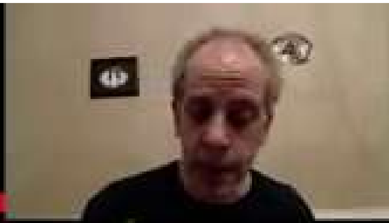
Noise Removal?

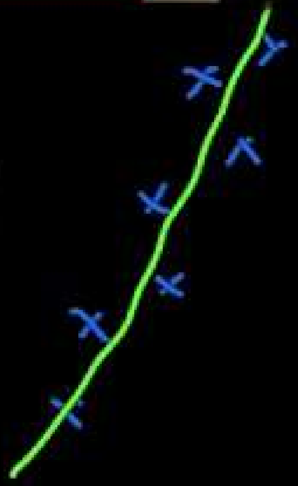
Our story begins with image denoising …

Remove
Additive
Noise

Denoising By Energy Minimization



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



$\underline{y}$: Given measurements

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

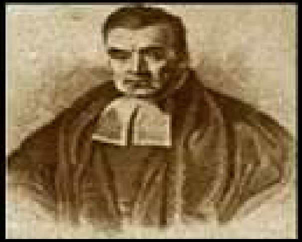
Denoising By Energy Minimization

$$f(\underline{x}) = \underbrace{\frac{1}{2}\left\|\underline{x}-\underline{y}\right\|_2^2}_{\text{Relation to measurements}} + \underbrace{G(\underline{x})}_{\text{Prior or regularization}}$$

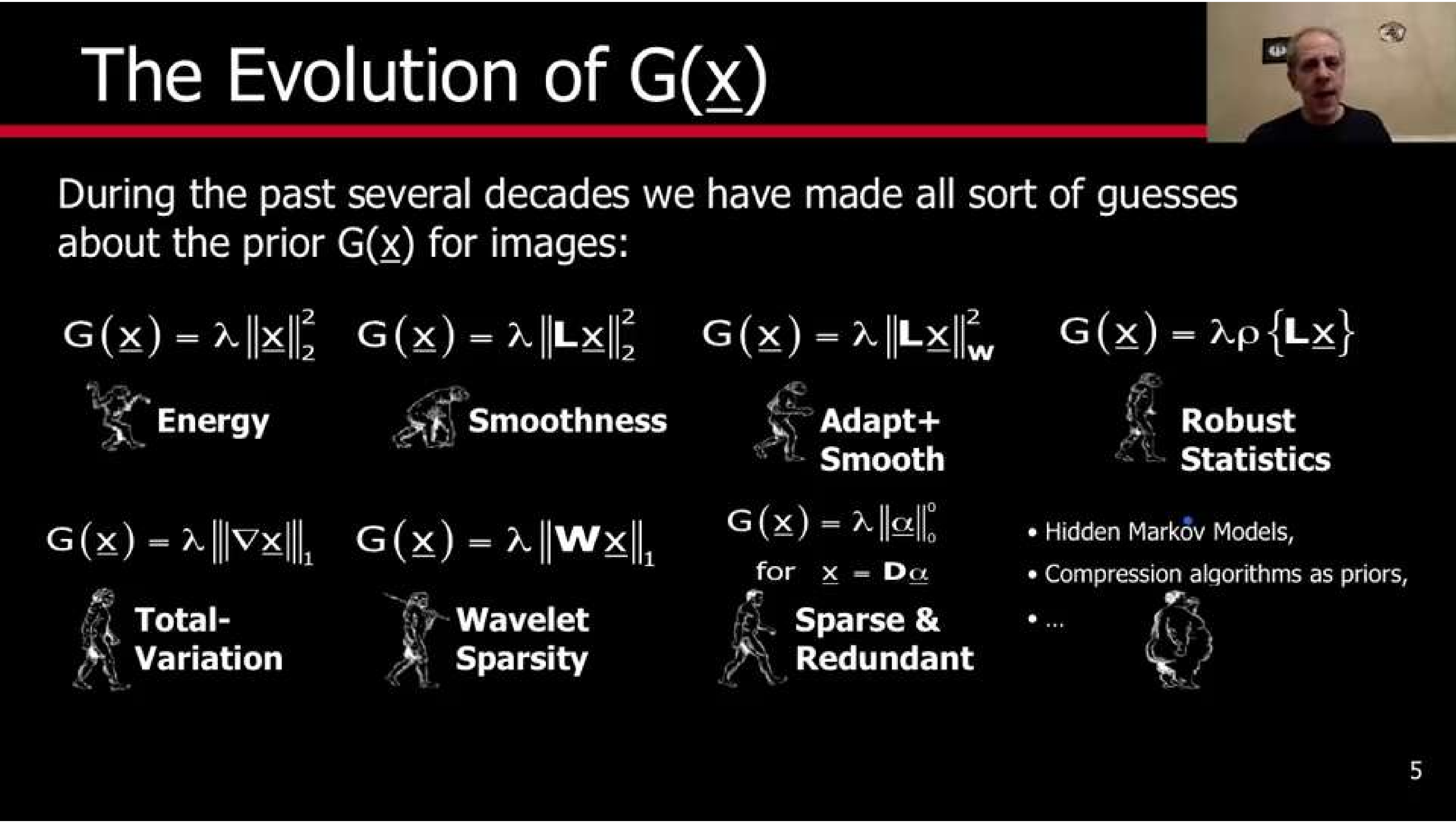
$\underline{y}$: Given measurements

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

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



Thomas Bayes
1702 - 1761



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

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

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

Energy

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

Smoothness

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

Adapt+ Smooth

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

Robust Statistics

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

Total-Variation

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

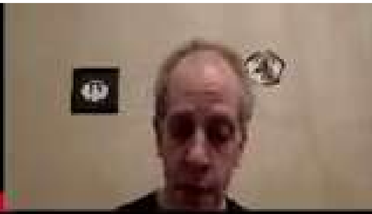
Wavelet Sparsity

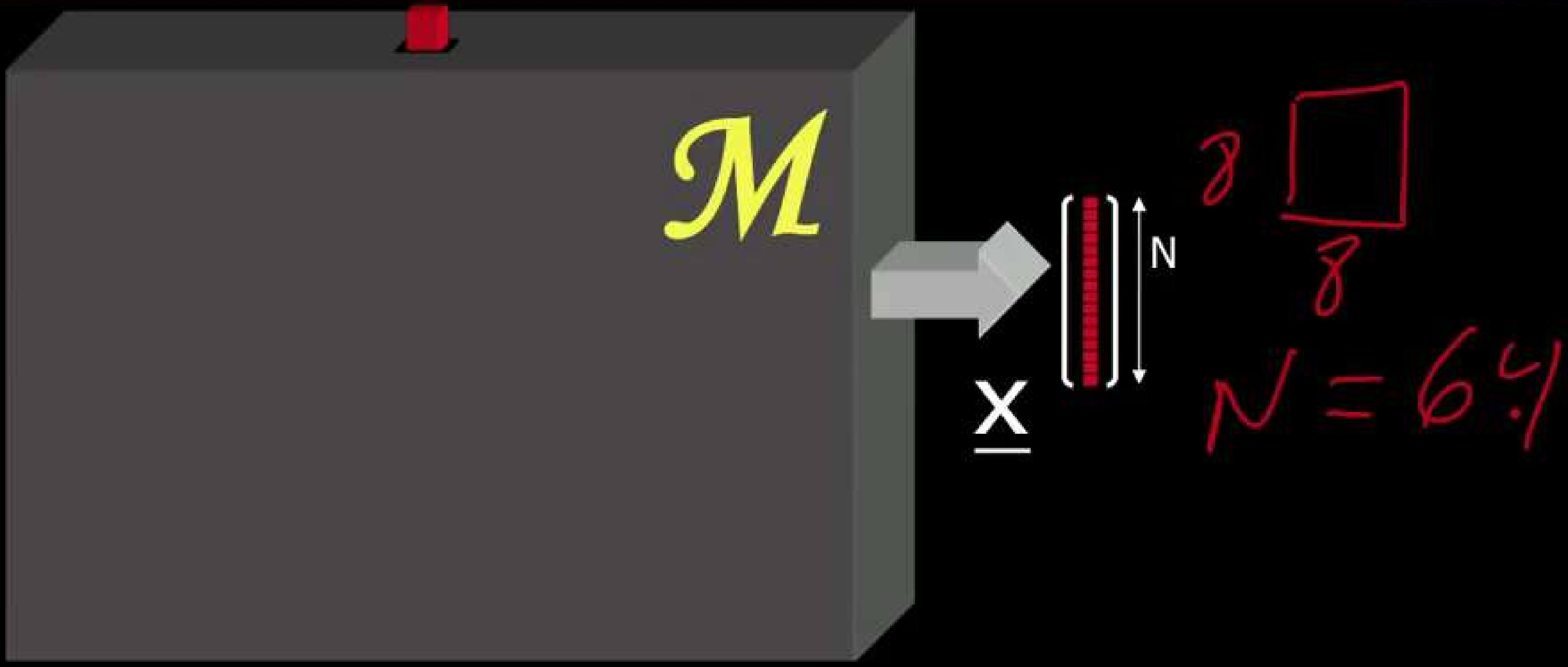
$$G(\underline{x}) = \lambda \|\underline{\alpha}\|_0^0 \quad \text{for} \quad \underline{x} = \mathbf{D}\underline{\alpha}$$

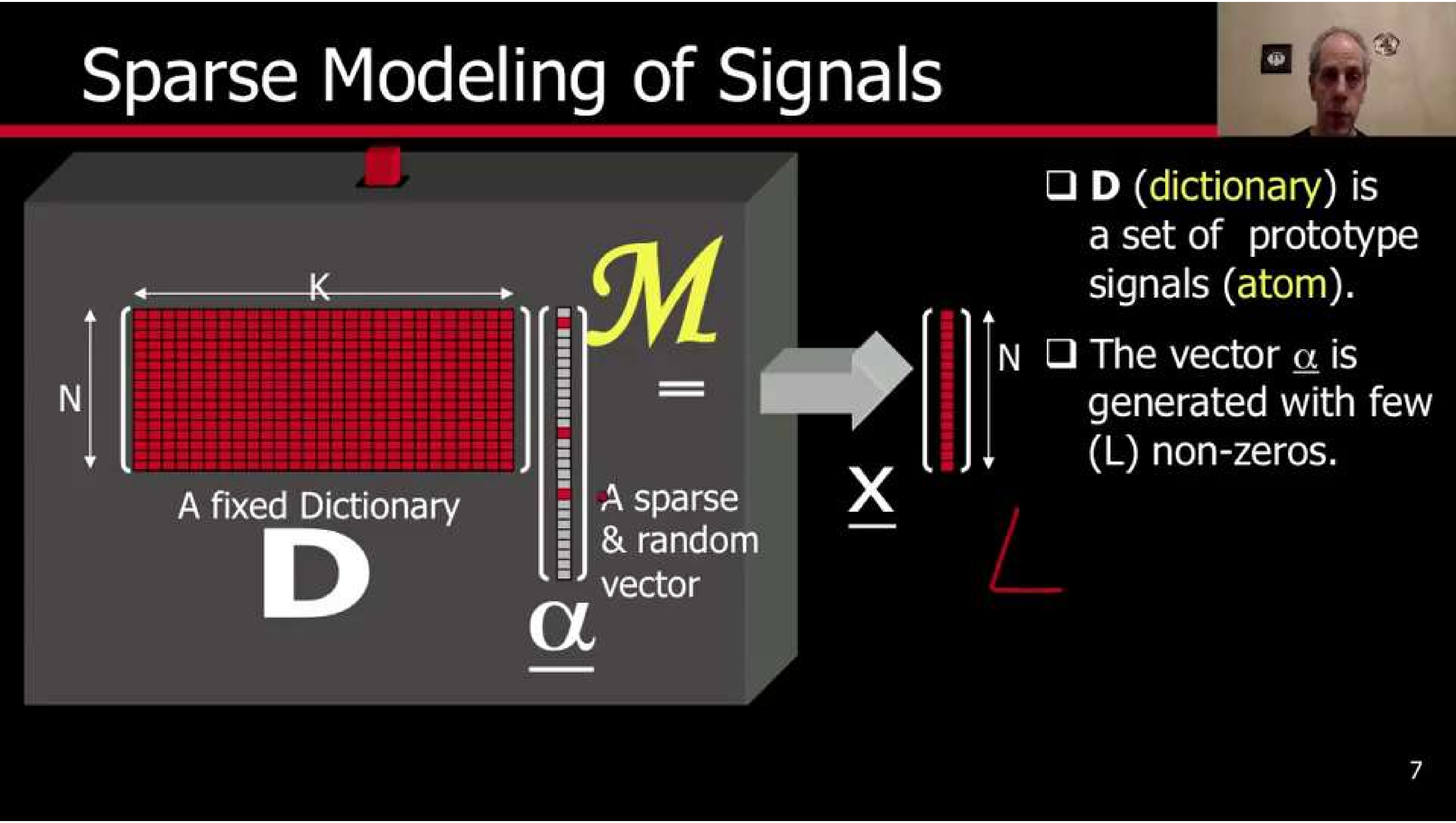
Sparse & Redundant

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

Sparse Modeling of Signals







Sparse Modeling of Signals

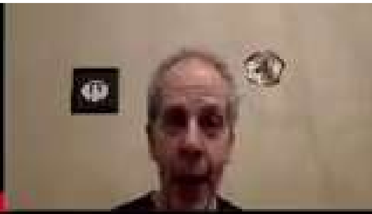
A fixed Dictionary

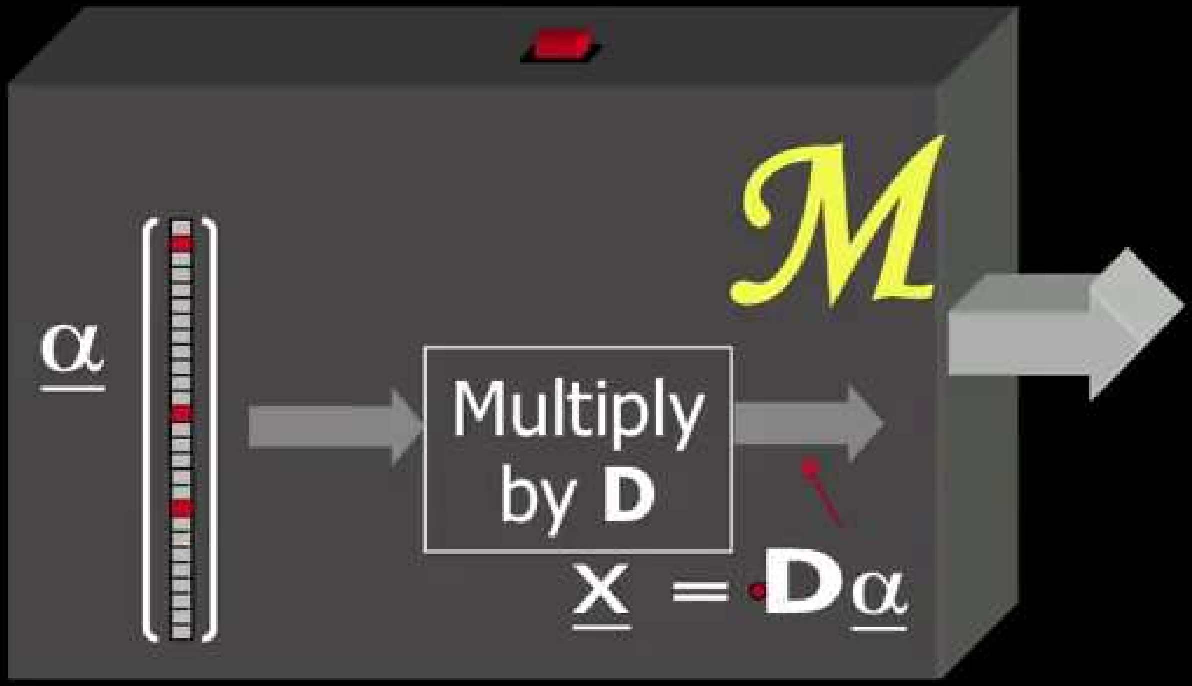
D

A sparse & random vector

- ❑ **D** (dictionary) is a set of prototype signals (atom).
- ❑ The vector $\underline{\alpha}$ is generated with few (L) non-zeros.

