D** is a known dictionary.



- This leads to: $\hat{\underline{\alpha}} = \underset{\underline{\alpha}}{\text{ArgMin}} \ \frac{1}{2}\left\| \underline{x} - \underline{y} \right\|_2^2$



$$\hat{\underline{\alpha}} = \underset{\underline{\alpha}}{\text{ArgMin}} \ \frac{1}{2}\left\| \mathbf{D}\underline{\alpha} - \underline{x} \right\|_2^2 + \lambda\rho(\underline{\alpha})$$

This is Our Problem !!!

- ... solution $\underline{\alpha}$:
  - ...-convex problem.
  - The L_p norm ($\|\underline{\alpha}\|_p$) with 0<p≤1 is often found to be equivalent.
  - Many other **ADDITIVE** sparsity measures are possible.

General (linear) Inverse Problems

- Assume that $\underline{x}$ is known to emerge from $\mathcal{M}$, as before.
- Suppose we observe $\underline{y} = \mathbf{H}\underline{x} + \underline{v}$, a "blurred" and noisy version of $\underline{x}$. How could we recover $\underline{x}$?
- A MAP estimator leads to:

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

$$\hat{\underline{\alpha}} = \underset{\underline{\alpha}}{\text{ArgMin}} \ \frac{1}{2}\left\| \mathbf{D}\underline{\alpha} - \underline{x} \right\|_2^2 + \lambda\rho(\underline{\alpha})$$

This is Our Problem !!!

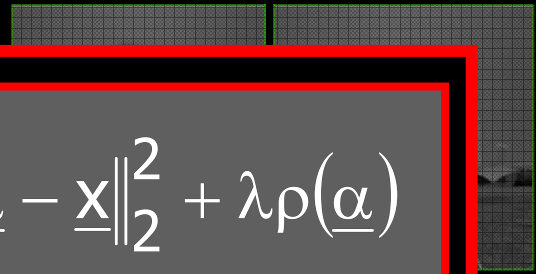
$\mathbf{D}\underline{\alpha}$

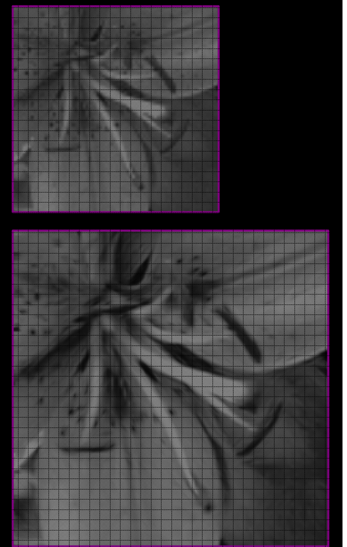
Noise

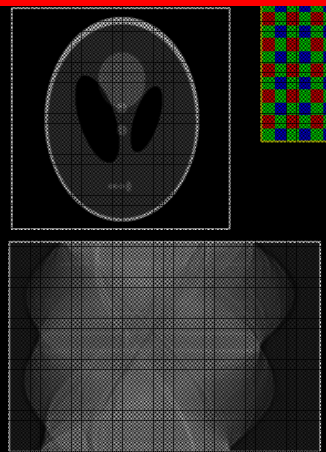
$\hat{\underline{x}}$

Inverse Problems of Interest

- De-Noising
- De-Blurring
- In-Painting
- De-Mosaicing
- Tomography
- Image Scale-Up & super-resolution
- And more …



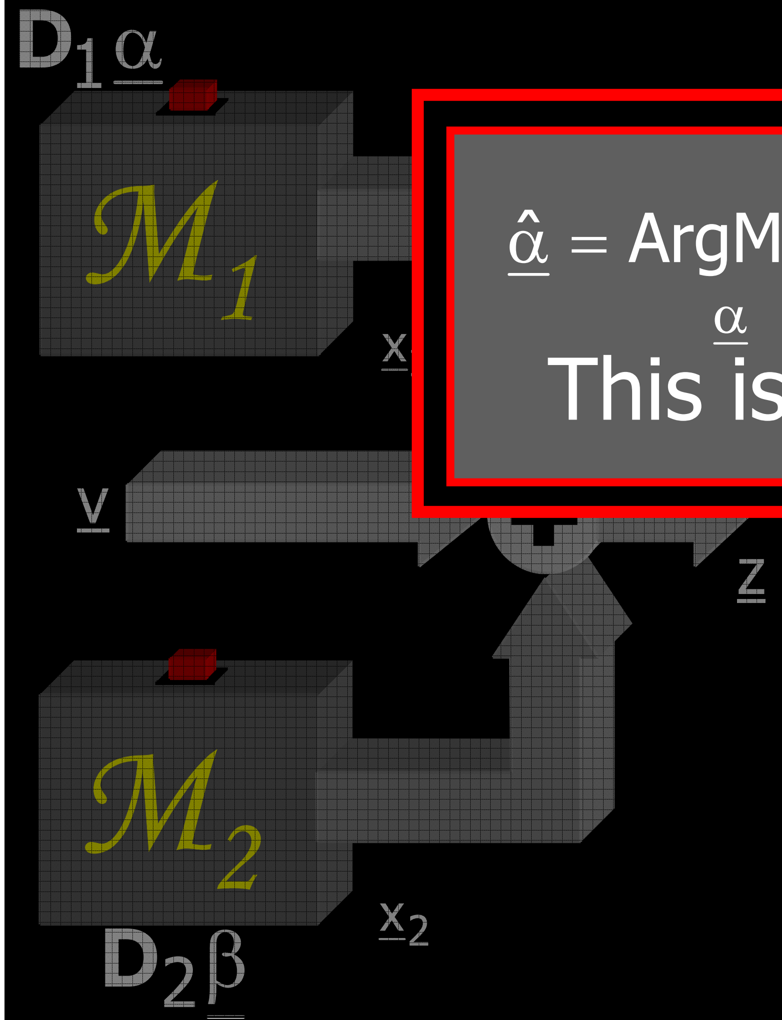




$$\underline{\hat{\alpha}} = \underset{\underline{\alpha}}{\mathrm{ArgMin}} \ \frac{1}{2}\left\|\mathbf{D}\underline{\alpha} - \underline{x}\right\|_2^2 + \lambda\rho\left(\underline{\alpha}\right)$$

This is Our Problem !!!

Signal Separation



- Given a mixture $\underline{z}=\underline{x}_1+\underline{x}_2+\underline{v}$ – two ... d white ... sire to separate

$$\ldots\ \mathbf{D}_2]\begin{bmatrix}\underline{\alpha}\\ \underline{\beta}\end{bmatrix}+\underline{v}$$

$$\hat{\underline{\alpha}} = \underset{\underline{\alpha}}{\text{ArgMin}}\ \frac{1}{2}\left\|\mathbf{D}\underline{\alpha}-\underline{x}\right\|_2^2+\lambda\rho(\underline{\alpha})$$

This is Our Problem !!!

- Thus, solving this problem using MAP leads to the Morphological Component Analysis (MCA) [Starck, Elad, Donoho, 2005]:

$$\begin{bmatrix}\hat{\underline{\alpha}}\\ \hat{\underline{\beta}}\end{bmatrix}=\underset{\underline{\alpha}}{\text{argmin}}\frac{1}{2}\left\|[\mathbf{D}_1,\mathbf{D}_2]\begin{bmatrix}\underline{\alpha}\\ \underline{\beta}\end{bmatrix}-\underline{z}\right\|_2^2+\lambda\rho\left(\begin{bmatrix}\underline{\alpha}\\ \underline{\beta}\end{bmatrix}\right)$$

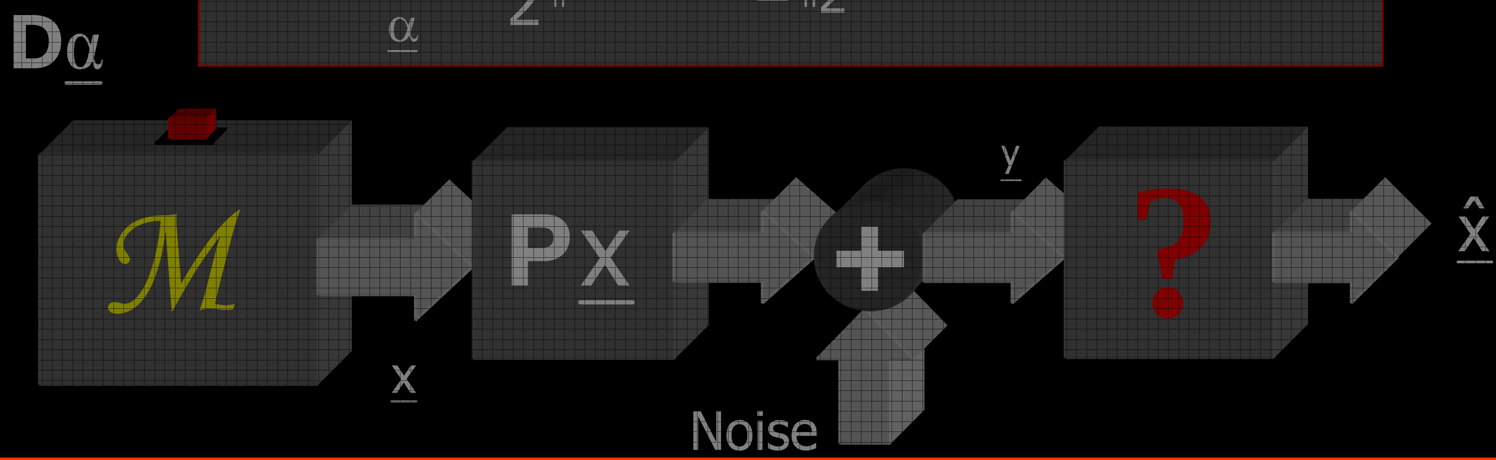
Compressed-Sensing [Candes et.al. 2006], [Donoho, 2006]

- In compressed-sensing we compress the signal $\underline{x}$ by exploiting its origin. This is do…
- The core idea: $\underline{y} \approx \mathbf{P}$… information about th…
- Reconstruction?

$$\underline{\hat{\alpha}} = \underset{\underline{\alpha}}{\text{ArgMin}} \; \frac{1}{2}\left\|\mathbf{D}\underline{\alpha} - \underline{x}\right\|_2^2 + \lambda\rho(\underline{\alpha})$$

This is Our Problem !!!

$$\underline{\hat{\alpha}} = \underset{\underline{\alpha}}{\operatorname{argmin}} \frac{1}{2}\left\|\mathbf{PD}\underline{\alpha} - \underline{y}\right\|_2^2 + \lambda\rho(\underline{\alpha}) \;\Rightarrow\; \underline{\hat{x}} = \mathbf{D}\underline{\hat{\alpha}}$$



Brief Summary #1

The minimization of the function

$$f(\underline{\alpha})=\frac{1}{2}\left\|\mathbf{D}\underline{\alpha}-\underline{x}\right\|_2^2+\lambda\rho(\underline{\alpha})$$

is a worthy task,
serving many & various applications.

So, How This Should be Done?