Sparseland Signals are Special





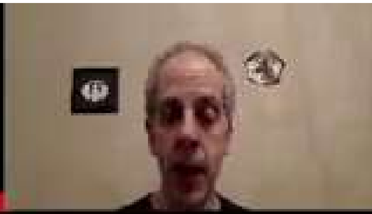
Multiply by **D**

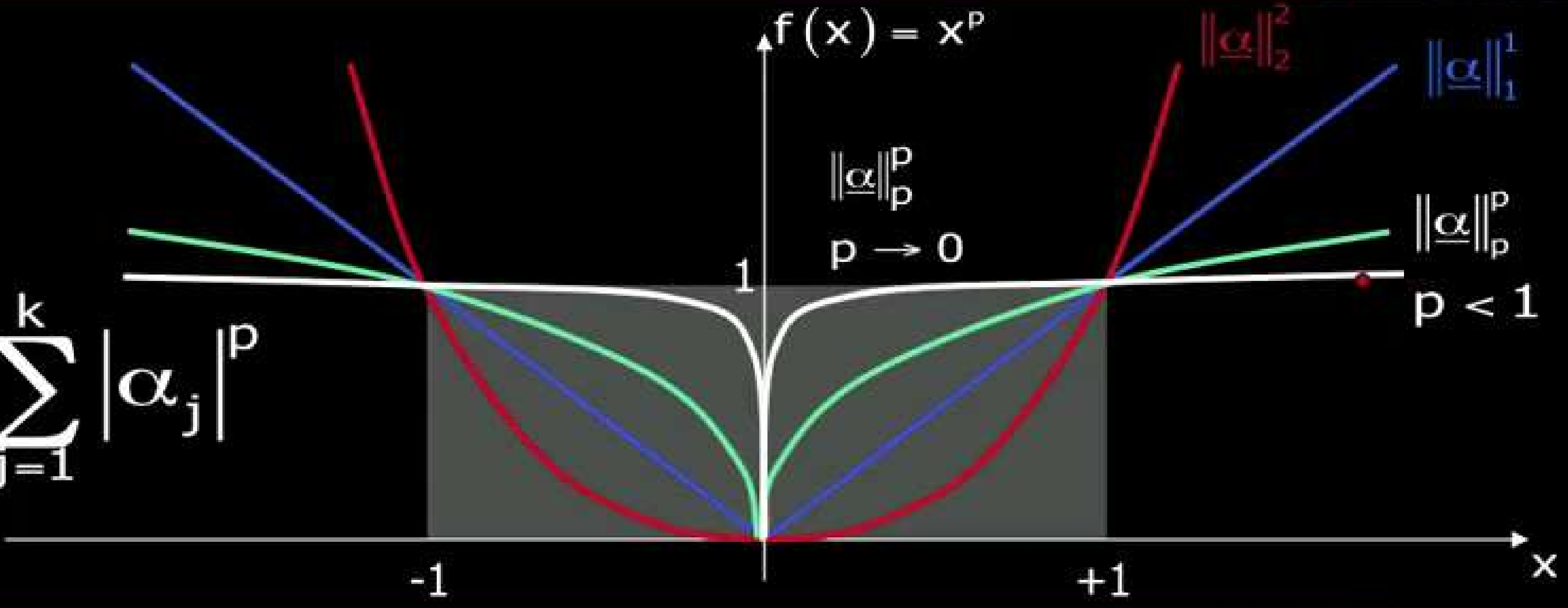
$\underline{x} = \mathbf{D}\underline{\alpha}$

- **Simple:** Every generated signal is built as a linear combination of **few** atoms from our dictionary **D**
- **Rich:** A general model: the obtained signals are a union of many low-dimensional spaces.
- **Familiar:** We have been using this model in other context for a while now

JPEG

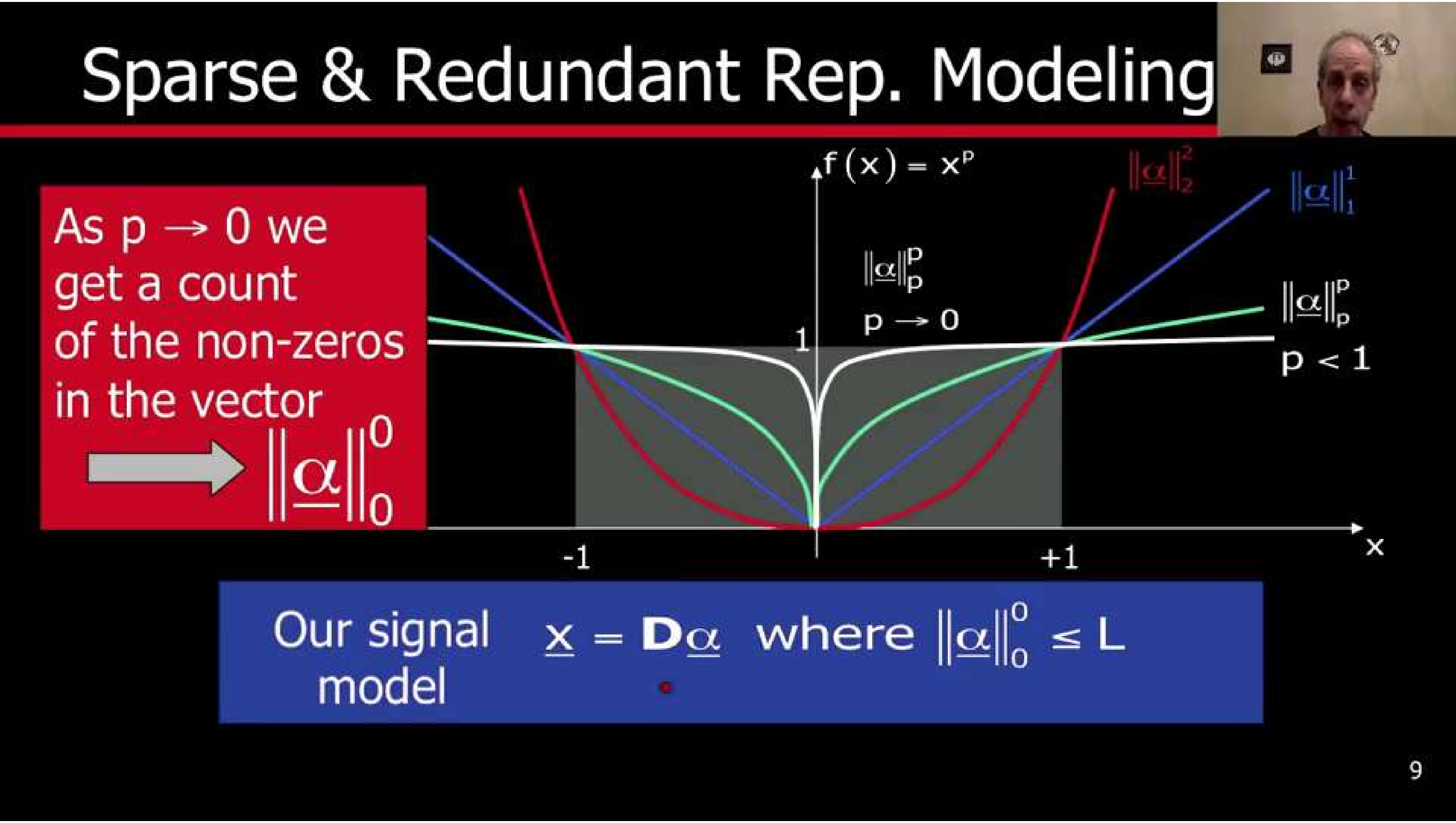
Sparse & Redundant Rep. Modeling





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

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

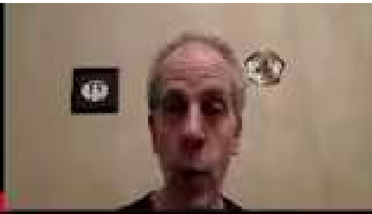


Sparse & Redundant Rep. Modeling

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

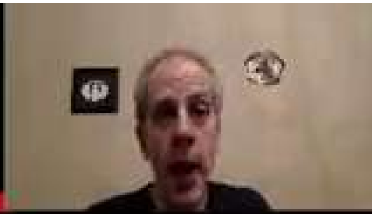
Our signal model $\underline{x} = \mathbf{D}\underline{\alpha}$ where $\|\underline{\alpha}\|_0^0 \le L$

Back to Our MAP Energy Function

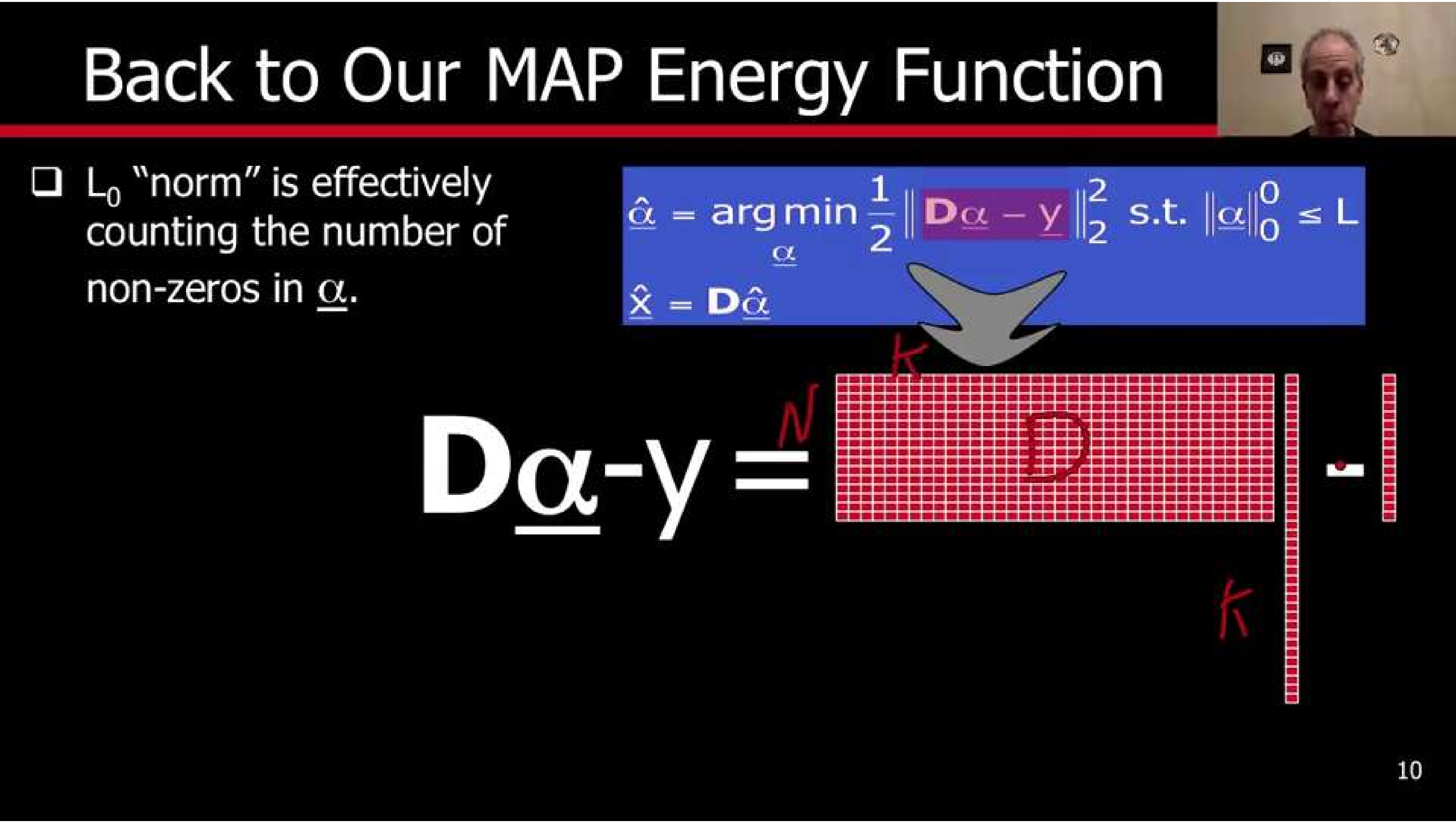


$$\frac{1}{2}\left\| \underline{x} - \underline{y} \right\|_2^2$$

Back to Our MAP Energy Function



$$\frac{1}{2}\left\| \mathbf{D}\underline{\alpha} - \underline{y} \right\|_2^2$$



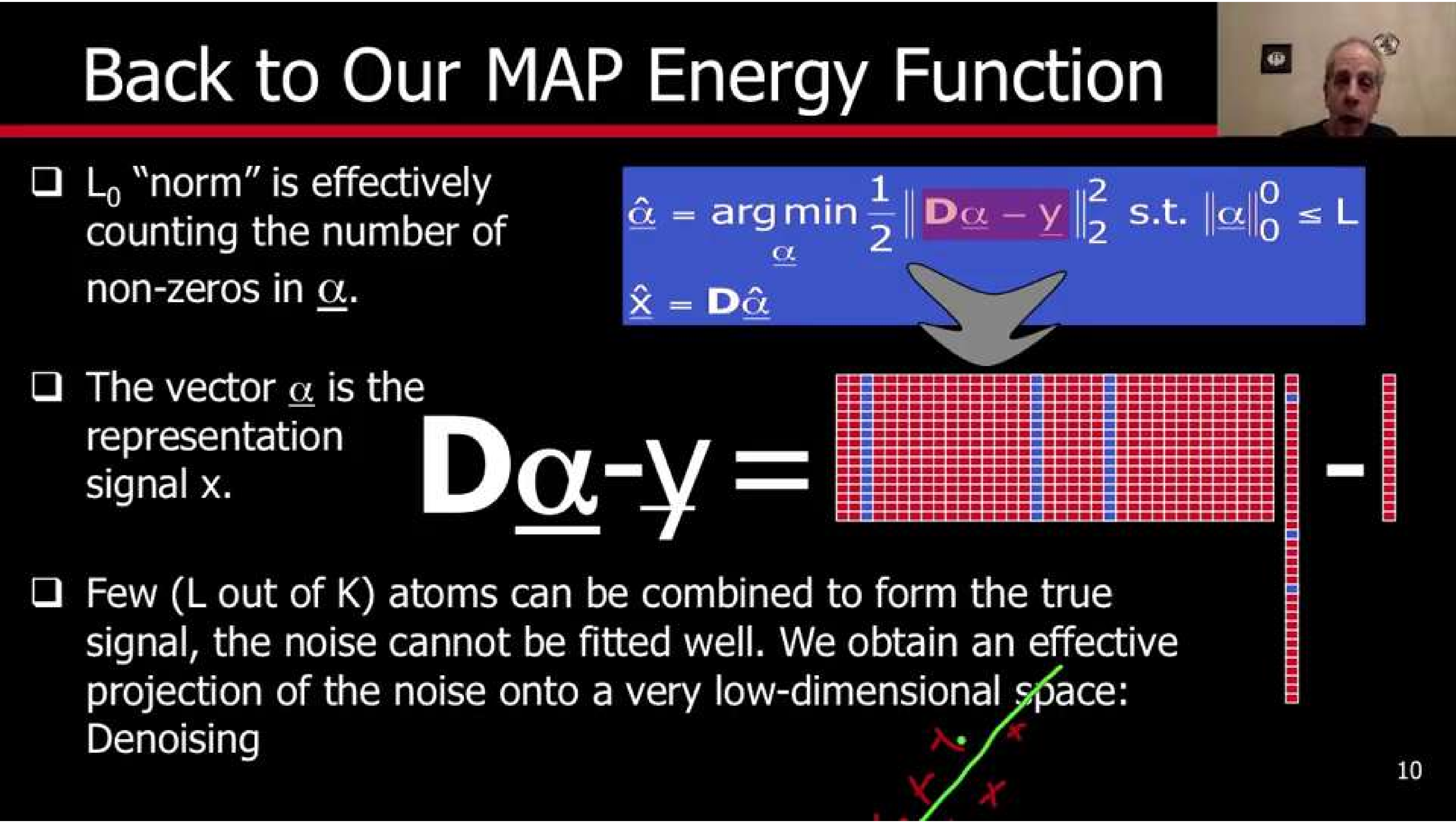
Back to Our MAP Energy Function

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

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

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

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



Back to Our MAP Energy Function

- ☐ L_0 "norm" is effectively counting the number of non-zeros in $\underline{\alpha}$.

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

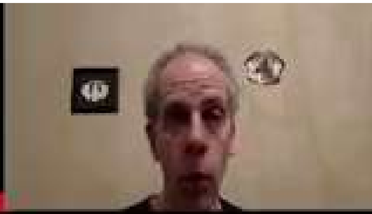
$$\hat{\underline{x}} = \mathbf{D}\hat{\underline{\alpha}}$$

- ☐ The vector $\underline{\alpha}$ is the representation signal x.

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

- ☐ Few (L out of K) atoms can be combined to form the true signal, the noise cannot be fitted well. We obtain an effective projection of the noise onto a very low-dimensional space: Denoising

Wait! There are Some Issues



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

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

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

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

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

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

To Summarize So Far …

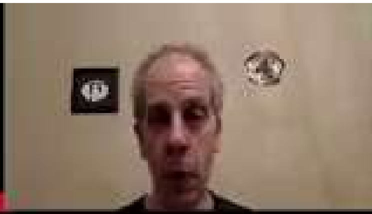


Image denoising (and many other problems in image processing) requires a model for the desired image

What do we do?

We proposed a model for signals/ images based on sparse and redundant representatio

Great! No?