Agenda

1. Motivating the Minimization of $f(\alpha)$

   Describing various applications that need this minimization

2. **Some Motivating facts**

   General purpose optimization tools, and the unitary case

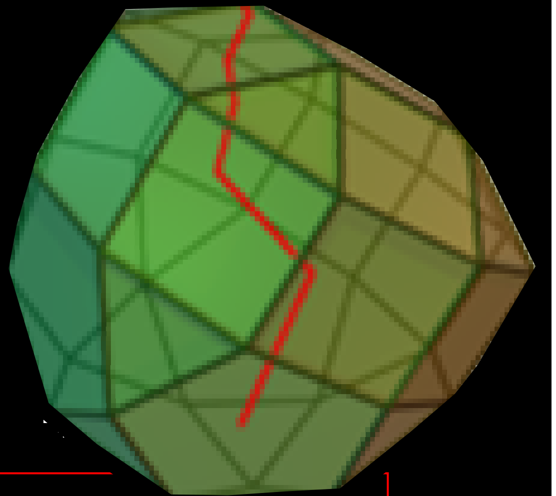
3. Iterated-Shrinkage Algorithms

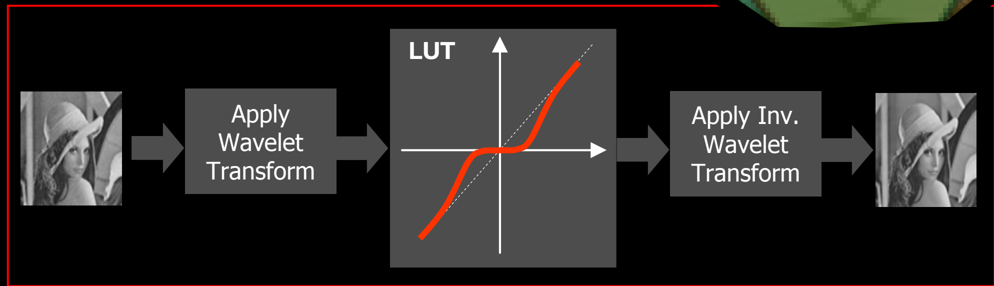
   We describe five versions of those in detail

4. Some Results

   Image deblurring results

5. Conclusions



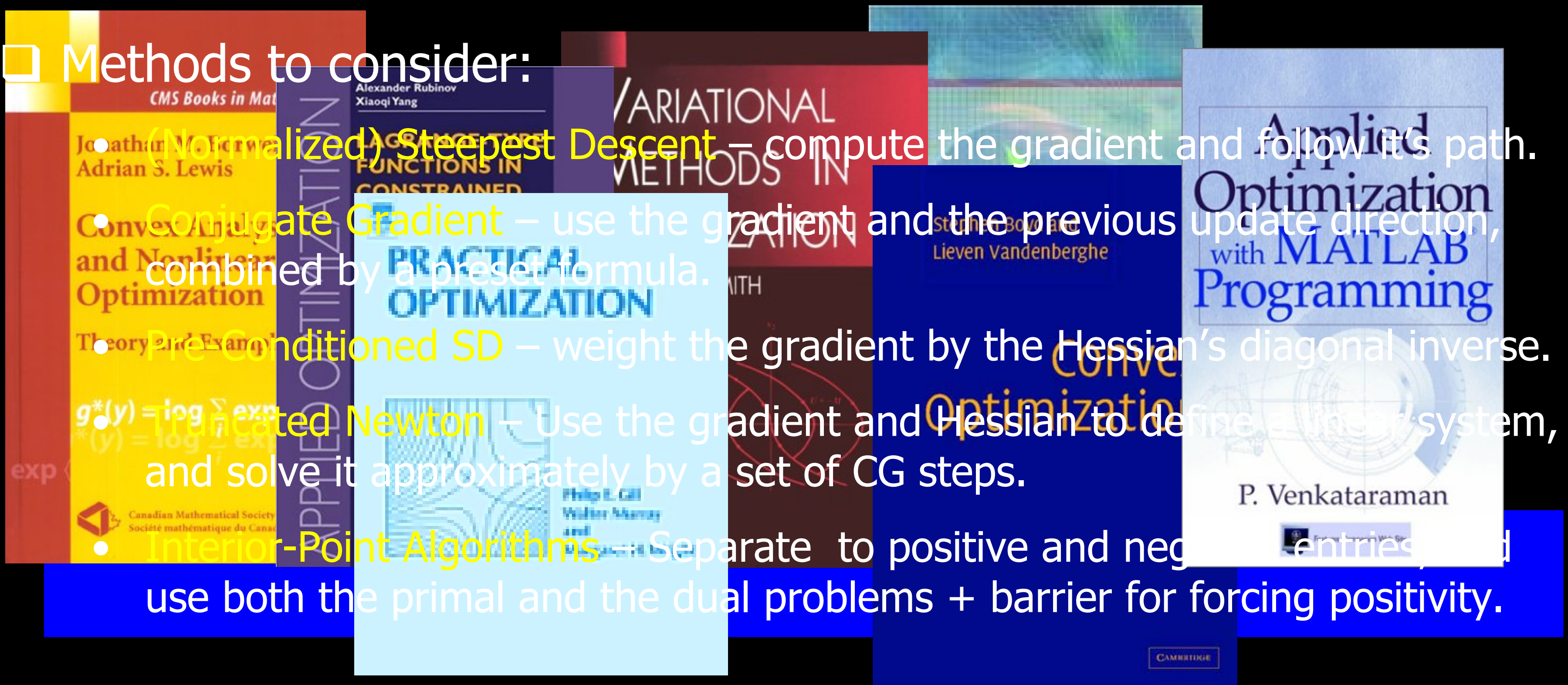


Is there a Problem?

$$f(\underline{\alpha}) = \frac{1}{2}\left\| \mathbf{D}\underline{\alpha} - \underline{x} \right\|_2^2 + \lambda\rho(\underline{\alpha})$$

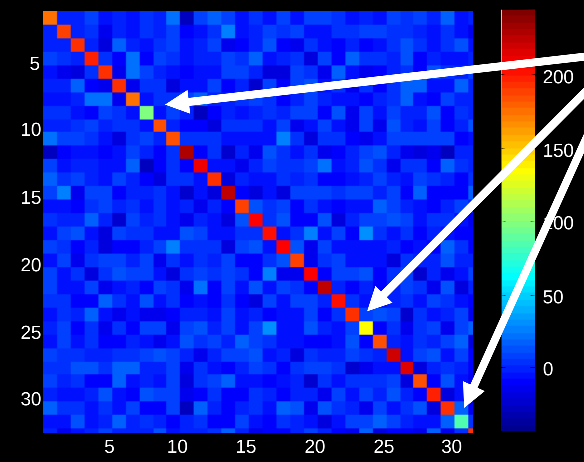
- **The first thought:** With all the existing knowledge in optimization, we could find a solution.

- Methods to consider:



  - (Normalized) Steepest Descent – compute the gradient and follow it's path.
  - Conjugate Gradient – use the gradient and the previous update direction, combined by a preset formula.
  - Pre-Conditioned SD – weight the gradient by the Hessian's diagonal inverse.
  - Truncated Newton – Use the gradient and Hessian to define a linear system, and solve it approximately by a set of CG steps.
  - Interior-Point Algorithms – Separate to positive and negative entries, and use both the primal and the dual problems + barrier for forcing positivity.

General-Purpose Software?



Relatively small entries for the non-zero entries in the solution

- **A Problem:** General purpose software-packages (algorithms) are typically performing poorly on our task.
  - The fact that the solution is expected to be sparse (or nearly so) in our problem is not exploited in such algorithms.
  - The Hessian of $f(\underline{\alpha})$ tends to be highly ill-conditioned near the (sparse) solution.

$$\nabla f(\underline{\alpha}) = \mathbf{D}^H(\mathbf{D}\underline{\alpha} - \underline{x}) + \lambda\rho'(\underline{\alpha})$$
$$\nabla^2 f(\underline{\alpha}) = \mathbf{D}^H\mathbf{D} + \lambda\rho''(\underline{\alpha})$$

- So, are we stuck? Is this problem really that complicated?

Consider the Unitary Case ($\mathbf{DD}^H = \mathbf{I}$)

$$f(\underline{\alpha}) = \frac{1}{2}\left\|\mathbf{D}\underline{\alpha} - \underline{x}\right\|_2^2 + \lambda\rho(\underline{\alpha})$$

Use $\mathbf{DD}^H = \mathbf{I}$

$$f(\underline{\alpha}) = \frac{1}{2}\left\|\mathbf{D}\underline{\alpha} - \mathbf{DD}^H\underline{x}\right\|_2^2 + \lambda\rho(\underline{\alpha})$$

Define $\mathbf{D}^H\underline{x} = \underline{\beta}$

$$f(\underline{\alpha}) = \frac{1}{2}\left\|\mathbf{D}(\underline{\alpha} - \underline{\beta})\right\|_2^2 + \lambda\rho(\underline{\alpha})$$

L_2 is unitarily invariant

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

$$= \sum_{j=1}^{m}\left[\frac{1}{2}(\alpha_j - \beta_j)^2 + \lambda\rho(\alpha_j)\right]$$

We got a separable set of m identical 1D optimization problems

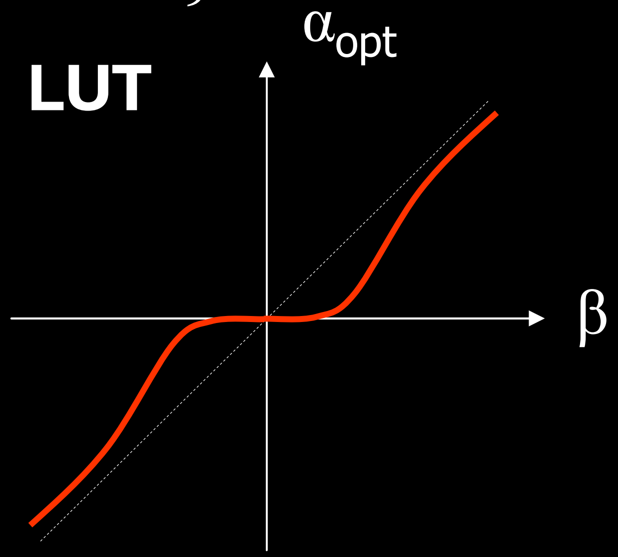
The 1D Task

We need to solve the following 1D problem:

$$\alpha_{opt} = \underset{\alpha}{\text{ArgMin}}\left\{\frac{1}{2}(\alpha-\beta)^2 + \lambda\rho(\alpha)\right\}$$

$$\boxed{\alpha_{opt} = S_{\rho,\lambda}(\beta)}$$

LUT



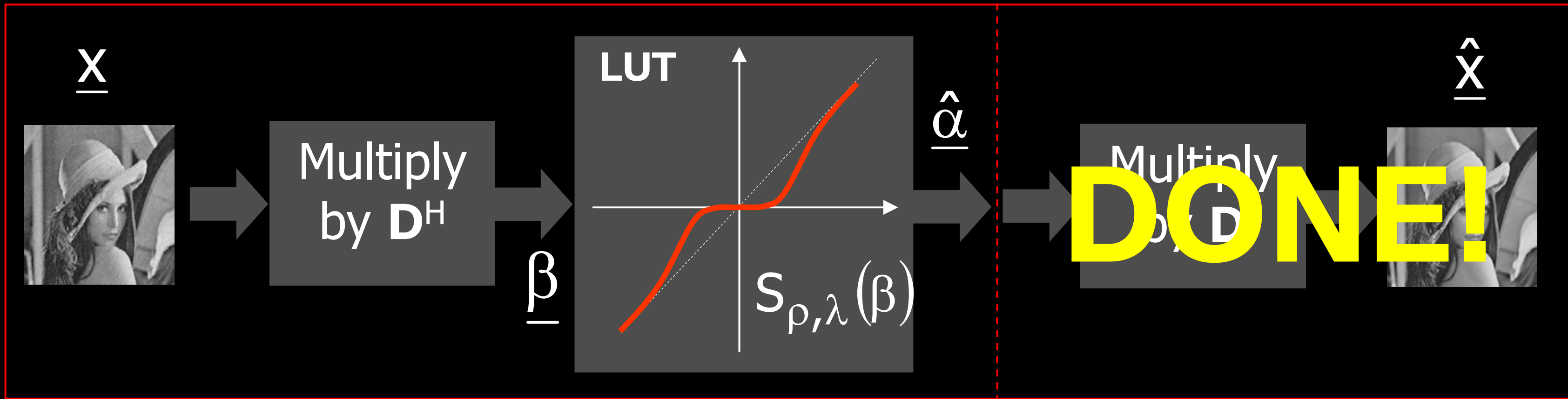
Such a Look-Up-Table (LUT) $\alpha_{opt}=S_{\rho,\lambda}(\beta)$ can be built for **ANY** sparsity measure function $\rho(\alpha)$, including non-convex ones and non-smooth ones (e.g., L_0 norm), giving in all cases the **GLOBAL** minimizer of $g(\alpha)$.