There are some issues:

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

Sparse Modeling: Some Theory and Implementation

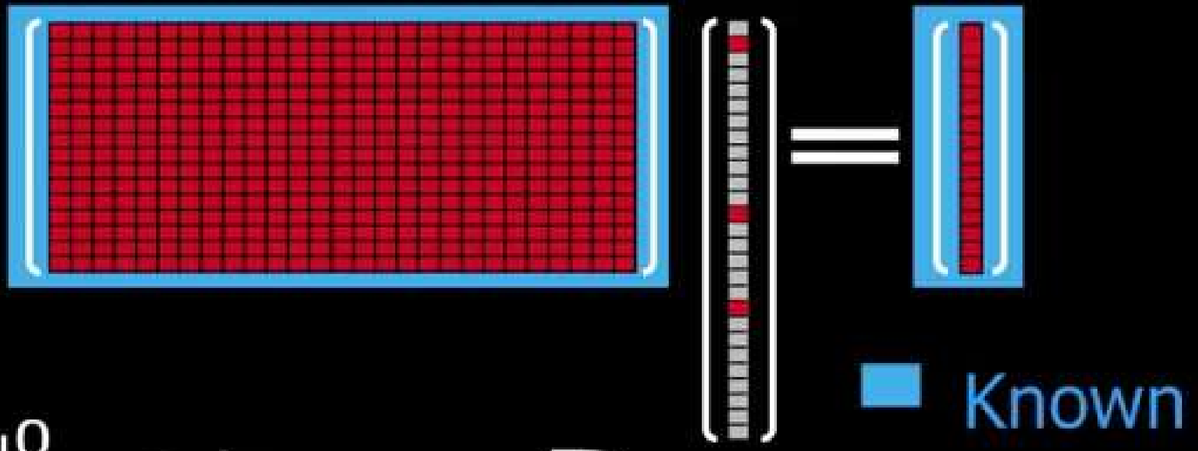
Image and Video Processing: From Mars to Hollywood with a Stop at the Hospital

Guillermo Sapiro

Duke
UNIVERSITY

Lets Start with the Noiseless Problem

Suppose we build a signal by the relation $\mathbf{D}\underline{\alpha} = \underline{x}$



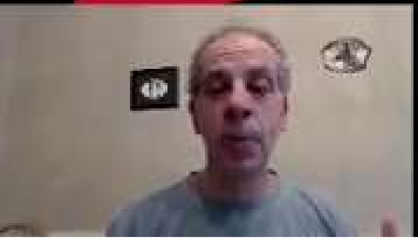
We aim to find the signal's representation:

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

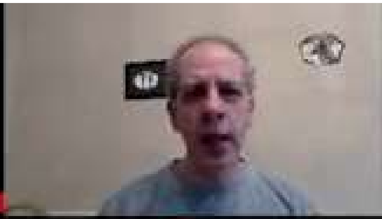
Uniqueness

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

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



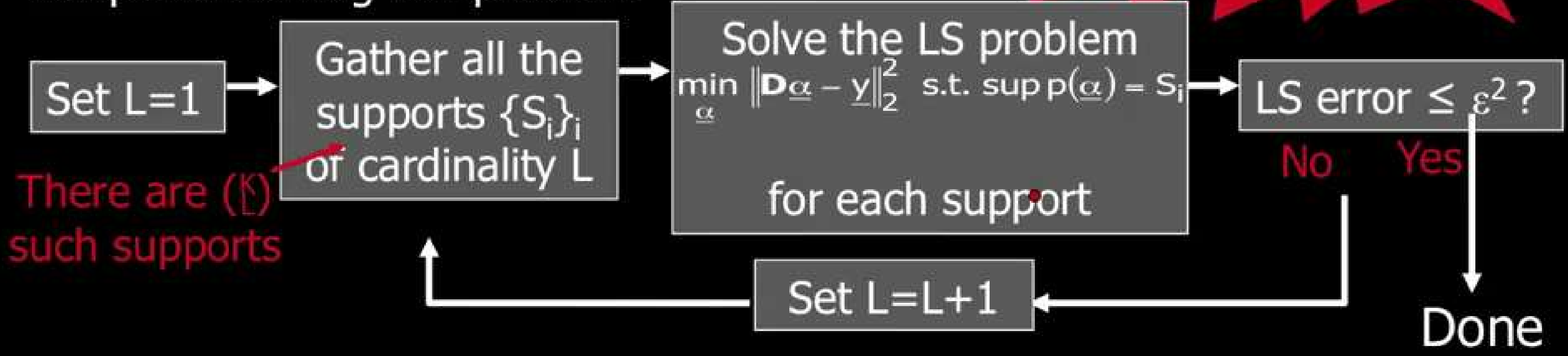
Our Goal



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

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

Recipe for solving this problem:



Set L=1

Gather all the supports $\{S_i\}_i$ of cardinality L

There are $\binom{K}{L}$ such supports

Solve the LS problem $\min_{\underline{\alpha}} \|\mathbf{D}\underline{\alpha} - \underline{y}\|_2^2 \text{ s.t. } \sup p(\underline{\alpha}) = S_i$ for each support

LS error ≤ ε^2 ?

No

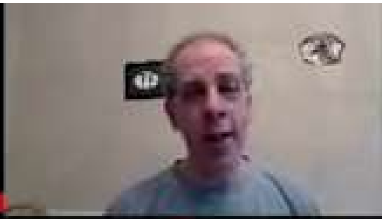
Yes

Set L=L+1

Done

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

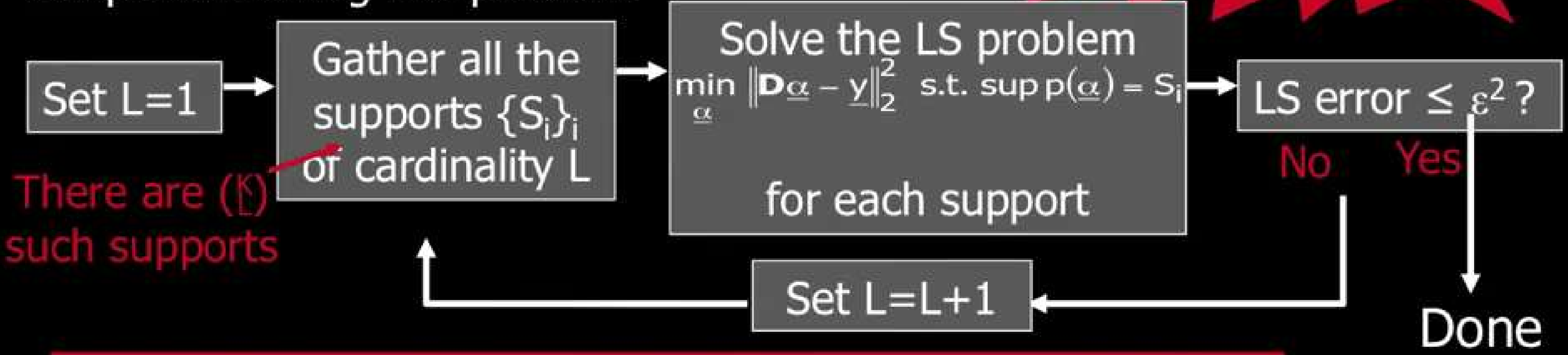
Our Goal



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

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

Recipe for solving this problem:



Set L=1

Gather all the supports $\{S_i\}_i$ of cardinality L

There are $\binom{K}{L}$ such supports

Solve the LS problem

$$\min_{\underline{\alpha}} \|\mathbf{D}\underline{\alpha} - \underline{y}\|_2^2 \text{ s.t. } \sup p(\underline{\alpha}) = S_i$$

for each support

LS error $\le \varepsilon^2$?

No

Yes

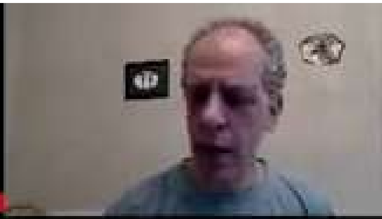
Set L=L+1

Done

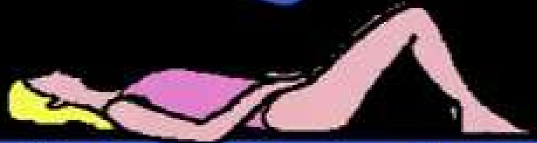
Assume: K=1000, L=10 (known!), 1 nano-sec per each LS

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

Lets Approximate



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



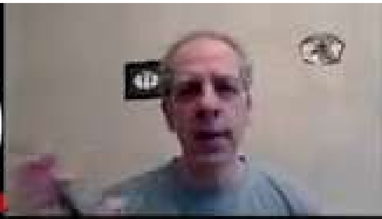
Relaxation methods

Smooth the L_0 and use continuous optimization techniques

Greedy methods

Build the solution one non-zero element at a time

Relaxation – The Basis Pursuit (BP)

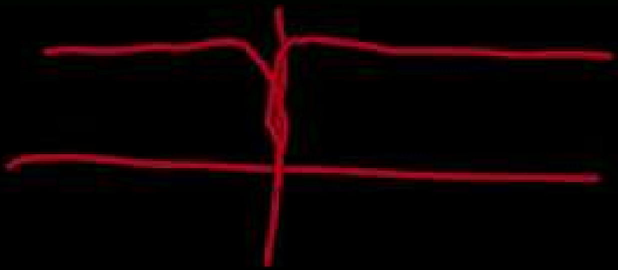


Instead of solving

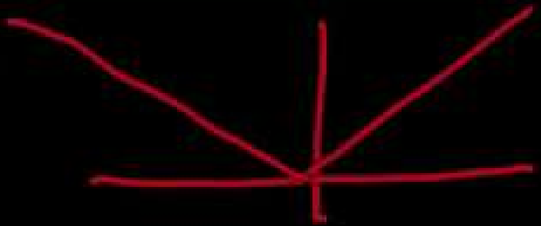
$$\underset{\underline{\alpha}}{\text{Min}}\ \|\underline{\alpha}\|_0^0\ \text{ s.t. }\ \|\mathbf{D}\underline{\alpha}-\underline{y}\|_2 \le \varepsilon$$

Solve Instead

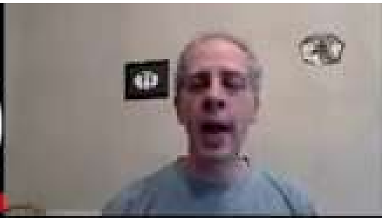
$$\underset{\underline{\alpha}}{\text{Min}}\ \|\underline{\alpha}\|_1\ \text{ s.t. }\ \|\mathbf{D}\underline{\alpha}-\underline{y}\|_2 \le \varepsilon$$



D, L



Relaxation – The Basis Pursuit (BP)



Instead of solving

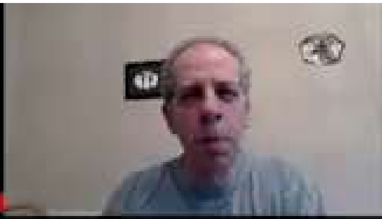
$$\underset{\underline{\alpha}}{\text{Min}}\ \|\underline{\alpha}\|_0^0 \ \text{ s.t. } \ \|\mathbf{D}\underline{\alpha}-\underline{y}\|_2 \le \varepsilon$$

Solve Instead

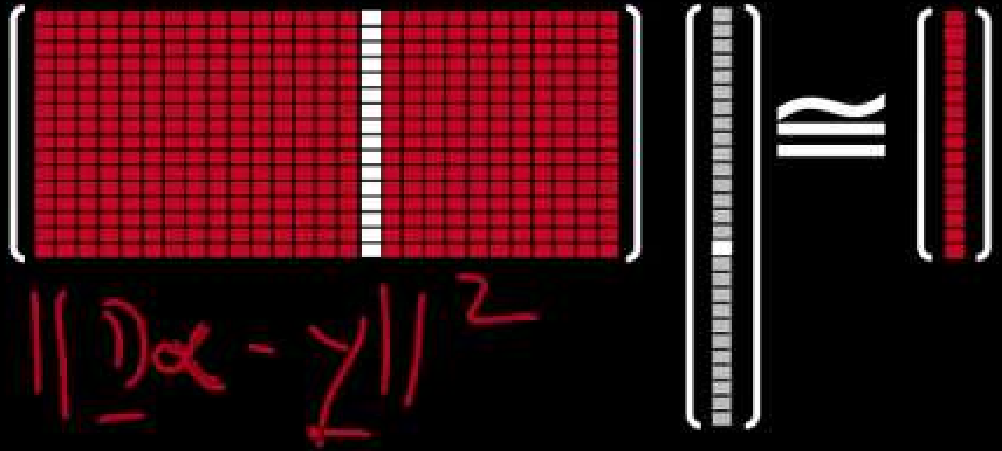
$$\underset{\underline{\alpha}}{\text{Min}}\ \|\underline{\alpha}\|_1 \ \text{ s.t. } \ \|\mathbf{D}\underline{\alpha}-\underline{y}\|_2 \le \varepsilon$$

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

Go Greedy: Matching Pursuit (MP)

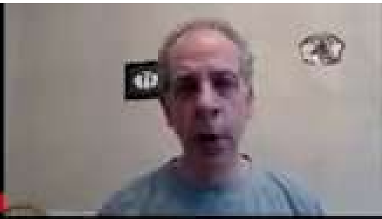


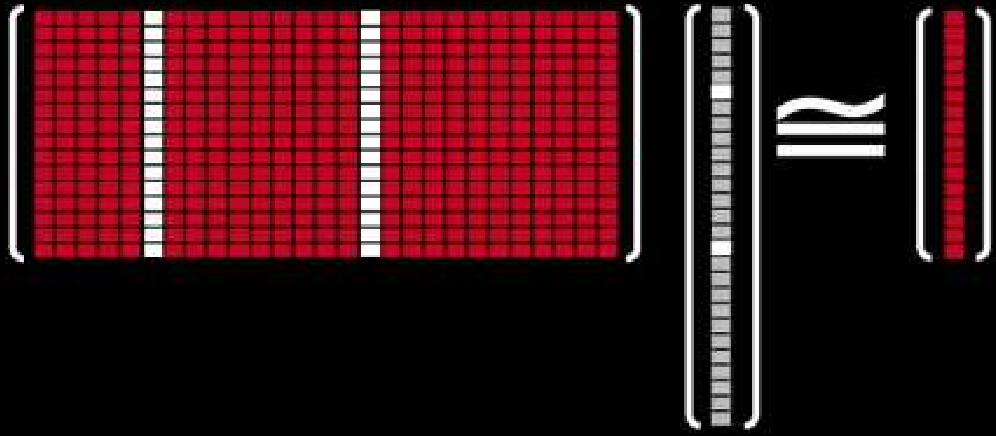
- ❑ The MP is one of the greedy algorithms that finds one atom at a time [Mallat & Zhang ('93)].
- ❑ Step 1: find the one atom that best matches the signal.



$\|\underline{D}\alpha - \underline{y}\|^2$

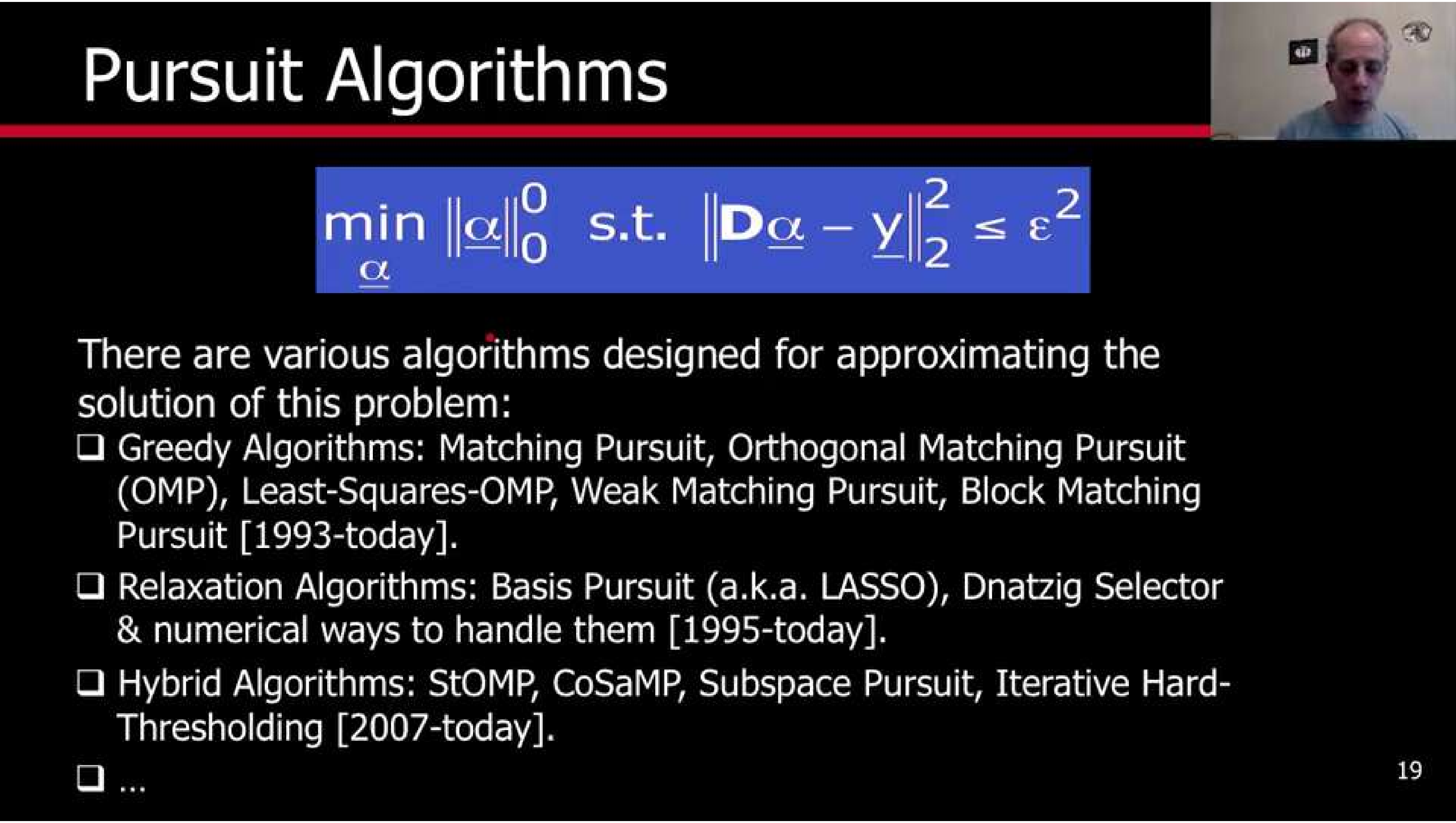
Go Greedy: Matching Pursuit (MP)





- The MP is one of the greedy algorithms that finds one atom at a time [Mallat & Zhang ('93)].
- Step 1: find the one atom that best matches the signal.
- Next steps: given the previously found atoms, find the next **one** to best fit the rsidual.
- The algorithm stops when the error $\|\mathbf{D}\underline{\alpha} - \underline{y}\|_2$ is below the destination threshold.