The Unitary Case: A Summary

Minimizing $f(\underline{\alpha}) = \frac{1}{2}\|\mathbf{D}\underline{\alpha} - \underline{x}\|_2^2 + \lambda\rho(\underline{\alpha}) = \frac{1}{2}\|\underline{\alpha} - \underline{\beta}\|_2^2 + \lambda\rho(\underline{\alpha})$

$\mathbf{D}^H\underline{x} = \underline{\beta}$

is done by:



The obtained solution is the GLOBAL minimizer of $f(\underline{\alpha})$, even if $f(\underline{\alpha})$ is non-convex.

Brief Summary #2

The minimization of

$$f(\underline{\alpha}) = \frac{1}{2}\|\mathbf{D}\underline{\alpha} - \underline{x}\|_2^2 + \lambda\rho(\underline{\alpha})$$

Leads to two very **Contradicting Observations**:

1. The problem is **quite hard** – classic optimization find it hard.
2. The problem is **trivial** for the case of unitary **D**.

How Can We Enjoy This Simplicity in the General Case?

Agenda

1. Motivating the Minimization of f(α)

   Describing various applications that need this mi

2. Some Motivating Facts

   General purpose optimization tools, and the unit

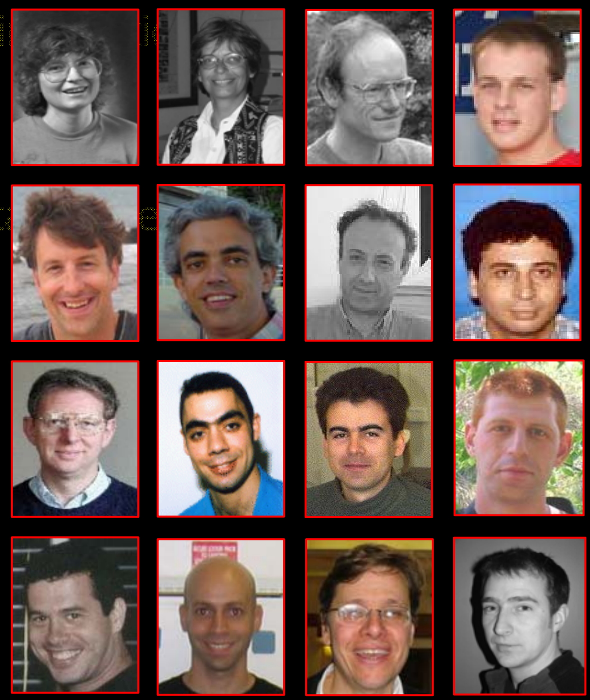
3. **Iterated-Shrinkage Algorithms**

   We describe five versions of those in detail

4. Some Results

   Image deblurring results

5. Conclusions

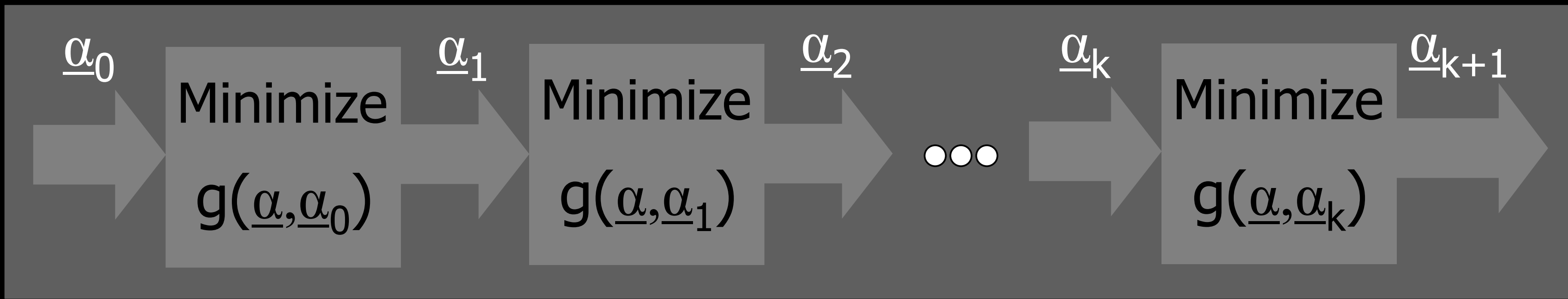


Iterated-Shrinkage Algorithms?

- We will present **THE PRINCIPLES** of several leading methods:
  - Bound-Optimization and EM [Figueiredo & Nowak, `03],
  - Surrogate-Separable-Function (SSF) [Daubechies, Defrise, & De-Mol, `04],
  - Parallel-Coordinate-Descent (PCD) algorithm [Elad `05], [Matalon, et.al. `06],
  - IRLS-based algorithm [Adeyemi & Davies, `06], and
  - Stepwise-Ortho-Matching Pursuit (StOMP) [Donoho et.al. `07].
- Common to all is a set of operations in every iteration that includes:
  (i) Multiplication by **D**,
  (ii) Multiplication by $\mathbf{D}^H$, and
  (iii) A Scalar shrinkage on the solution $S_{\rho,\lambda}(\underline{\alpha})$.
- Some of these algorithms pose a direct generalization of the unitary case, their 1st iteration is the solver we have seen.

1. The Proximal-Point Method

- Aim: minimize $f(\underline{\alpha})$ – Suppose it is found to be too hard.
- Define a surrogate-function $g(\underline{\alpha},\underline{\alpha}_0)=f(\underline{\alpha})+dist(\underline{\alpha}-\underline{\alpha}_0)$, using a general (uni-modal, non-negative) distance function.
- Then, the following algorithm necessarily converges to a local minima of $f(\underline{\alpha})$ [Rokafellar, `76]:



- Comments: (i) Is the minimization of $g(\underline{\alpha},\underline{\alpha}_0)$ easier? It better be!

  (ii) Looks like it will slow-down convergence. Really?

The Proposed Surrogate-Functions

- Our original function is: $f(\underline{\alpha}) = \frac{1}{2}\|\mathbf{D}\underline{\alpha} - \underline{x}\|_2^2 + \lambda\rho(\underline{\alpha})$

- The distance to use: $\text{dist}(\underline{\alpha}, \underline{\alpha}_0) = \frac{c}{2}\cdot\|\underline{\alpha} - \underline{\alpha}_0\|_2^2 - \frac{1}{2}\|\mathbf{D}\underline{\alpha} - \mathbf{D}\underline{\alpha}_0\|_2^2$

  Proposed by [Daubechies, Defrise, & De-Mol `04]. Require $c > r(\mathbf{D}^H\mathbf{D})$.

- The beauty in this choice: the term $\|\mathbf{D}\underline{\alpha}\|_2^2$ vanishes ………

  $$g(\underline{\alpha}, \underline{\alpha}_0) = \lambda\rho(\underline{\alpha}) + \frac{c}{2}\|\underline{\alpha}\|_2^2 - \underline{\alpha}^H\underline{\beta}_0 \quad \text{where} \quad \underline{\beta}_0 = \mathbf{D}^H(\underline{x} - \mathbf{D}\underline{\alpha}_0) + c\underline{\alpha}_0$$

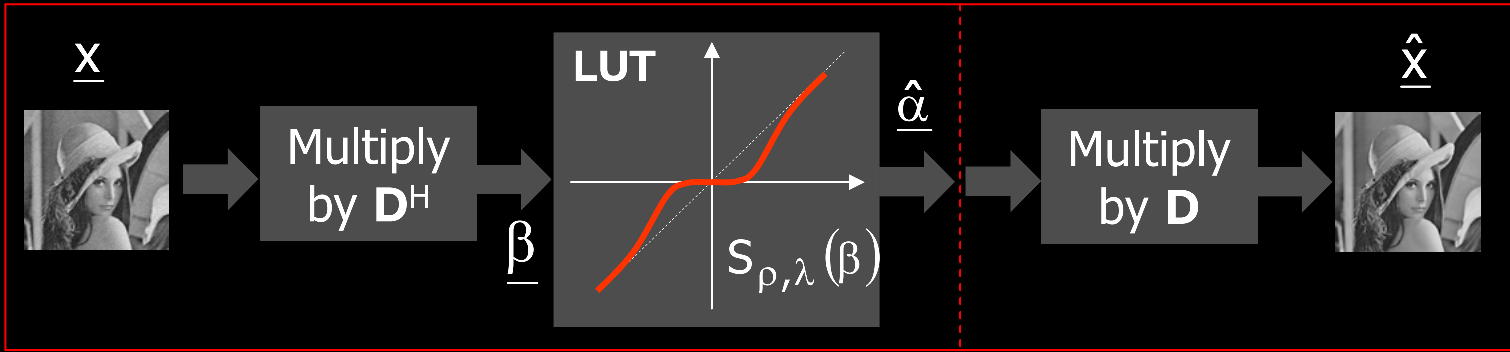
- Minimization of $g(\underline{\alpha}, \underline{\alpha}_0)$ is done in a closed form by shrinkage, done on the vector $\underline{\beta}_k$, and this generates the solution $\underline{\alpha}_{k+1}$ of the next iteration.

$$= c\sum_{j=1}^{m}\left[\frac{\lambda}{c}\rho(\alpha_j) + \frac{1}{2}\alpha_j^2 - \frac{\alpha_j^*\beta_j^0}{c}\right]$$

It is a separable sum of m 1D problems. Thus, we have a closed form solution by THE SAME SHRINKAGE !!