- The Orthogonal MP (OMP) is an improved version that re-evaluates the coefficients by Least-Squares after each round.



Pursuit Algorithms

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

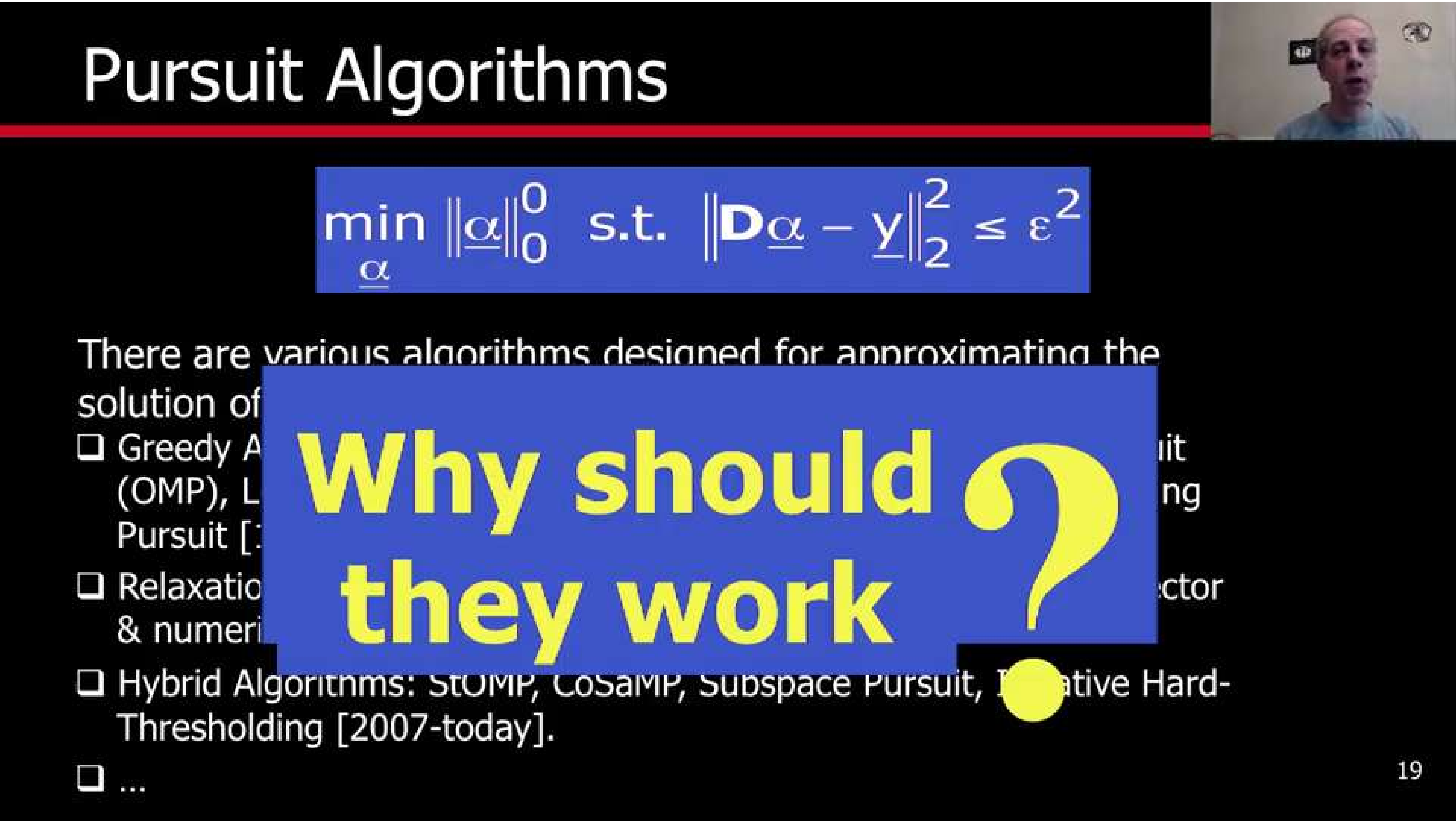
There are various algorithms designed for approximating the solution of this problem:

☐ Greedy Algorithms: Matching Pursuit, Orthogonal Matching Pursuit (OMP), Least-Squares-OMP, Weak Matching Pursuit, Block Matching Pursuit [1993-today].

☐ Relaxation Algorithms: Basis Pursuit (a.k.a. LASSO), Dnatzig Selector & numerical ways to handle them [1995-today].

☐ Hybrid Algorithms: StOMP, CoSaMP, Subspace Pursuit, Iterative Hard-Thresholding [2007-today].

☐ …



Pursuit Algorithms

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

There are various algorithms designed for approximating the solution of

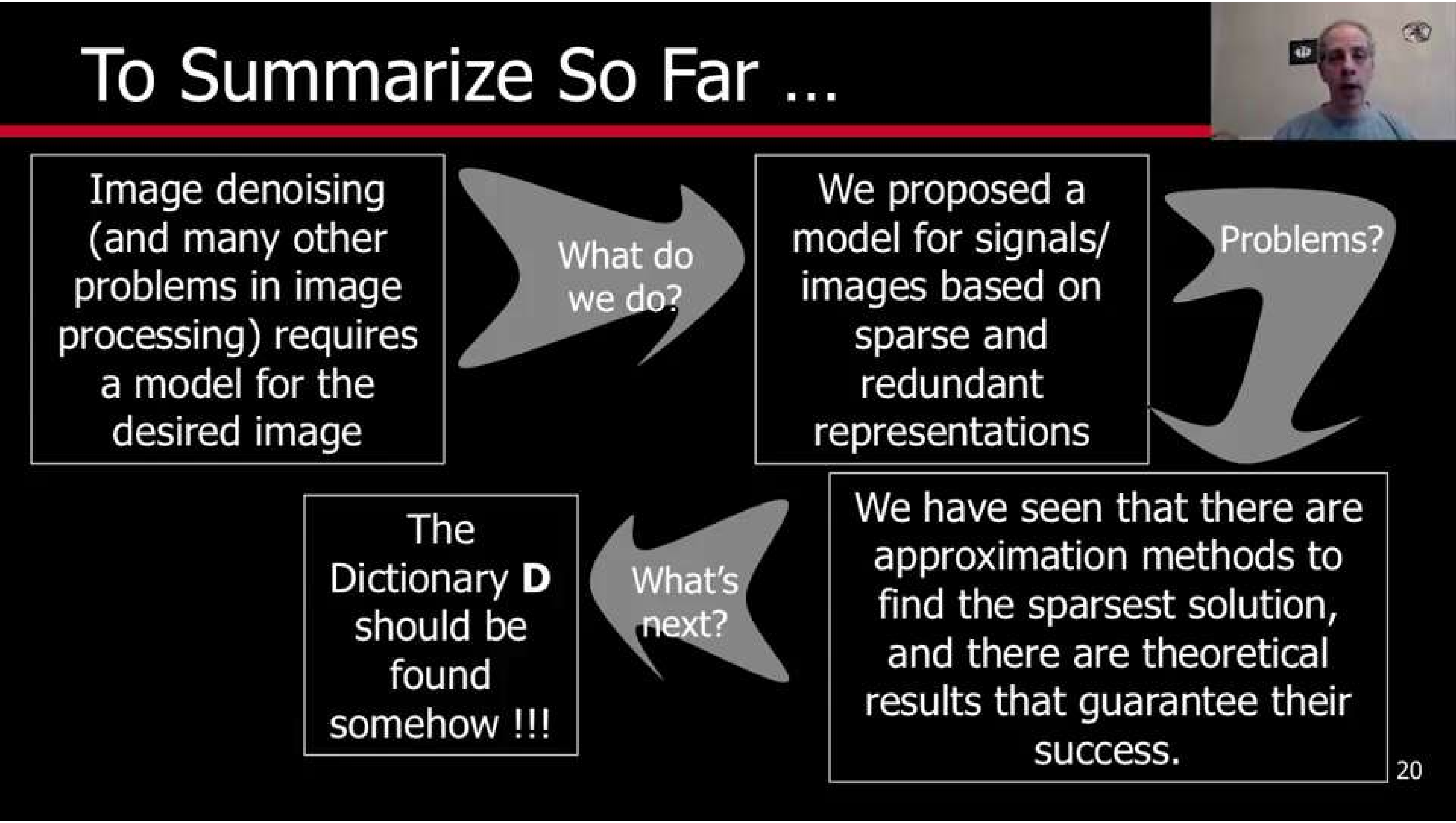
☐ Greedy A uit (OMP), L ng Pursuit [

☐ Relaxatio ector & numeri

☐ Hybrid Algorithms: StOMP, CoSaMP, Subspace Pursuit, I ative Hard-Thresholding [2007-today].

☐ …

Why should they work ?



To Summarize So Far …

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

What do we do?

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

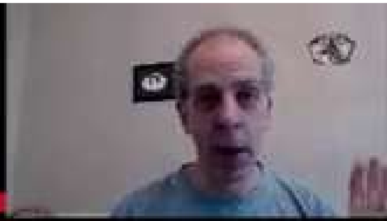
Problems?

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

What's next?

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

What Should **D** Be?



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

Our Assumption: Good-behaved Images
have a sparse representation

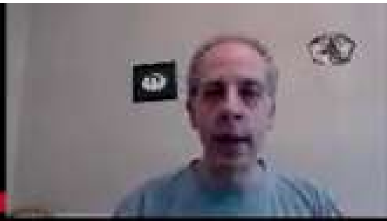
D should be chosen such that it sparsifies the representations

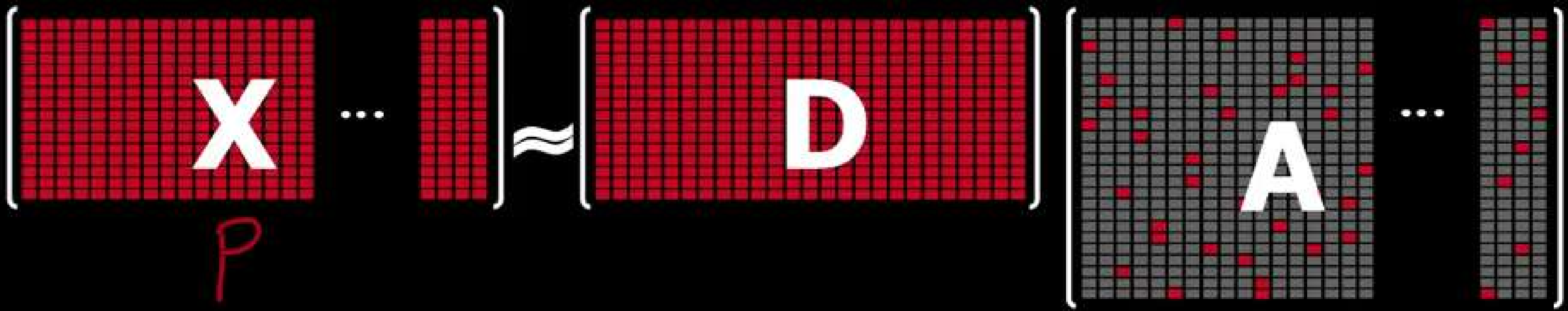
One approach to choose **D** is from a known set of transforms (Steerable wavelet, Curvelet, Contourlets, Bandlets, Shearlets …)

Training

Learning from **Examples**

Measure of Quality for **D**

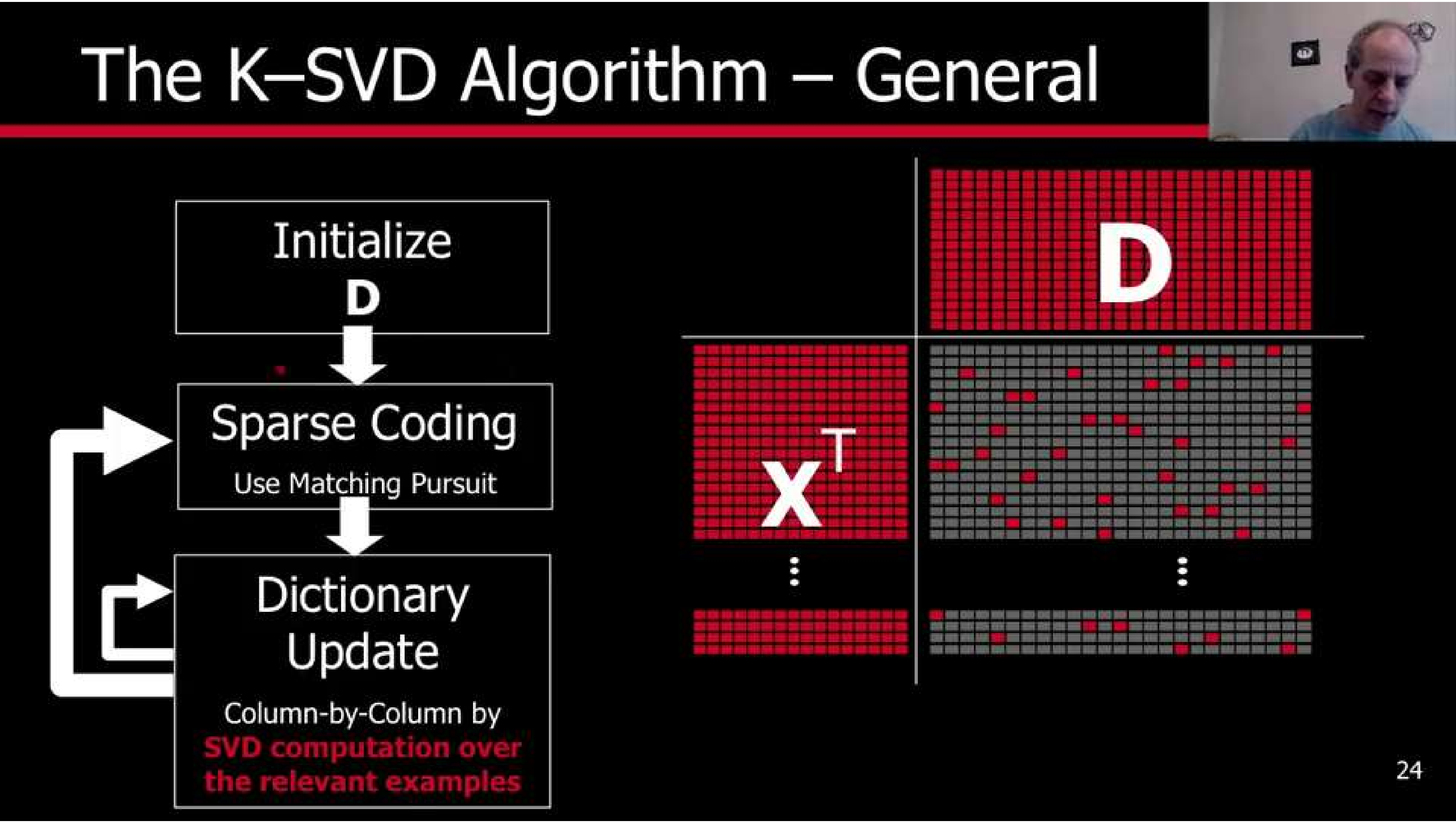




P

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

on-line.



The K–SVD Algorithm – General

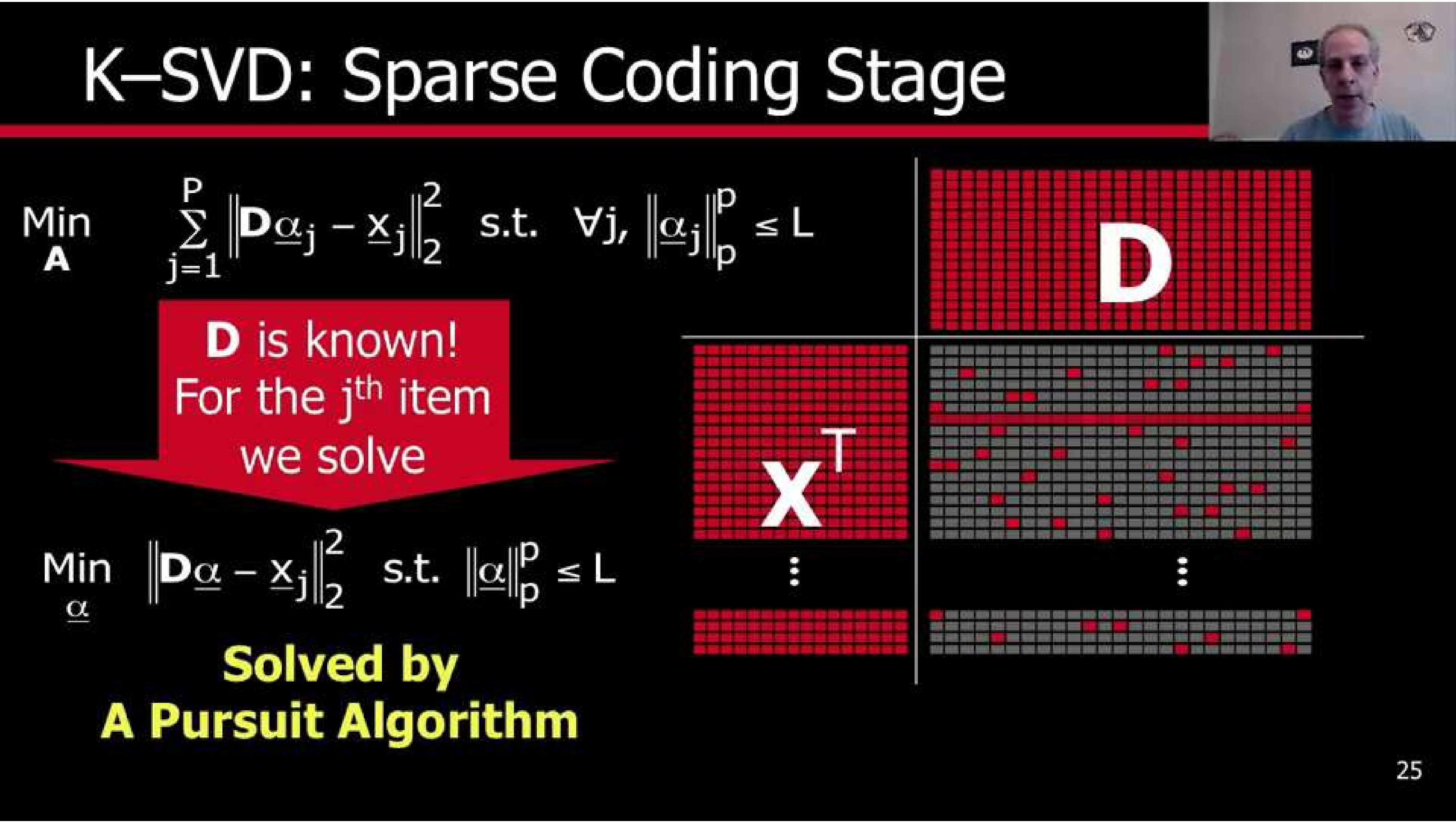
Initialize
D

Sparse Coding

Use Matching Pursuit

Dictionary Update

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



K–SVD: Sparse Coding Stage

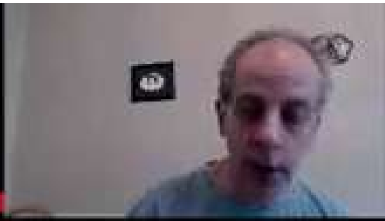
$$\underset{\mathbf{A}}{\text{Min}} \quad \sum_{j=1}^{P} \left\| \mathbf{D}\underline{\alpha}_j - \underline{x}_j \right\|_2^2 \quad \text{s.t.} \quad \forall j, \ \left\| \underline{\alpha}_j \right\|_p^p \le L$$

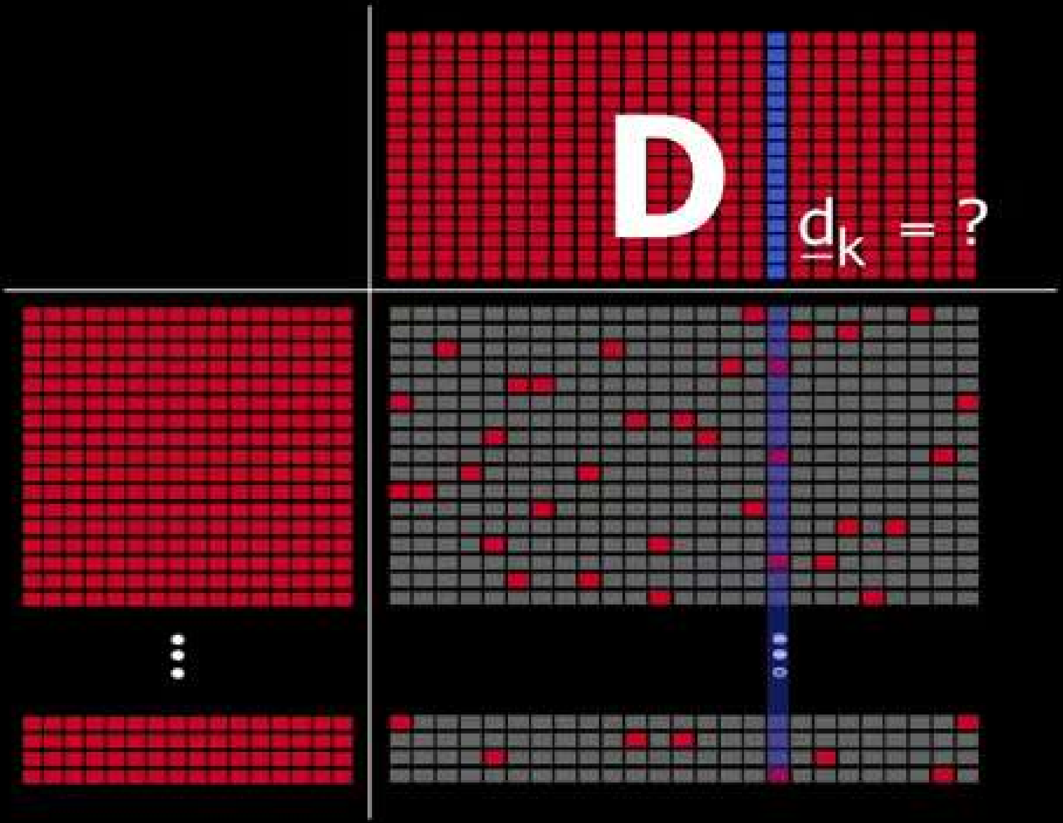
D is known!
For the jth item
we solve

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

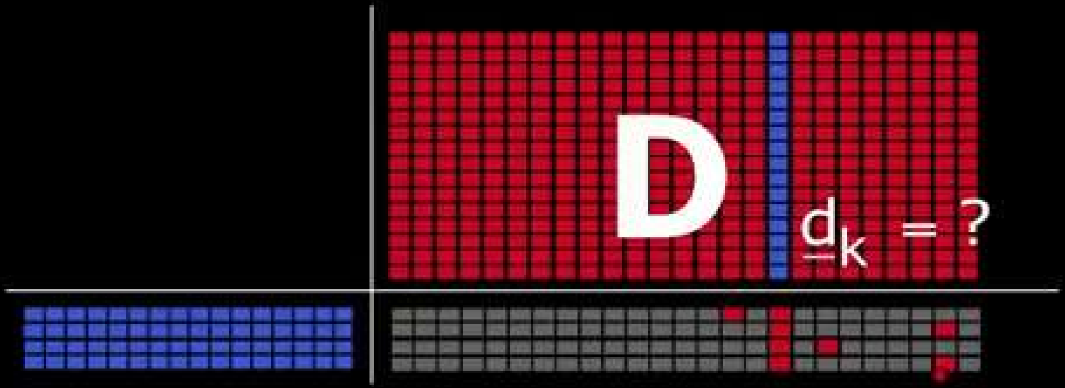
Solved by
A Pursuit Algorithm

K–SVD: Dictionary Update Stage

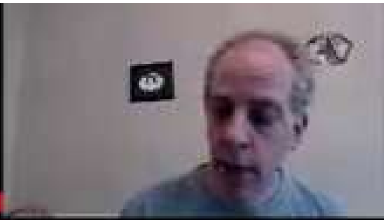




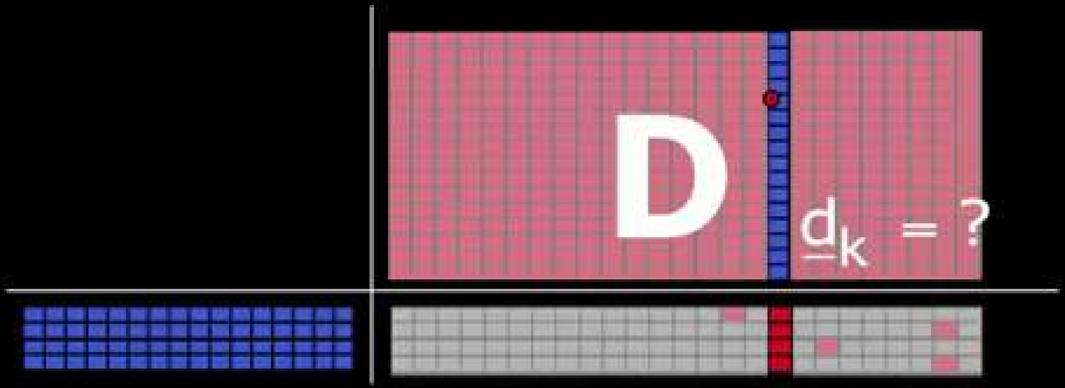
K–SVD: Dictionary Update Stage

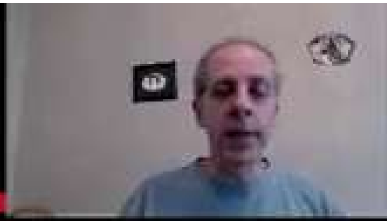


We refer only to the examples that use the column $\underline{d}_k$



K–SVD: Dictionary Update Stage

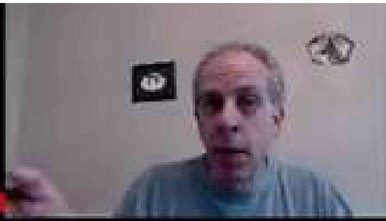


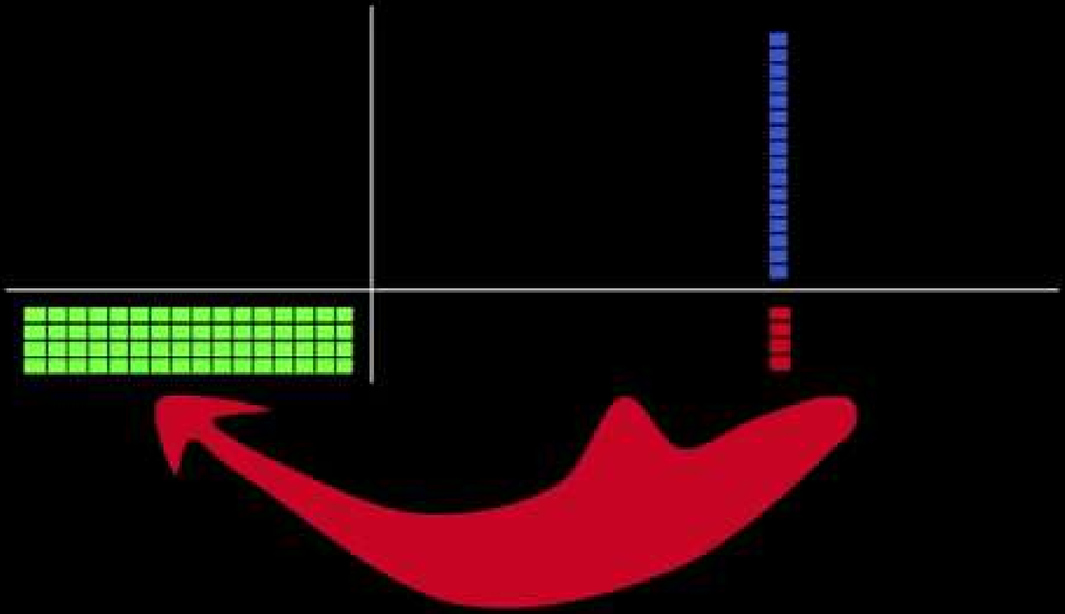


We refer only to the examples that use the column $\underline{d}_k$

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

K–SVD: Dictionary Update Stage

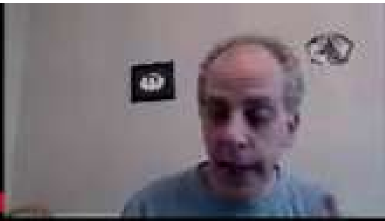




We refer only to the examples that use the column $\underline{d}_k$

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

K–SVD: Dictionary Update Stage

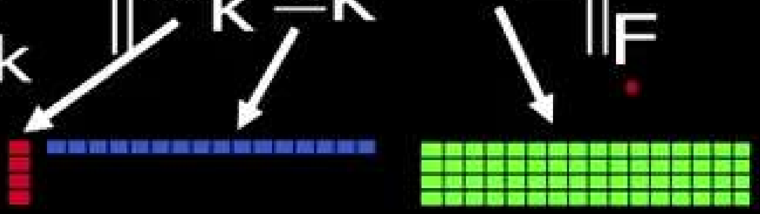


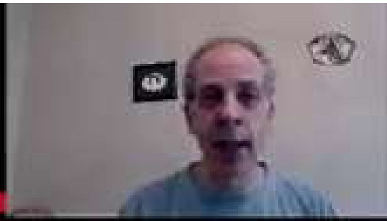
We refer only to the examples that use the column $\underline{d}_k$

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

We should solve:

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





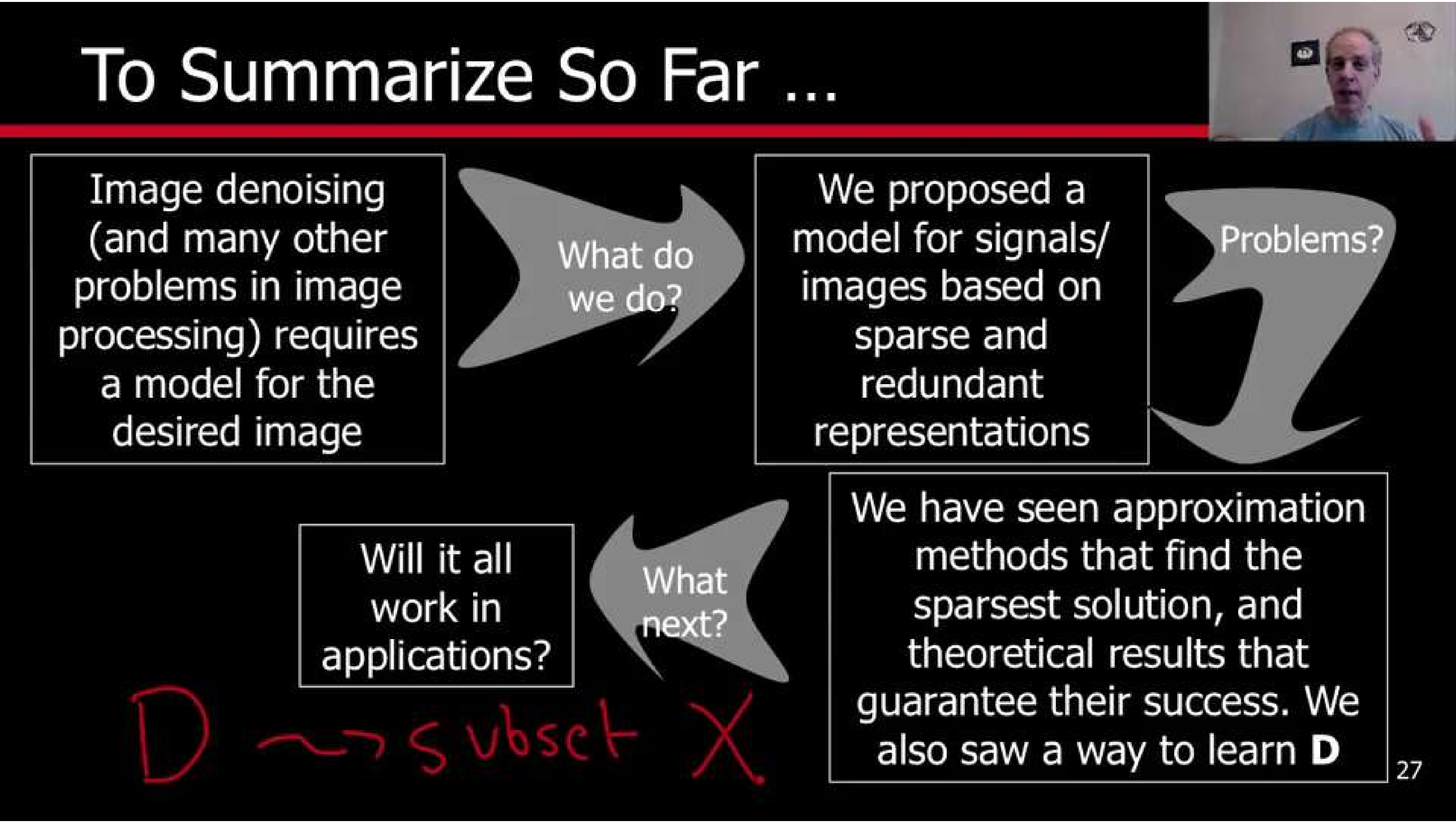
K–SVD: Dictionary Update Stage

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

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

We should solve:

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



To Summarize So Far …

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

What do we do?

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

Problems?