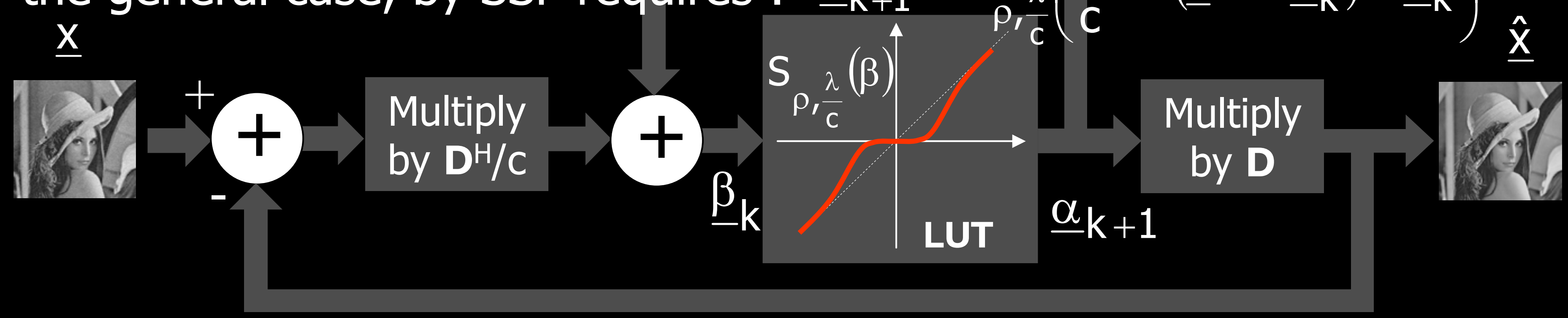
The Resulting SSF Algorithm

While the Unitary case solution is given by



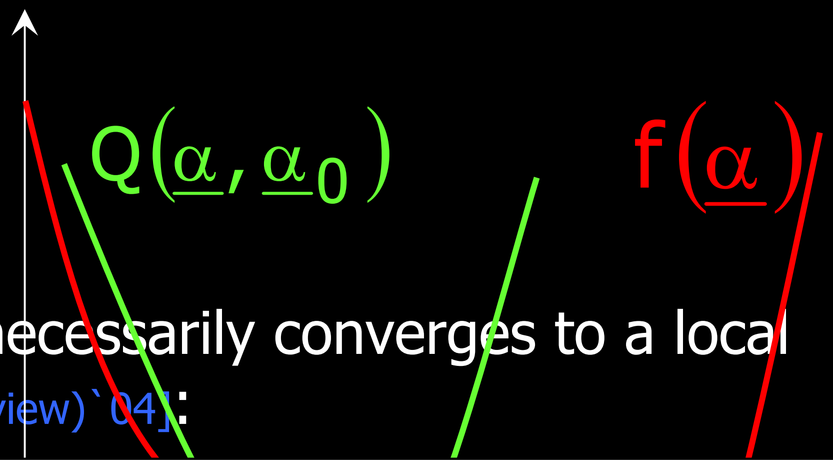
$$\hat{\underline{\alpha}} = S_{\rho,\lambda}\left(\mathbf{D}^H\underline{x}\right) \; ; \; \hat{\underline{x}} = \mathbf{D}\hat{\underline{\alpha}}$$

the general case, by SSF requires : $\underline{\alpha}_{k+1} = S_{\rho,\frac{\lambda}{c}}\left(\frac{1}{c}\mathbf{D}^H\left(\underline{x} - \mathbf{D}\underline{\alpha}_k\right) + \underline{\alpha}_k\right)$



2. Bound-Optimization Technique

- Aim: minimize $f(\underline{\alpha})$ – Suppose it is found to be too hard.
- Define a function $Q(\underline{\alpha},\underline{\alpha}_0)$ that satisfies the following conditions:
  - $Q(\underline{\alpha}_0,\underline{\alpha}_0)=f(\underline{\alpha}_0)$,
  - $Q(\underline{\alpha},\underline{\alpha}_0)\geq f(\underline{\alpha})$ for all $\underline{\alpha}$, and
  - $\nabla Q(\underline{\alpha},\underline{\alpha}_0)= \nabla f(\underline{\alpha})$ at $\underline{\alpha}_0$.



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

$\underline{\alpha}_0$ → Minimize $Q(\underline{\alpha},\underline{\alpha}_0)$ → $\underline{\alpha}_1$ → Minimize $Q(\underline{\alpha},\underline{\alpha}_1)$ → $\underline{\alpha}_2$ → ●●● → $\underline{\alpha}_k$ → Minimize $Q(\underline{\alpha},\underline{\alpha}_k)$ → $\underline{\alpha}_{k+1}$

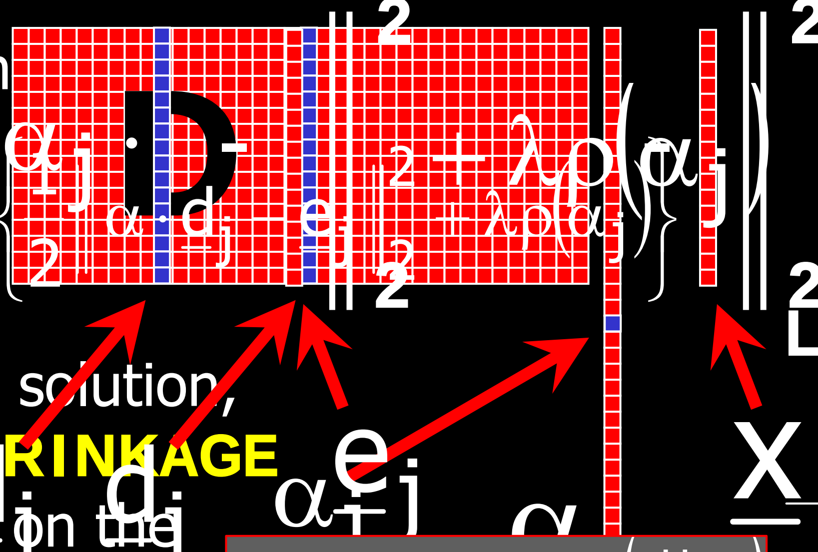
3. Start With Coordinate Descent

- We aim to minimize $f(\underline{\alpha}) = \frac{1}{2}\left\| \mathbf{D}\underline{\alpha} - \underline{x} \right\|_2^2 + \lambda\rho(\underline{\alpha})$.
- First, consider the Coordinate Descent (CD) algorithm.
- This is a 1D minim

$$f(\alpha_j) = \frac{1}{2}\left\| \alpha_j \underline{d}_j - \underline{e}_j \right\|_2^2 + \lambda\rho(\alpha_j)$$

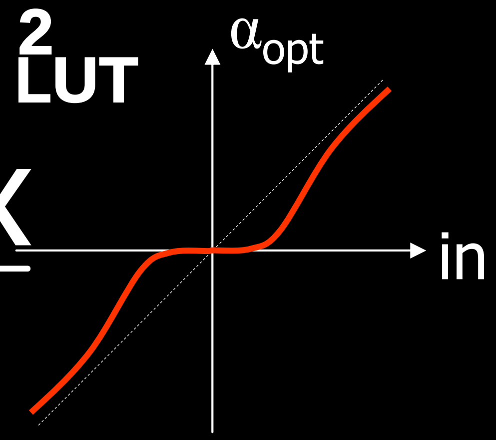
$$\alpha_{opt} = \underset{\alpha}{\text{ArgMin}}\left\{ \frac{1}{2}\left\| \alpha \cdot \underline{d}_j - \underline{e}_j \right\|_2^2 + \lambda\rho(\alpha) \right\}$$

$$f(\alpha_j) = \frac{1}{2}\left\| \mathbf{D} - \underline{x} \right\|_2^2 + \lambda\rho(\alpha_j)$$



- It has a closed for solution, using a simple **SHRINKAGE** as before, applied on the scalar $<\underline{e}_j, \underline{d}_j>$.

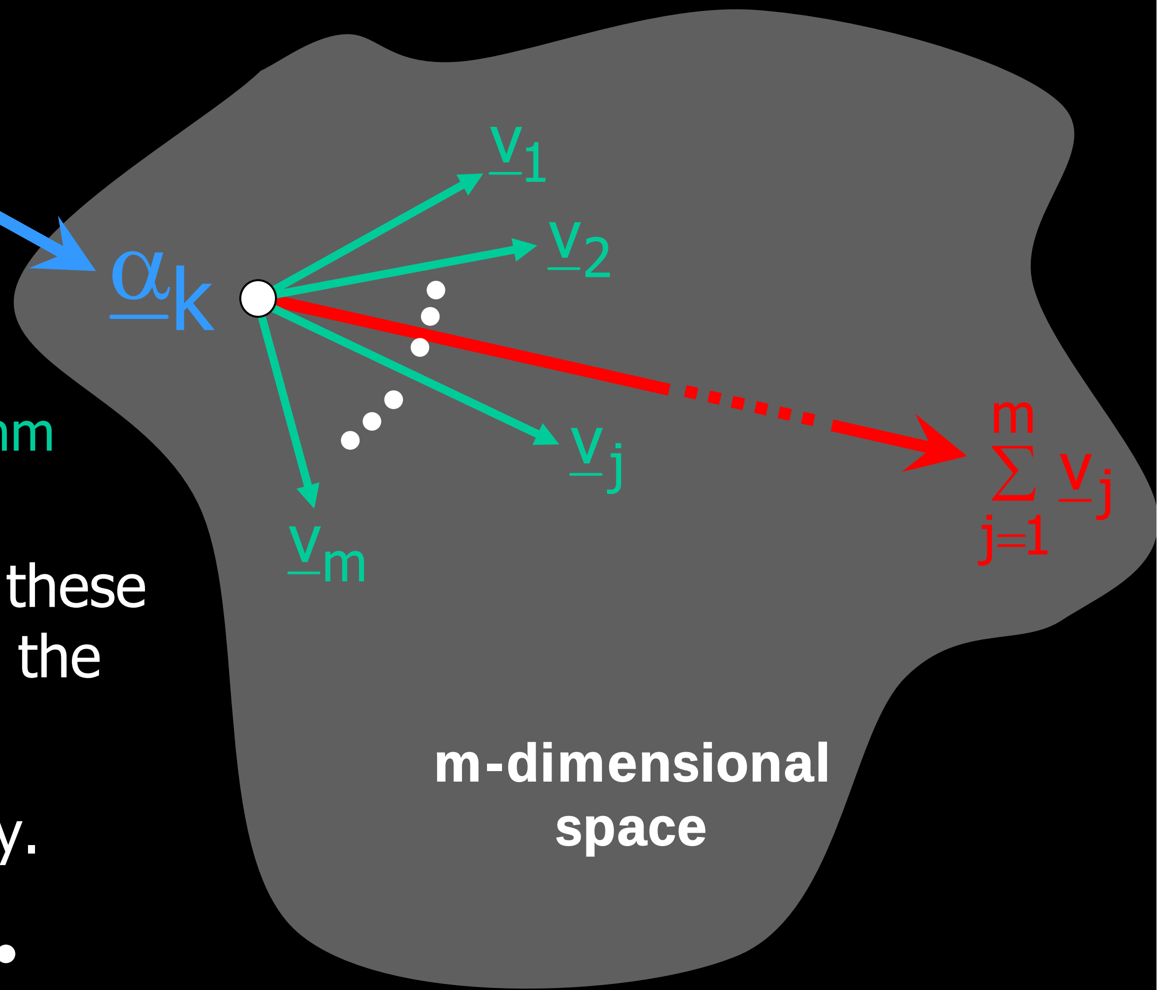
$$\alpha_{opt} = S_{\rho, \lambda / \left\| \underline{d}_j \right\|^2}\left(\underline{d}_j^H \underline{e}_j \right)$$



Parallel Coordinate Descent (PCD)

Current solution for minimization of $f(\underline{\alpha})$

$\{\underline{v}_j\}_{j=1}^{m}$: Descent directions obtained by the previous CD algorithm

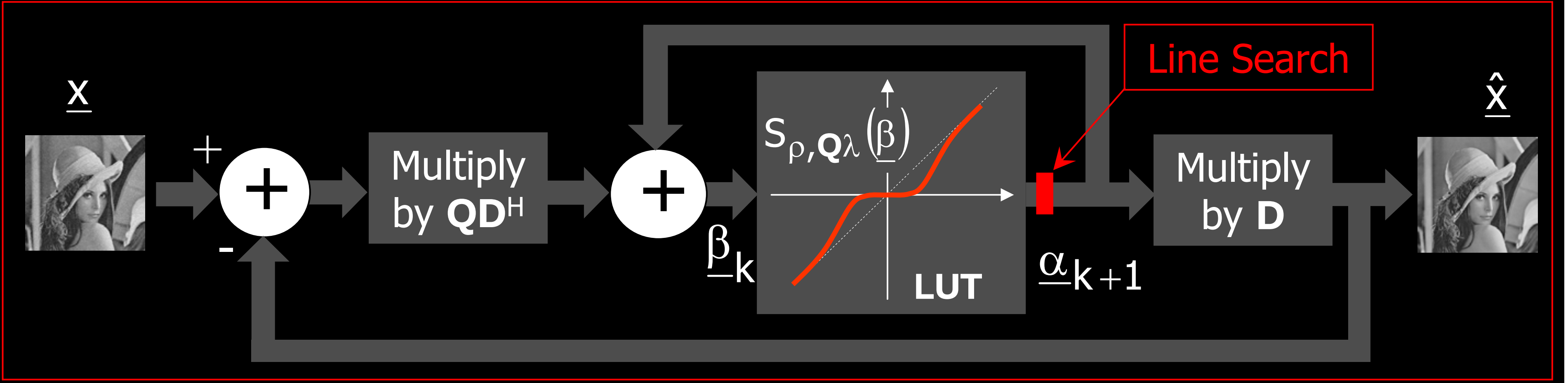


- We will take the sum of these m descent directions for the update step.
- Line search is mandatory.
- This leads to ………

The PCD Algorithm [Elad, `05] [Matalon, Elad, & Zibulevsky, `06]

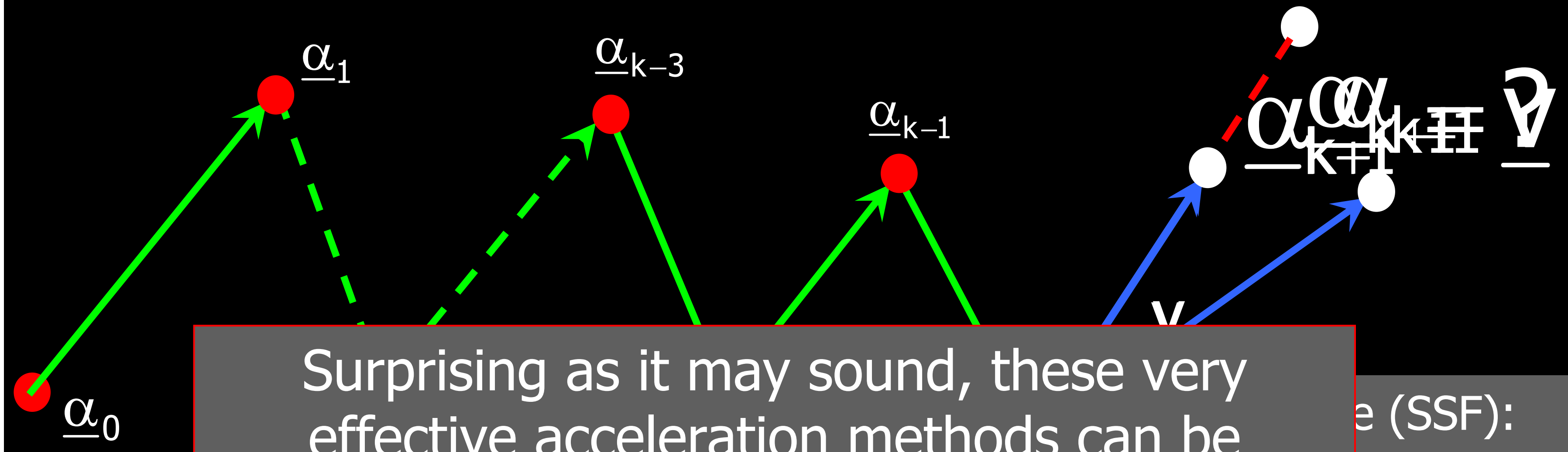
$$\underline{\alpha}_{k+1} = \underline{\alpha}_k + \mu\left[S_{\rho,\mathbf{Q}\lambda}\left(\mathbf{Q}\mathbf{D}^H\left(\underline{x} - \mathbf{D}\underline{\alpha}_k\right) + \underline{\alpha}_k\right) - \underline{\alpha}_k\right]$$

Where $\mathbf{Q} = \mathrm{diag}^{-1}\left(\mathbf{D}^H\mathbf{D}\right)$ and μ represents a line search (LS).



Note: **Q** can be computed quite easily off-line. Its storage is just like storing the vector $\underline{\alpha}_k$.

Algorithms' Speed-Up



Surprising as it may sound, these very effective acceleration methods can be implemented with no additional "cost" (i.e., multiplications by **D** or **D**^{T})

Option

e (SSF):

$(x - \mathbf{D}\underline{\alpha}_k) + \underline{\alpha}_k \}$

SOP):

$$\underline{\alpha}_{k+1} = \underline{\alpha}_k + \begin{bmatrix} | & & | & | & | \\ \underline{\alpha}_{k-M+1}-\underline{\alpha}_{k-M} & \cdots & \underline{\alpha}_{k-1}-\underline{\alpha}_{k-2} & \underline{\alpha}_k-\underline{\alpha}_{k-1} & \underline{v}-\underline{\alpha}_k \\ | & & | & | & | \end{bmatrix} \begin{bmatrix} \mu_M \\ \vdots \\ \mu_0 \end{bmatrix}$$

[Zibulevsky & Narkis, '05]

[Elad, Matalon, & Zibulevsky, `07]

Brief Summary #3

For an effective minimization of the function

$$f(\underline{\alpha}) = \frac{1}{2}\|\mathbf{D}\underline{\alpha} - \underline{x}\|_2^2 + \lambda\rho(\underline{\alpha})$$

we saw several iterated-shrinkage algorithms, built using

1. Proximal Point Method
2. Bound Optimization
3. Parallel Coordinate Descent
4. Iterative Reweighed LS
5. Fixed Point Iteration
6. Greedy Algorithms

How Are They Performing?

Agenda

1. Motivating the Minimization of f(α)

    Describing various a

2. Some Motivating

    General purpose op

3. Iterated-Shrinkag

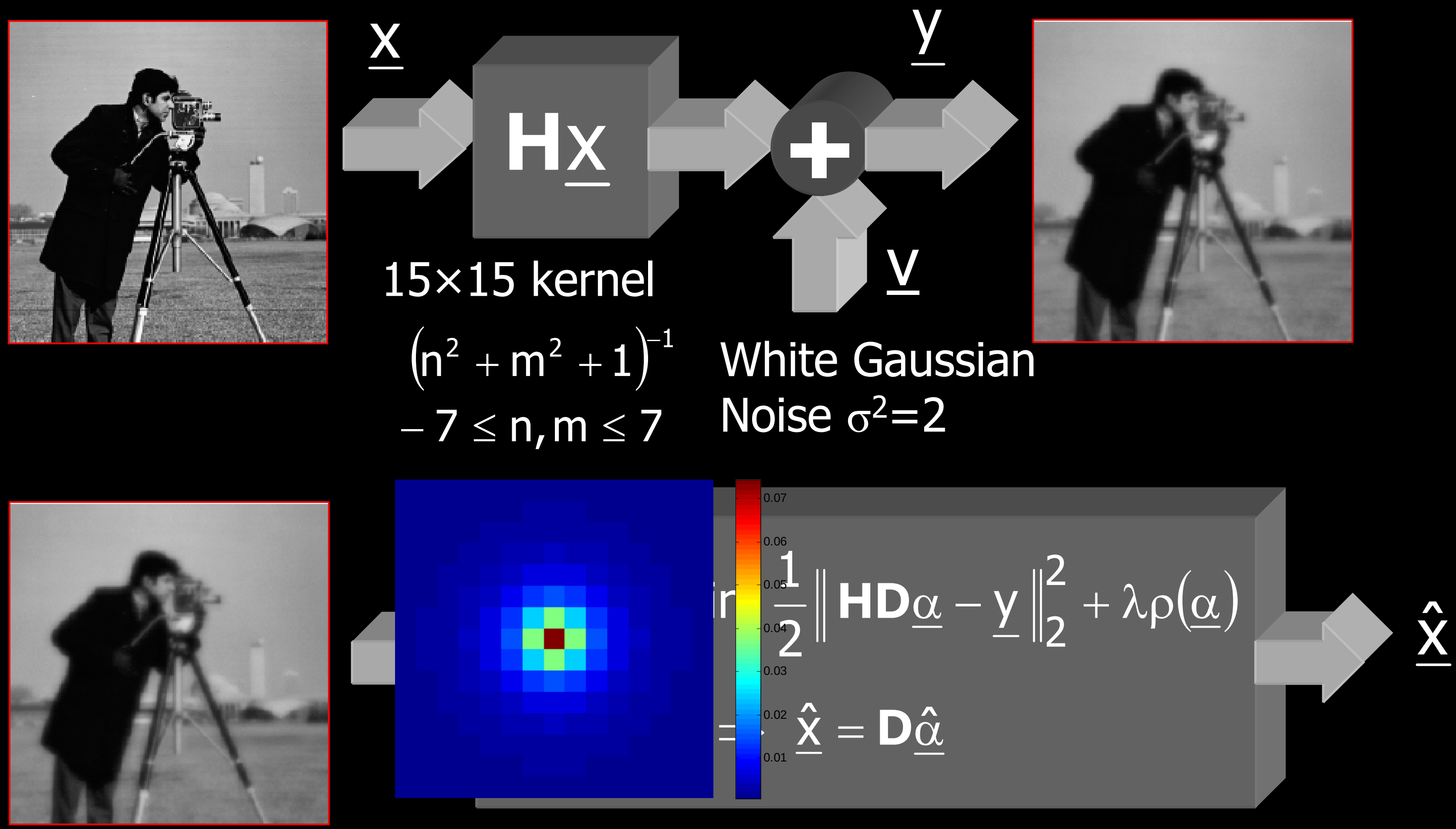
    We describe five ve

4. **Some Results**

    Image deblurring results

5. Conclusions

A Deblurring Experiment



Penalty Function: More Details

A given blurred and noisy image of size 256×256

$\lambda = 0.075$

The blur operator

$$f(\alpha) = \frac{1}{2}\left\| \mathbf{HD}\alpha - \mathbf{y} \right\|^2 + \lambda\rho(\alpha)$$