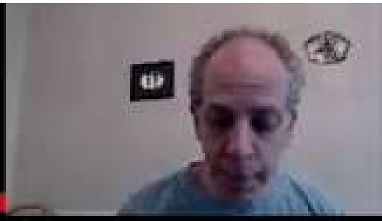
We have seen approximation methods that find the sparsest solution, and theoretical results that guarantee their success. We also saw a way to learn **D**

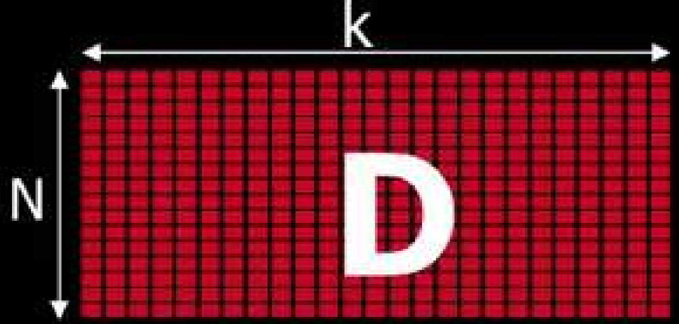
What next?

Will it all work in applications?

D ~→ subsct X

From Local to Global Treatment

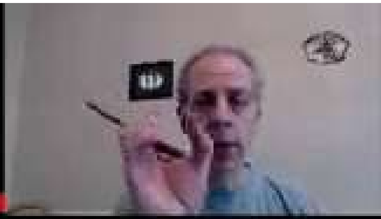


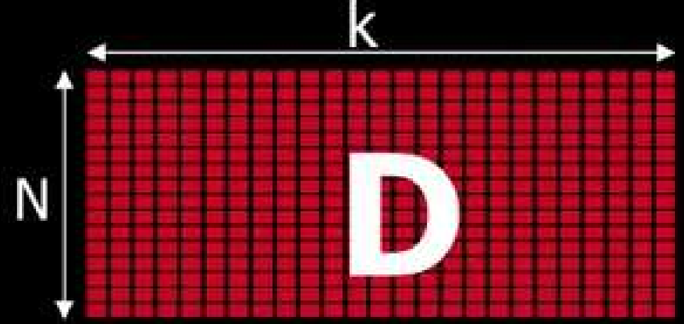


$$\underline{\hat{x}} = \underset{\underline{x},\{\underline{\alpha}_{ij}\}_{ij}}{\text{ArgMin}} \ \frac{1}{2}\left\|\underline{x}-\underline{y}\right\|_2^2 + \mu\sum_{ij}\left\|\mathbf{R}_{ij}\underline{x} - \mathbf{D}\underline{\alpha}_{ij}\right\|_2^2$$

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

From Local to Global Treatment



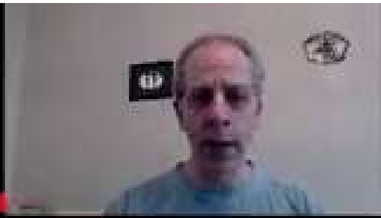


$$\hat{\underline{x}} = \underset{\underline{x},\{\underline{\alpha}_{ij}\}_{ij}}{\text{ArgMin}} \ \frac{1}{2}\left\|\underline{x}-\underline{y}\right\|_2^2 + \mu\sum_{ij}\left\|\mathbf{R}_{ij}\underline{x}-\mathbf{D}\underline{\alpha}_{ij}\right\|_2^2 \quad \text{s.t.} \quad \left\|\underline{\alpha}_{ij}\right\|_0^0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

Extracts a patch in the ij location

Our prior

What Data to Train On?



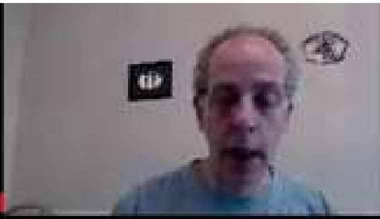
Option 1:

☐ Use a database of images

Option 2:

☐ Use the corrupted image itself !!

K-SVD Image Denoising



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

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

Compute α_{ij} per patch

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

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

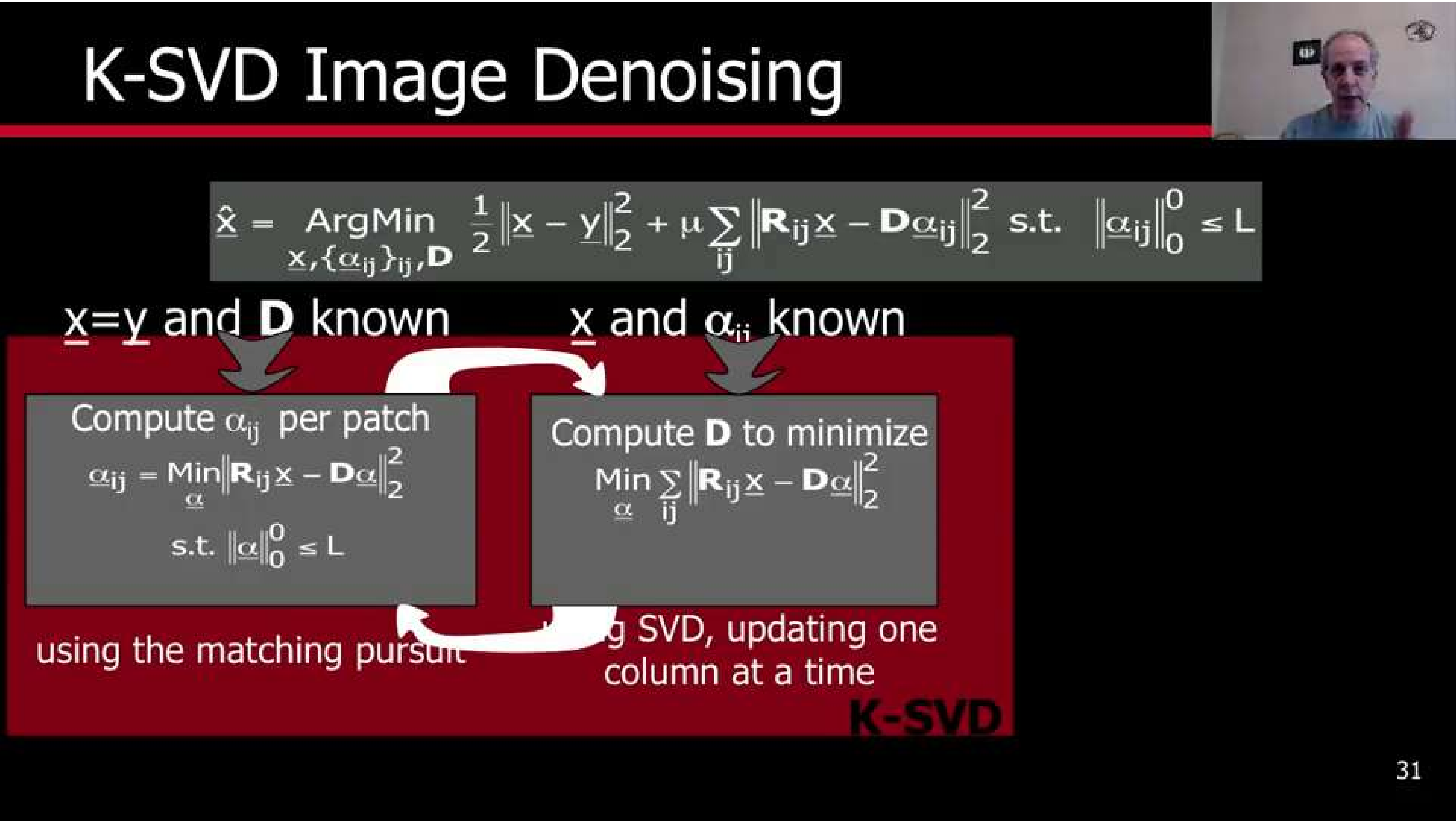
using the matching pursuit

$\underline{x}$ 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 Image Denoising

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

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

$\underline{x}$ and α_{ij} known

Compute α_{ij} per patch

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

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

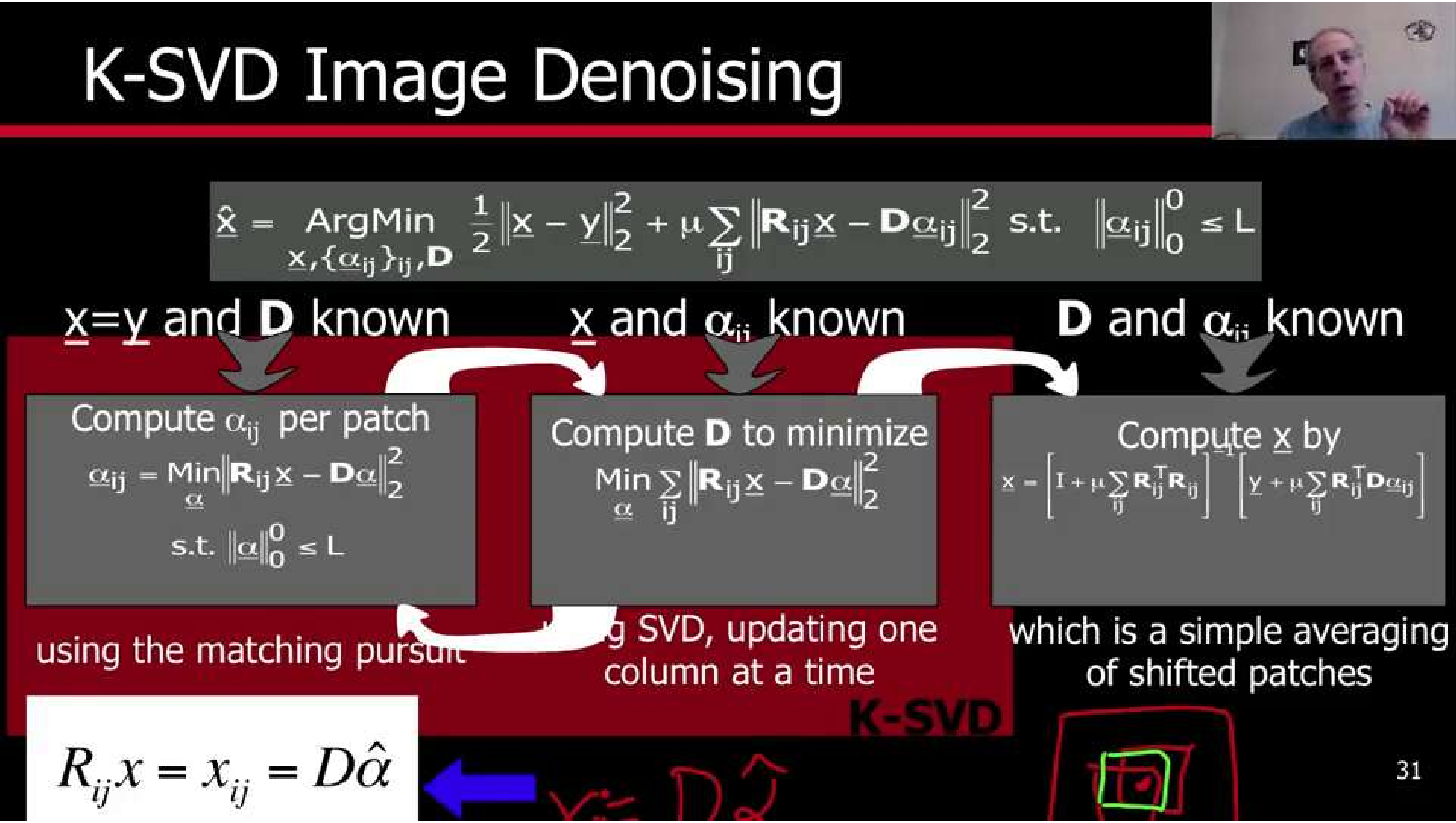
using the matching pursuit

Compute **D** to minimize

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

g SVD, updating one column at a time

K-SVD



K-SVD Image Denoising

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

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

Compute α_{ij} per patch

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

using the matching pursuit

$\underline{x}$ and α_{ij} known

Compute **D** to minimize

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

g SVD, updating one column at a time

D and α_{ij} known

Compute $\underline{x}$ by

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

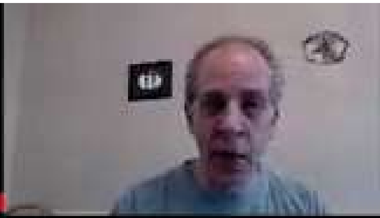
which is a simple averaging of shifted patches

K-SVD

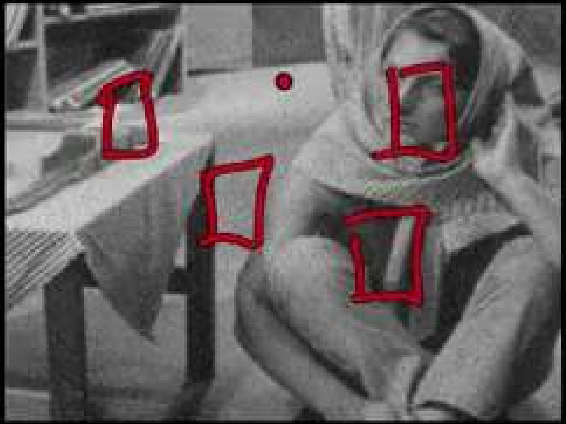
$$R_{ij}x = x_{ij} = D\hat{\alpha}$$

$x_i = D\hat{\alpha}$

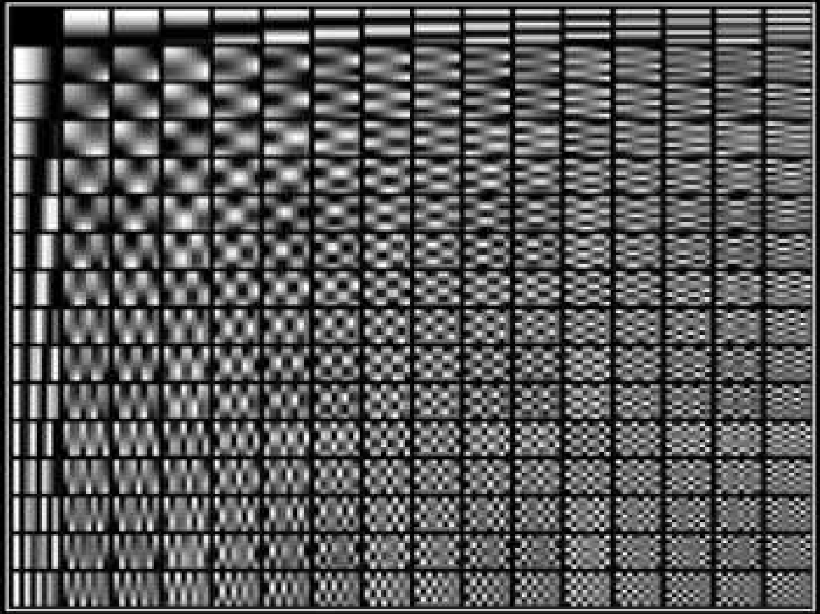
Image Denoising (Gray)



Source



Noisy image
$\sigma = 20$



Initial dictionary (overcomplete DCT) 64×256

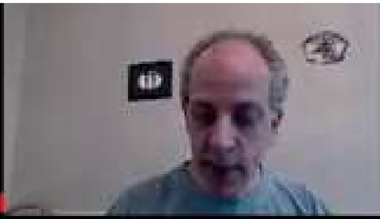
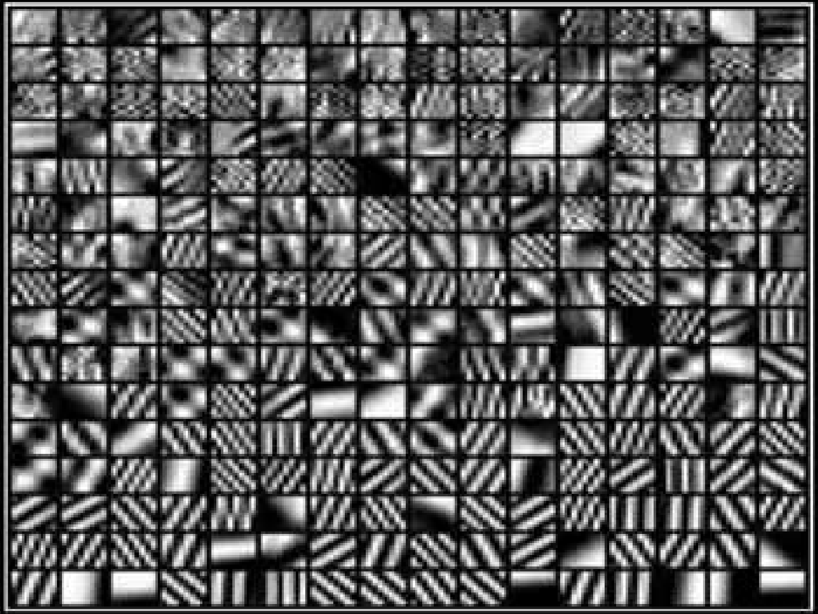


Image Denoising (Gray)

Source

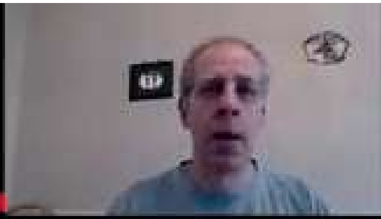
Result 30.829dB

Noisy image
$\sigma = 20$



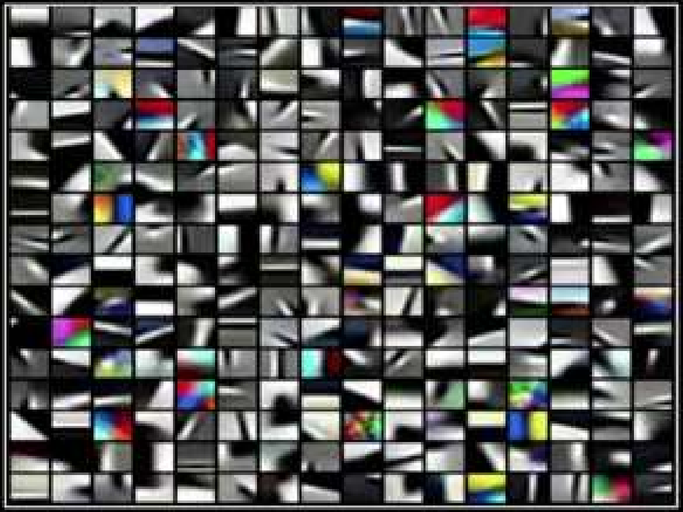
The obtained dictionary after
10 iterations

Denoising (Color)

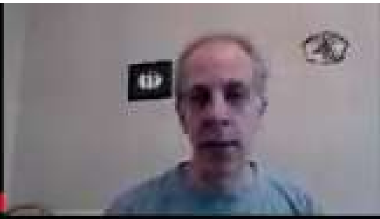


☐ When turning to handle color images, the main difficulty is in defining the relation between the color layers – R, G, and B.