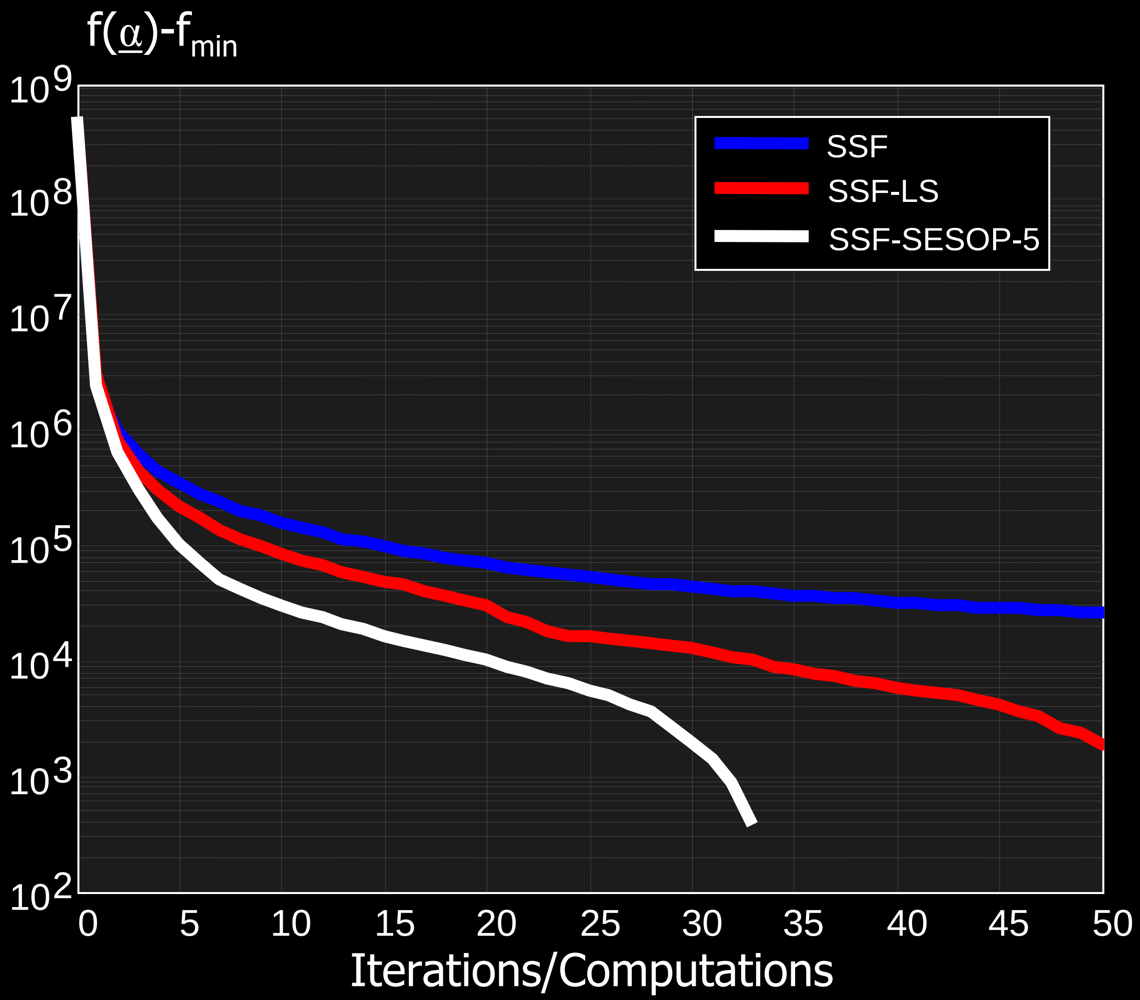
Note: This experiment is similar (but not eqiuvalent) to one of tests done in [Figueiredo & Nowak `05], that leads to state-of-the-art results.

2D un-d
transform, 5 resolution layers,
7:1 redundancy

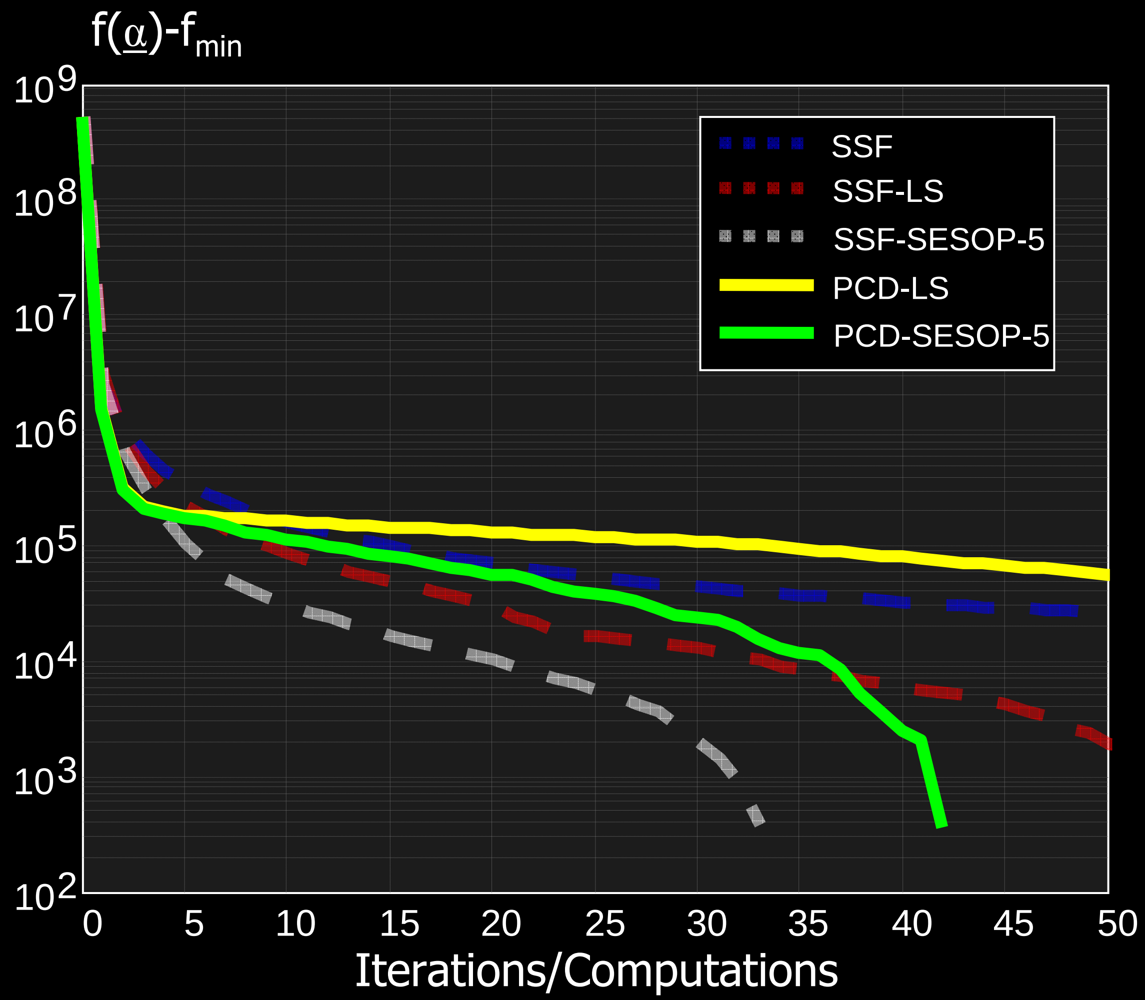
nation of L_1 with slight (S=0.01) smoothing

$$\rho(\alpha) = |\alpha| - S \cdot \log\left(1 + \frac{|\alpha|}{S}\right)$$

So, The Results: The Function Value



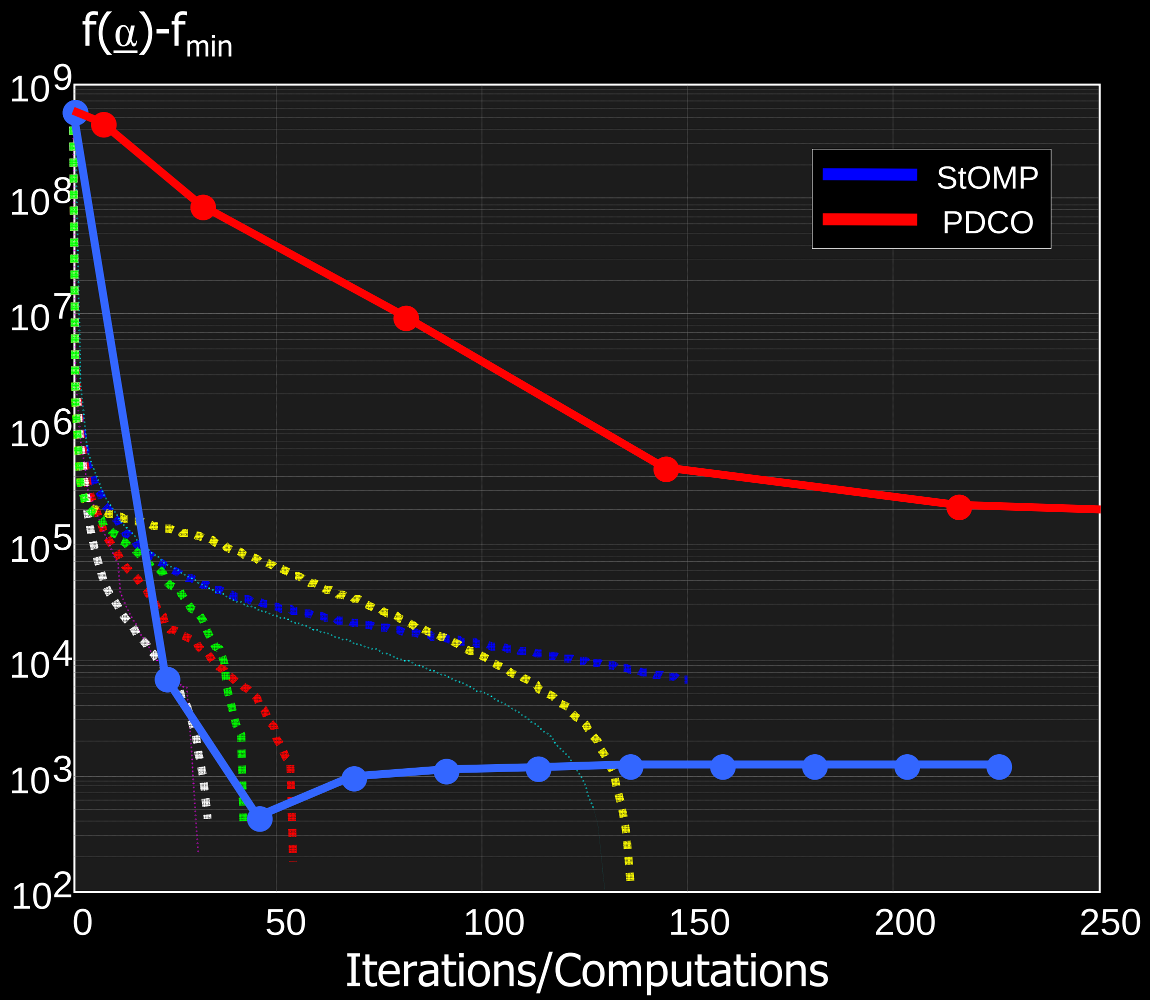
So, The Results: The Function Value



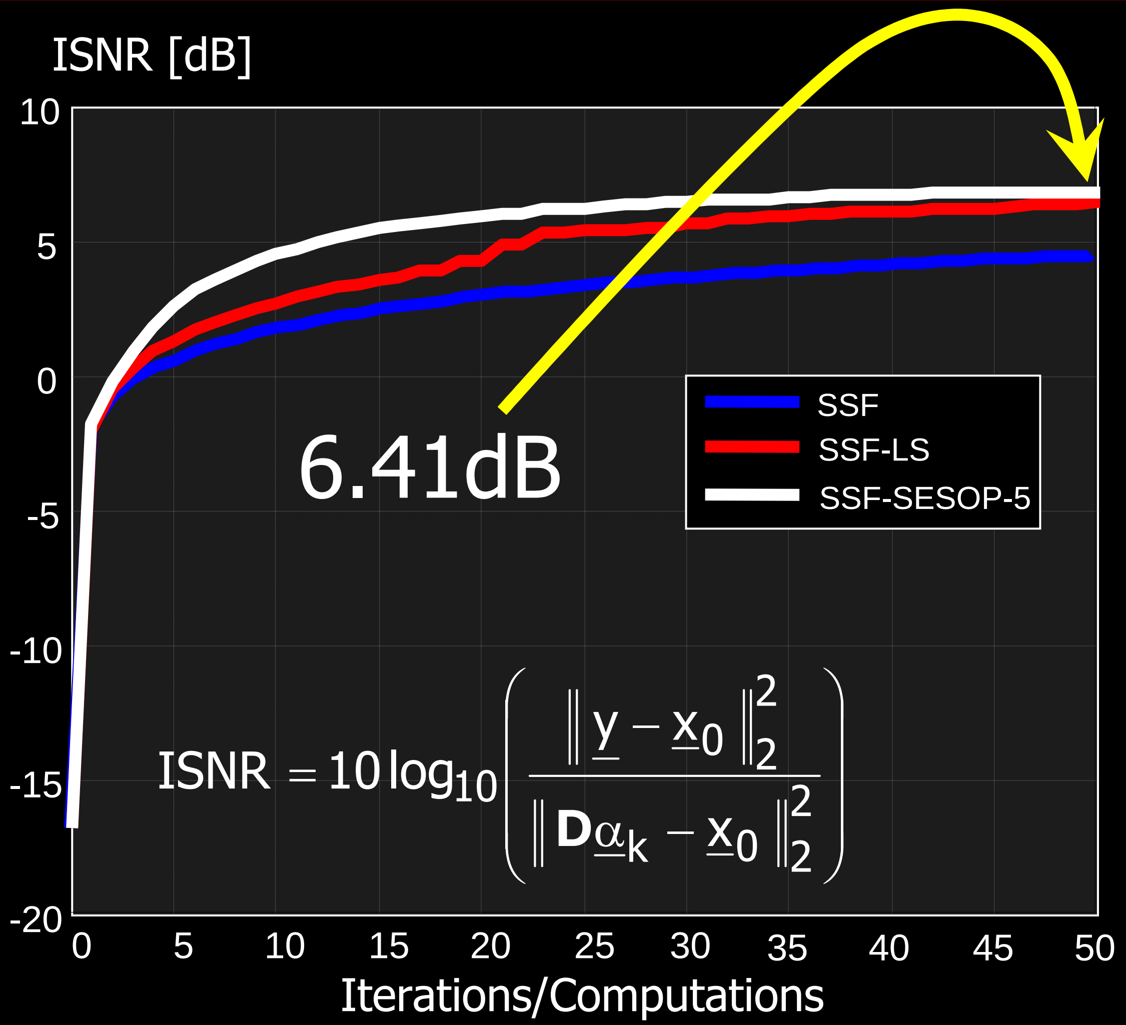
Comment:

Both SSF and PCD (and their accelerated versions) are provably converging to the minima of $f(\underline{\alpha})$.

So, The Results: The Function Value

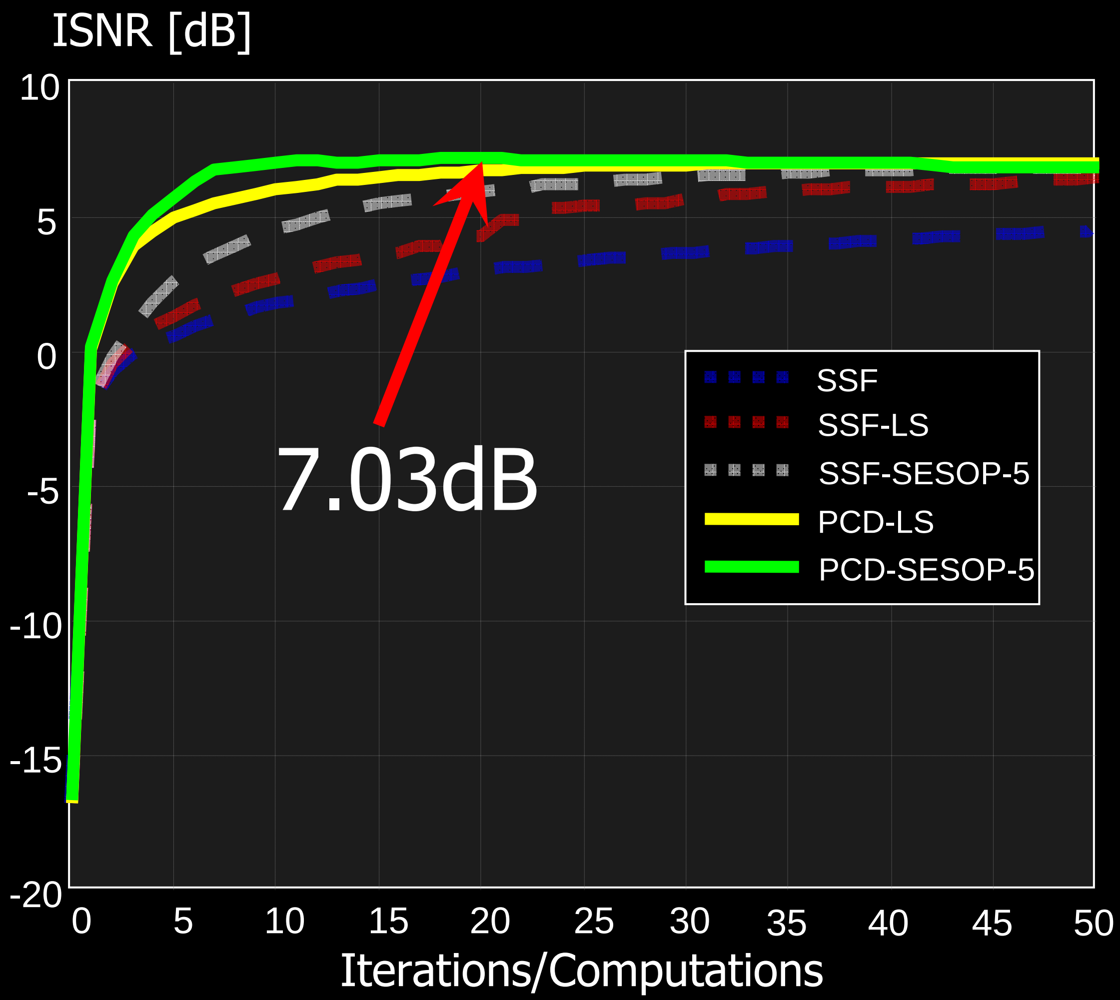


So, The Results: ISNR

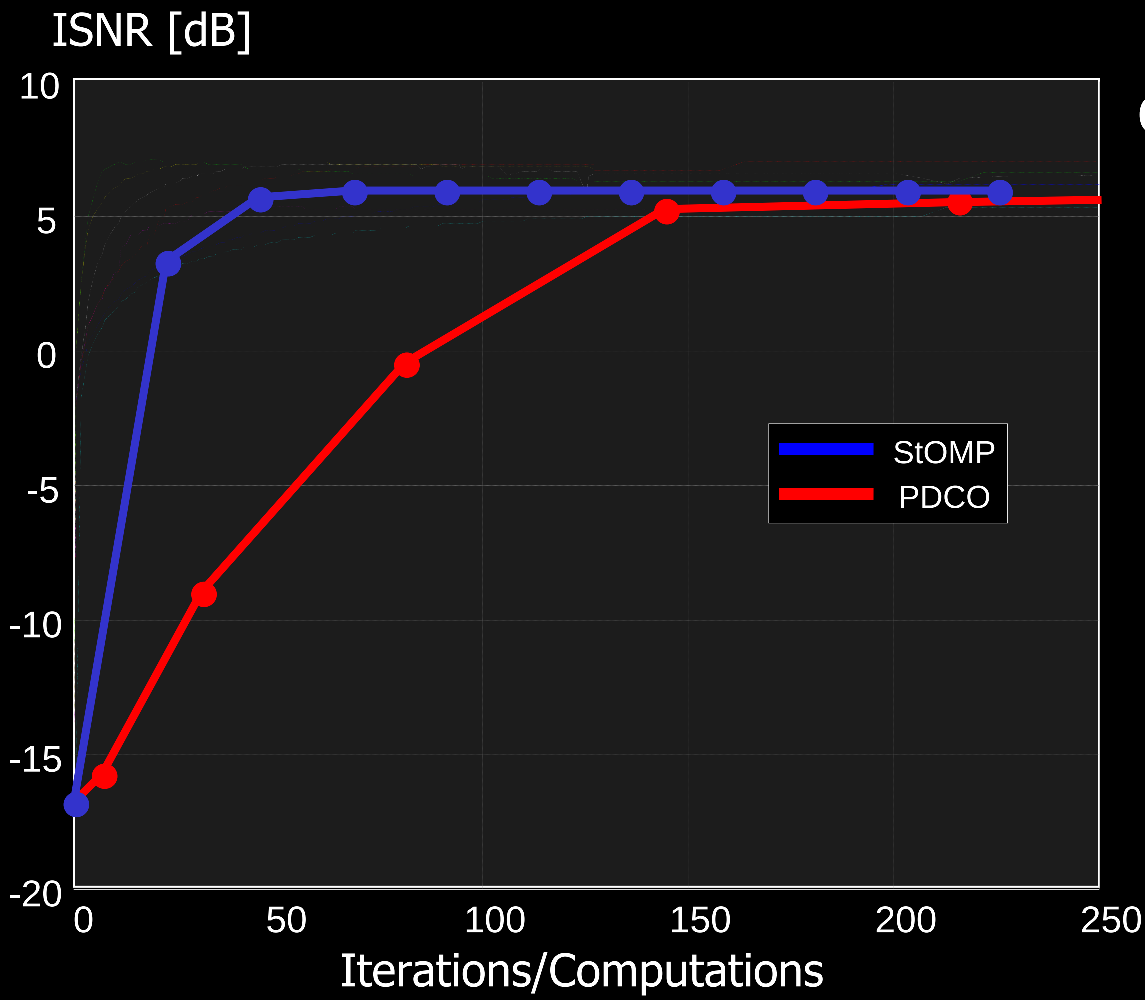


$$\mathrm{ISNR} = 10\log_{10}\left(\frac{\|\underline{y}-\underline{x}_0\|_2^2}{\|\mathbf{D}\underline{\alpha}_k-\underline{x}_0\|_2^2}\right)$$

So, The Results: ISNR



So, The Results: ISNR



Comments:

StOMP is inferior in speed and final quality (ISNR=5.91dB) due to to over-estimated support.

PDCO is very slow due to the numerous inner Least-Squares iterations done by CG. It is not competitive with the Iterated-Shrinkage methods.