☐ The solution with the above algorithm is simple – consider 3D patches or 8-by-8 with the 3 color layers, and the dictionary will detect the proper relations.



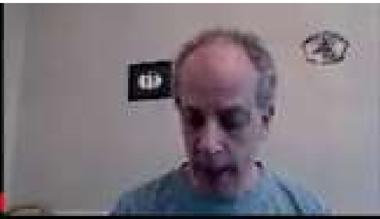
Denoising (Color)



Original

Noisy (20.43dB)

Result (30.75dB)



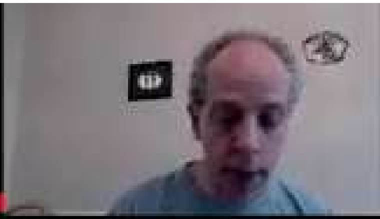
Denoising (Color)

Original

Noisy (12.77dB)

Result (29.87dB)

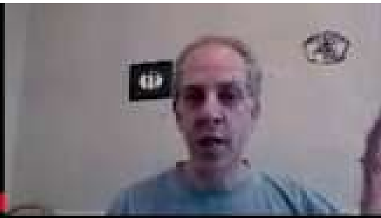
Inpainting



Original

80% missing

Result



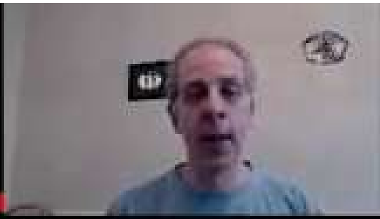
Inpainting

Original

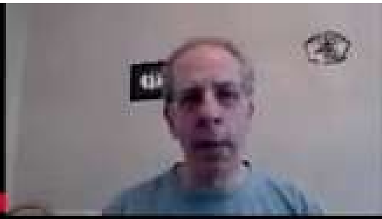
80% missing

Result

Inpainting

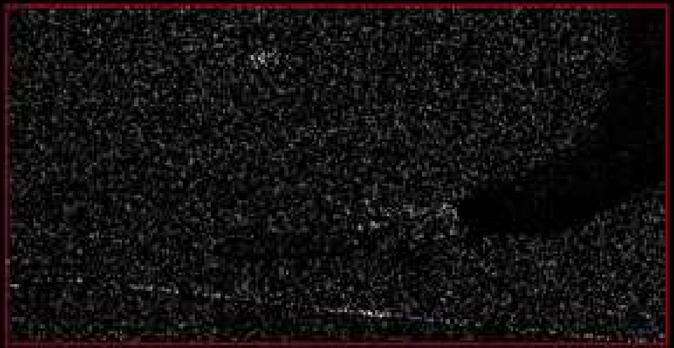



Since 1699, when French explorers
landed at the great bend of the
Mississippi River and celebrated
the first Mardi Gras in North America,
New Orleans has brewed a fascinating
melange of cultures. It was French,
then Spanish, then French again, then
sold to the United States. Through all
these years,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

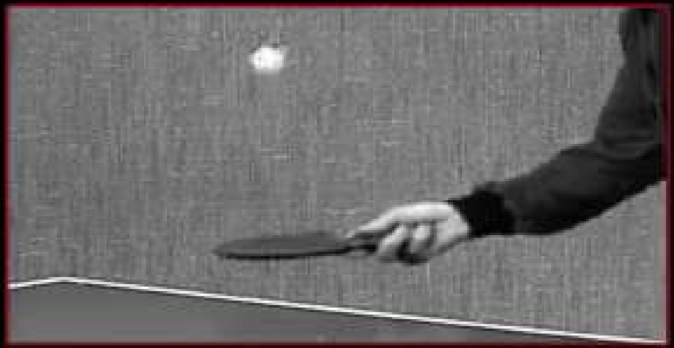


Video Inpainting

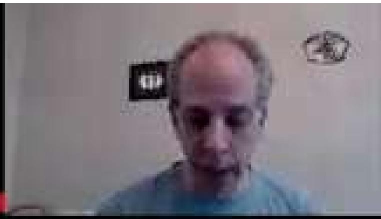
Original



80% missing

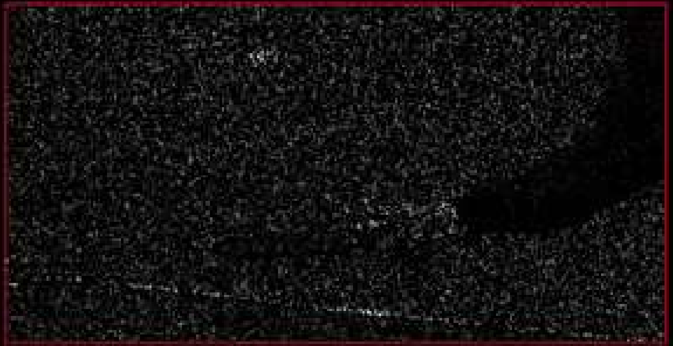


Result

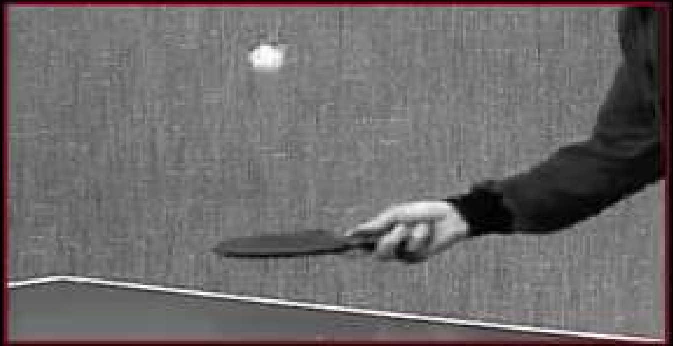


Video Inpainting

~~Original~~

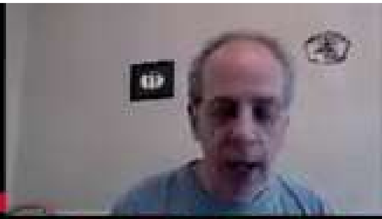


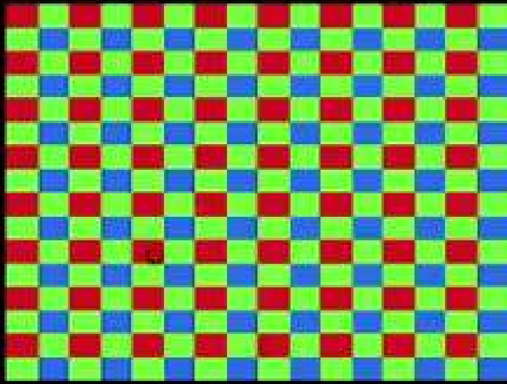
80% missing



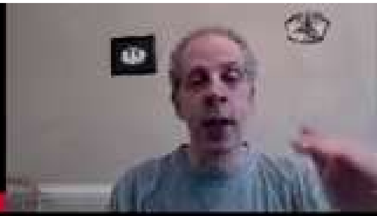
Result

Demosaicing

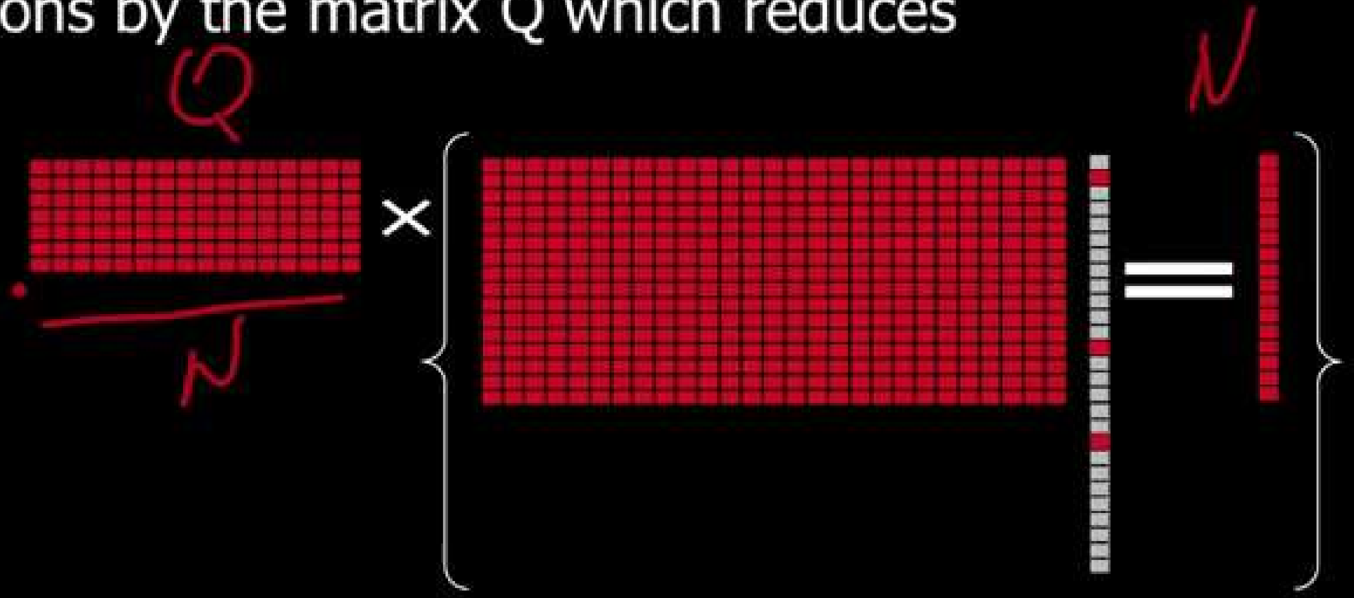




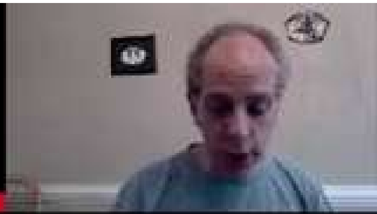
Side Note: Compressed-Sensing



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

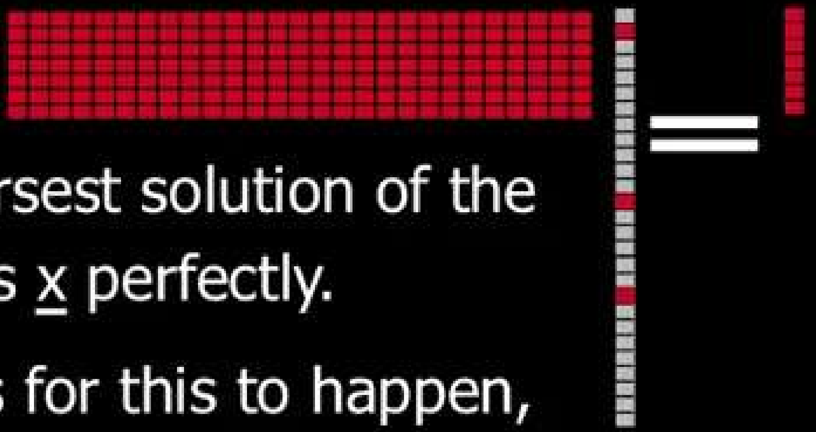


Side Note: Compressed-Sensing



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

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



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

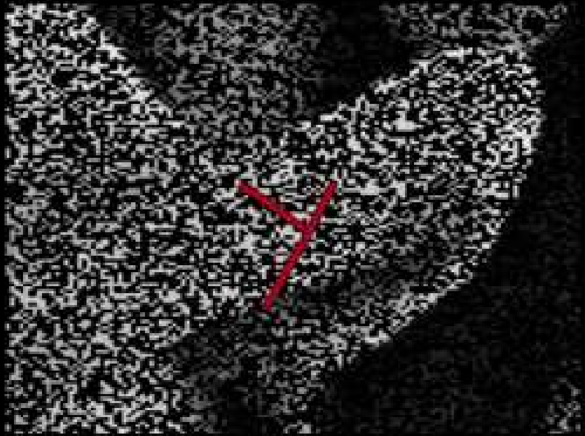
Inverse Problems

$$\underline{\mathbf{y}} = \mathbf{U}\underline{\mathbf{f}} + \mathbf{w}$$

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma^2 Id)$$

$\mathbf{f}$

Inpainting

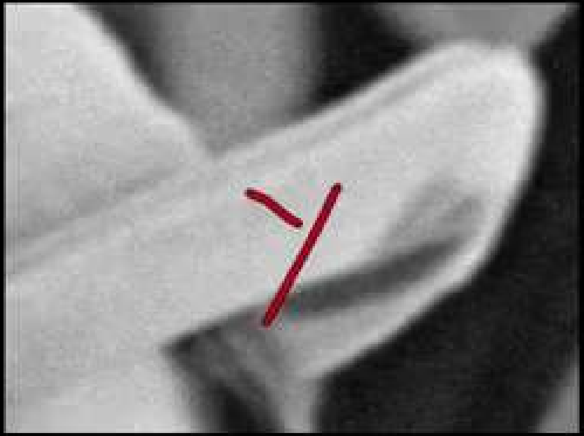


U masking

Deblurring

Zooming

U subsampling



U convolution

$\mathbf{w} \sim \mathcal{N}(\mathbf{0}, \sigma^2 Id)$

Gaussian Mixture Models of Patches

$$\mathbf{y}_i = \mathbf{U}_i \mathbf{f}_i + \mathbf{w}_i$$

where

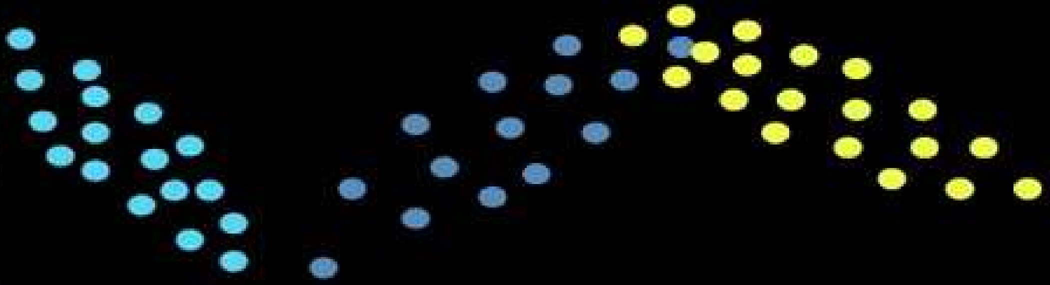
$$\mathbf{w}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 Id)$$

8x8

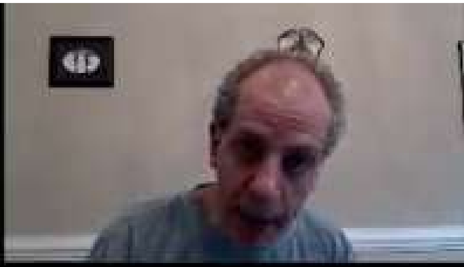
64

- *K* Gaussian distributions or PCAs $\{\mathcal{N}(\mu_k, \Sigma_k)\}_{1 \leq k \leq K}$
- $\mathbf{f}_i \sim \mathcal{N}(\mu_k, \Sigma_k)$