Visual Results

PCD-SESOP-5 Results:

original (left), Measured (middle), and Restored (right): Iteration [illegible]

Agenda

1. **Motivating the Minimization of f(α)**
   Describing various applications that need this minimization
2. **Some Motivating Facts**
   General purpose optimization tools, and the unitary case
3. **Iterated-Shrinkage Algorithms**
   We describe five versions of those in detail
4. **Some Results**
   Image deblurring results
5. **Conclusions**

Conclusions – The Bottom Line

If your work leads you to the need to minimize the problem:

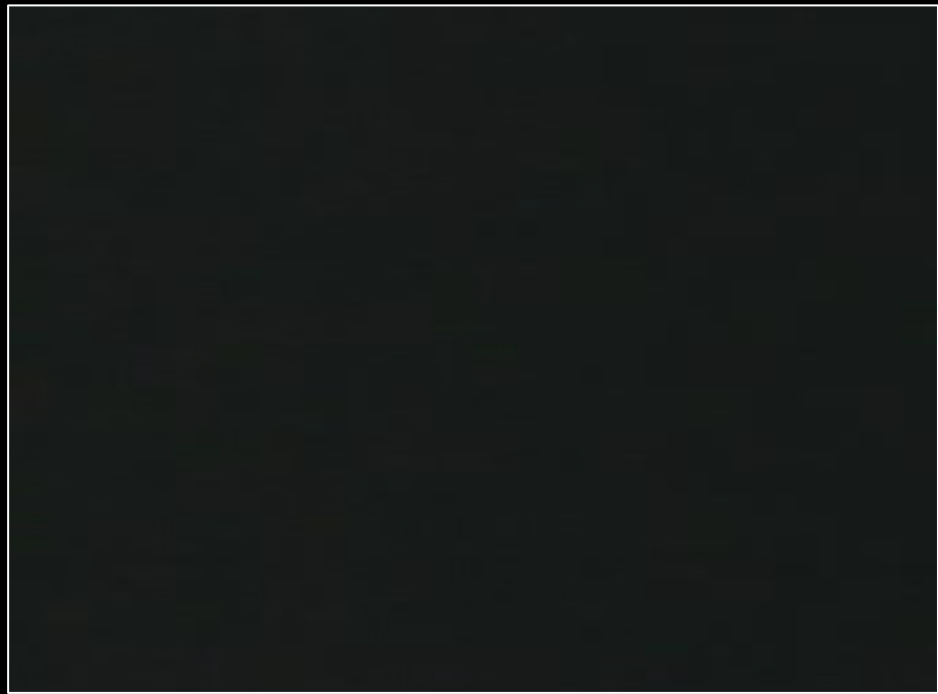
$$f(\underline{\alpha}) = \frac{1}{2}\left\|\mathbf{D}\underline{\alpha} - \underline{x}\right\|_2^2 + \lambda\rho(\underline{\alpha})$$

Then:

- We recommend you use an **Iterated-Shrinkage** algorithm.
- **SSF and PCD are Preferred**: both are provably converging to the (local) minima of $f(\underline{\alpha})$, and their performance is very good, getting a reasonable result in few iterations.
- Use **SESOP Acceleration** – it is very effective, and with hardly any cost.
- There is **Room** for more work on various aspects of these algorithms – see the accompanying paper.

Thank You for Your Time & Attention

This field of research is very hot …



More information, including these slides and the accompanying paper, can be found on my web-page
http://www.cs.technion.ac.il/~elad

THE END !!

3. The IRLS-Based Algorithm

- Use the following principles [Edeyemi & Davies `06]:

  (1) Iterative Reweighed Least-Squares (IRLS)

$$f(\underline{\alpha}) = \frac{1}{2}\|\mathbf{D}\underline{\alpha} - \underline{x}\|_2^2 + \lambda\boxed{\rho(\underline{\alpha})} \Rightarrow \boxed{\rho(\underline{\alpha}) = \underline{\alpha}^H \mathbf{W}(\underline{\alpha})\underline{\alpha}}$$

$$\mathbf{W}(\underline{\alpha}) = \begin{bmatrix} \frac{\rho(\alpha_1)}{\alpha_1^2} & & & 0 \\ & \ddots & & \\ & & \frac{\rho(\alpha_j)}{\alpha_j^2} & \\ 0 & & & \ddots \\ & & & \frac{\rho(\alpha_m)}{\alpha_m^2} \end{bmatrix}$$

$$\underline{\alpha}_{k+1} = \min_{\underline{\alpha}} \left\{ \frac{1}{2}\|\mathbf{D}\underline{\alpha} - \underline{x}\|_2^2 + \lambda \underline{\alpha}^H \mathbf{W}(\underline{\alpha}_k)\underline{\alpha} \right\}$$

Solve somehow: $\mathbf{D}^H(\mathbf{D}\underline{\alpha} - \underline{x}) + 2\lambda \mathbf{W}(\underline{\alpha})\underline{\alpha} = 0$

This is the IRLS algorithm, used in FOCUSS [Gorodinsky & Rao `

The IRLS-Based Algorithm

- Use the following principles [Edeyemi & Davies `06]:

  (2) Fixed-Point Iteration

$$-\mathbf{D}^H(\underline{x}-\mathbf{D}\underline{\alpha})+2\lambda\mathbf{W}(\underline{\alpha}_k)\underline{\alpha}+c\underline{\alpha}-c\underline{\alpha}=0$$

The Fixed-Point-Iteration method:

$$-\mathbf{D}^H(\underline{x}-\mathbf{D}\underline{\alpha}_k)+2\lambda\mathbf{W}(\underline{\alpha}_k)\underline{\alpha}_{k+1}+c\underline{\alpha}_{k+1}-c\underline{\alpha}_k=0$$

Task: solve the system $\Phi(\underline{x})-\underline{x}=0$

The actual algorithm: $\underline{x}_{k+1}=\Phi(\underline{x}_k)$

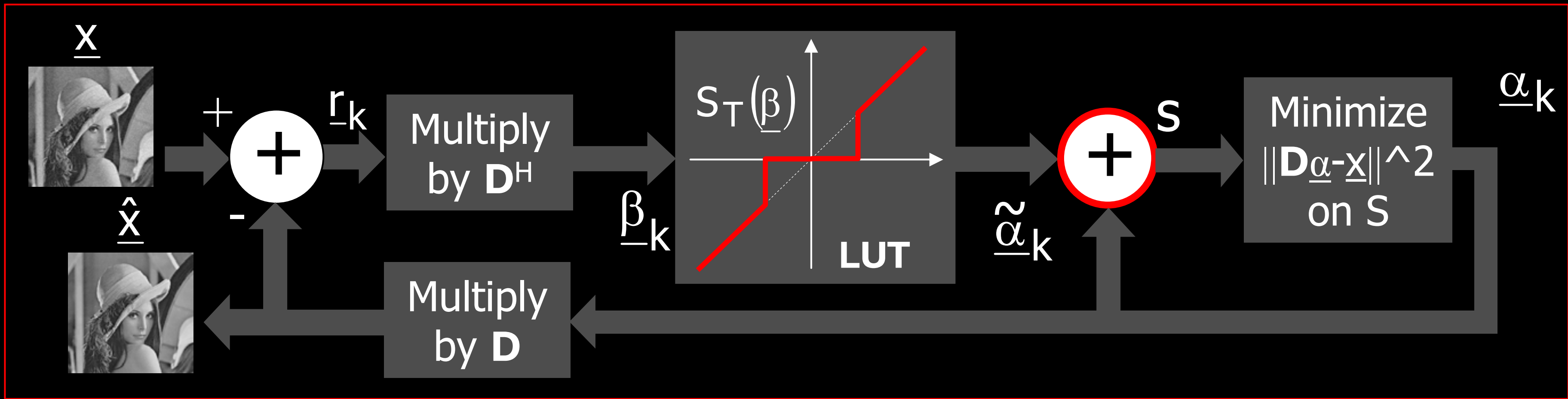
$$\underline{\alpha}_{k+1}=\left(\frac{2\lambda}{c}\mathbf{W}(\underline{\alpha}_k)+\mathbf{I}\right)^{-1}\left(\frac{1}{c}\mathbf{D}^H(\underline{x}-\mathbf{D}\underline{\alpha}_k)+\underline{\alpha}_k\right)$$

Diagonal Entries

$$\frac{\underline{\alpha}_j^2}{\frac{2\lambda}{c}\rho(\underline{\alpha}_j)+\underline{\alpha}_j^2}$$

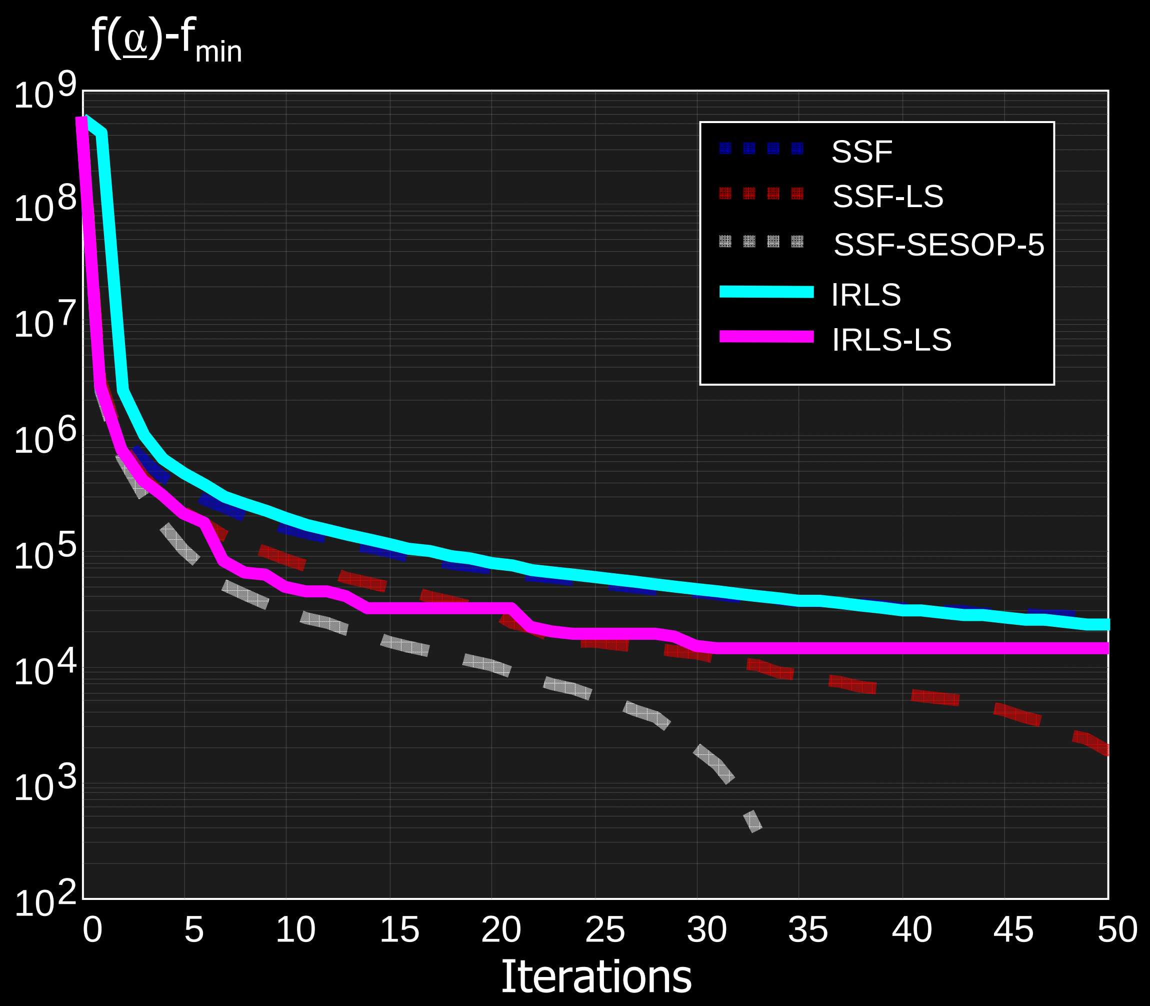
Notes: (1) For convergence, we should require $c>r(\mathbf{D}^H\mathbf{D})/2$.
(2) This algorithm cannot guarantee local-minimum.

4. Stagewise-OMP [Donoho, Drori, Starck, & Tsaig, `07]



- StOMP is originally … solv … in $\frac{1}{2}\|\mathbf{D}\underline{\alpha}-\underline{x}\|_2^2+\lambda\|\underline{\alpha}\|_0$ and especially so f … dictio … $\mathbf{D}$ (Compressed-Sensing).
- Nevertheless, it is used elsewhere (restoration) [Fadili & Starck, `06].
- If S grows by one item at each iteration, this becomes OMP.
- LS uses K_0 CG steps, each equivalent to 1 iterated-shrinkage step.

So, The Results: The Function Value



So, The Results: ISNR