K=10



Gaussian Mixture Models of Patches



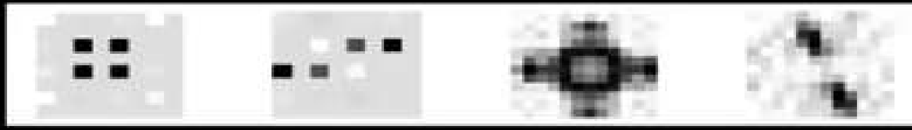
$$\mathbf{y}_i = \mathbf{U}_i\mathbf{f}_i + \mathbf{w}_i$$

where

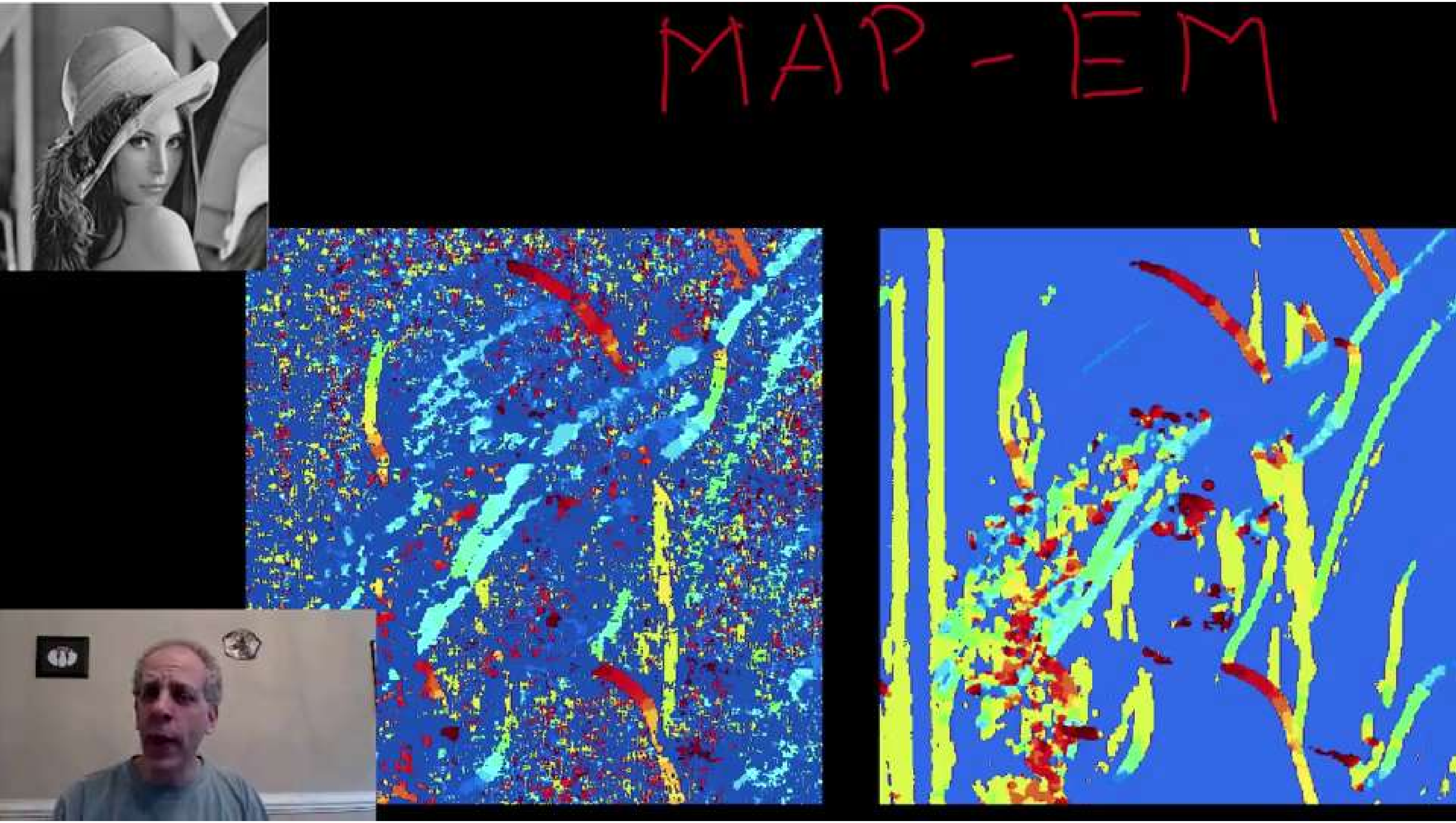
$$\mathbf{w}_i \sim \mathcal{N}(\mathbf{0}, \sigma^2 Id)$$

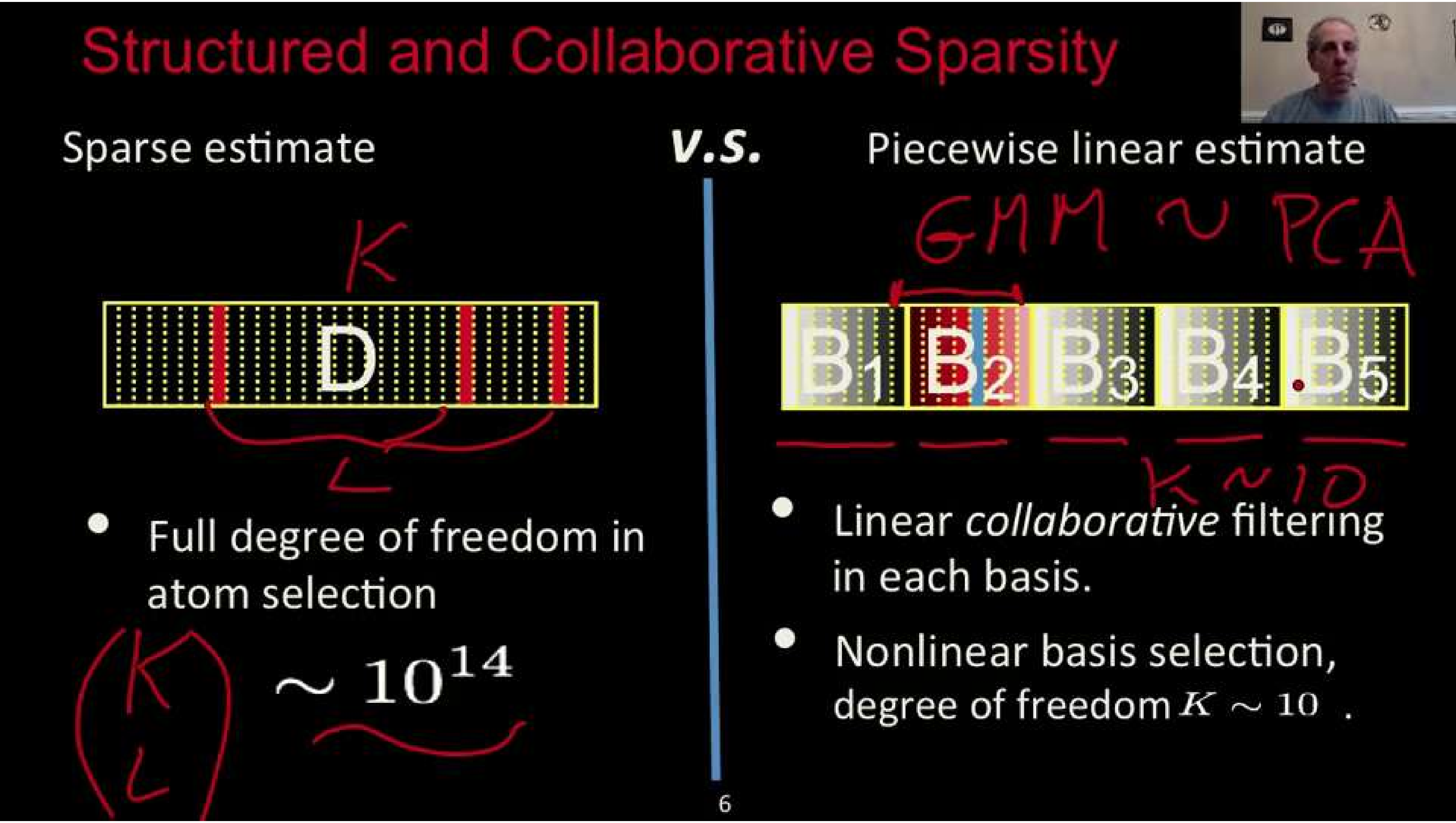
- Estimate $\{(\mu_k, \Sigma_k)\}_{1 \leq k \leq K}$ from $\{\mathbf{y}_i\}_{1 \leq i \leq I}$
- Identify the Gaussian k_i that generates $\mathbf{f}_i \ \forall i$
- Estimate $\tilde{\mathbf{f}}_i$ from $\mathcal{N}(\mu_{k_i}, \Sigma_{k_i}) \ \forall i$

Efficiently solved via MAP-EM



MAP - EM





Structured and Collaborative Sparsity

Sparse estimate

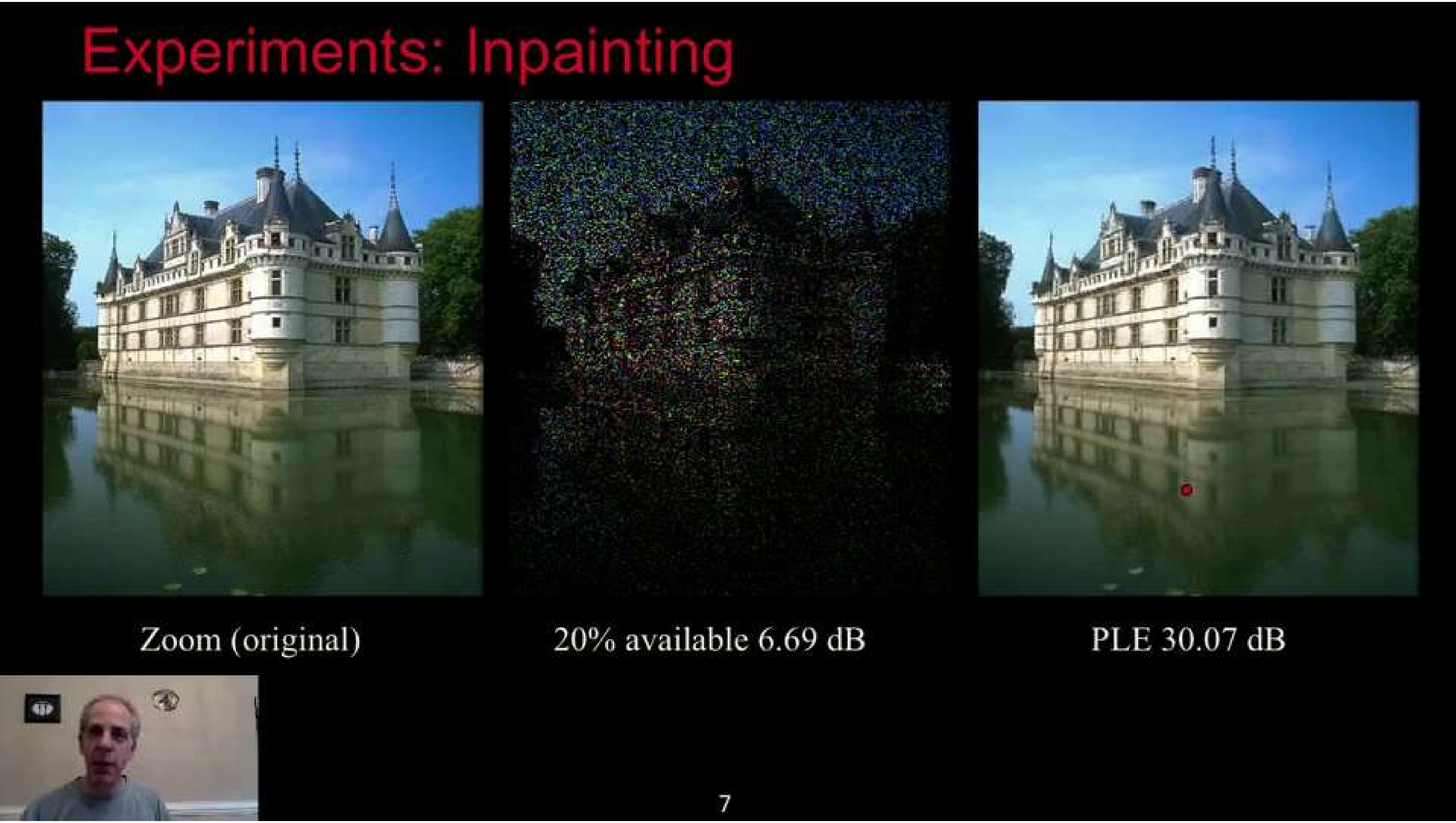
V.S.

Piecewise linear estimate

- Full degree of freedom in atom selection

$$\binom{K}{L} \sim 10^{14}$$

- Linear *collaborative* filtering in each basis.
- Nonlinear basis selection, degree of freedom $K \sim 10$.



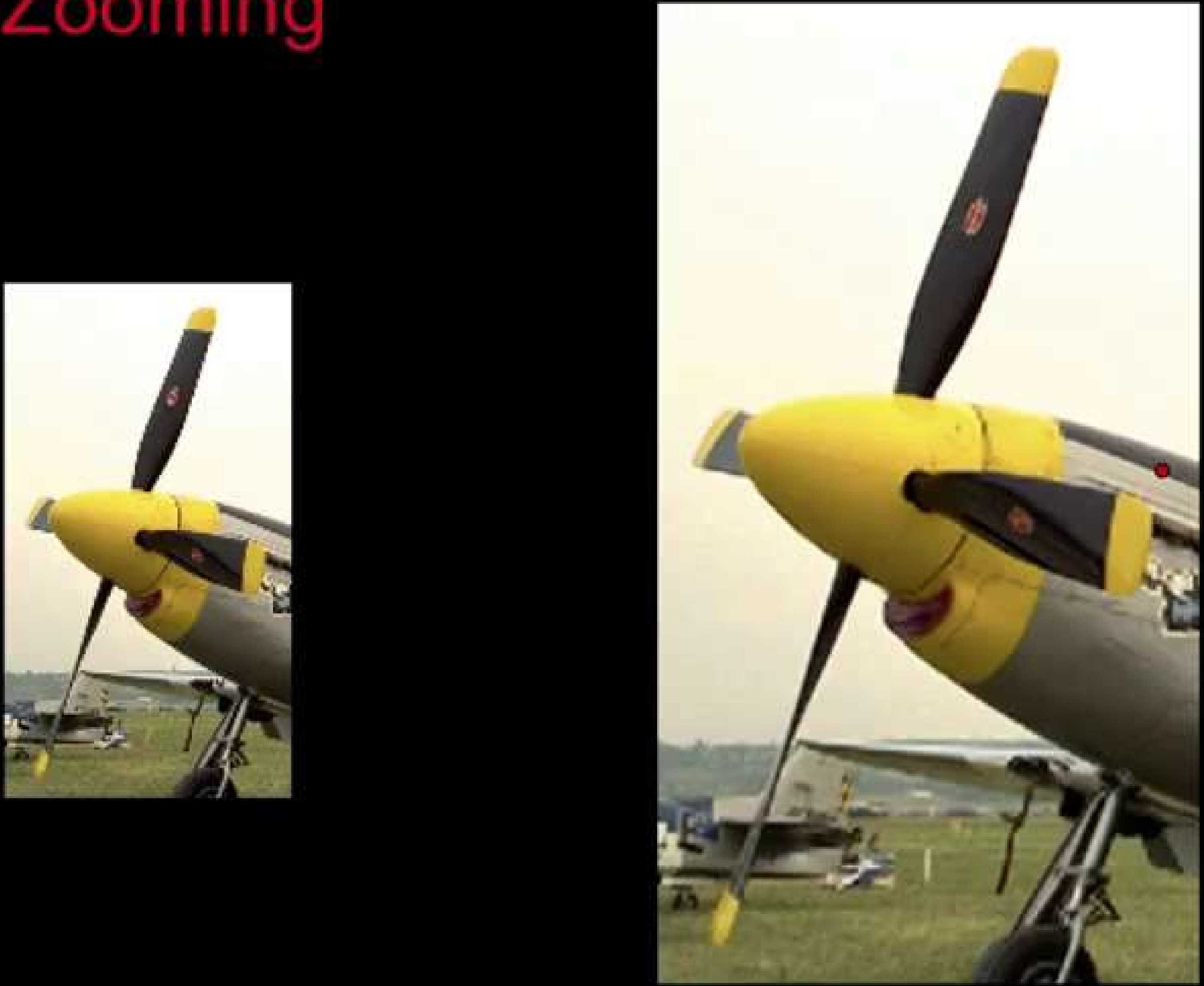
Experiments: Inpainting

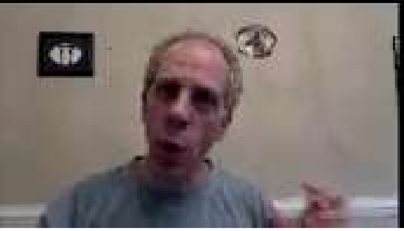
Zoom (original)

20% available 6.69 dB

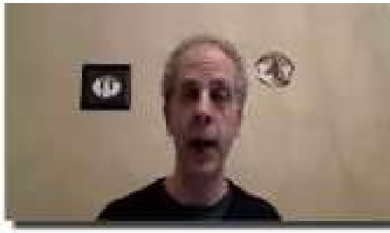
PLE 30.07 dB

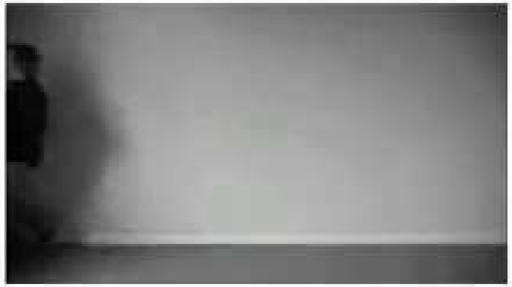
Experiments: Zooming





Motivation



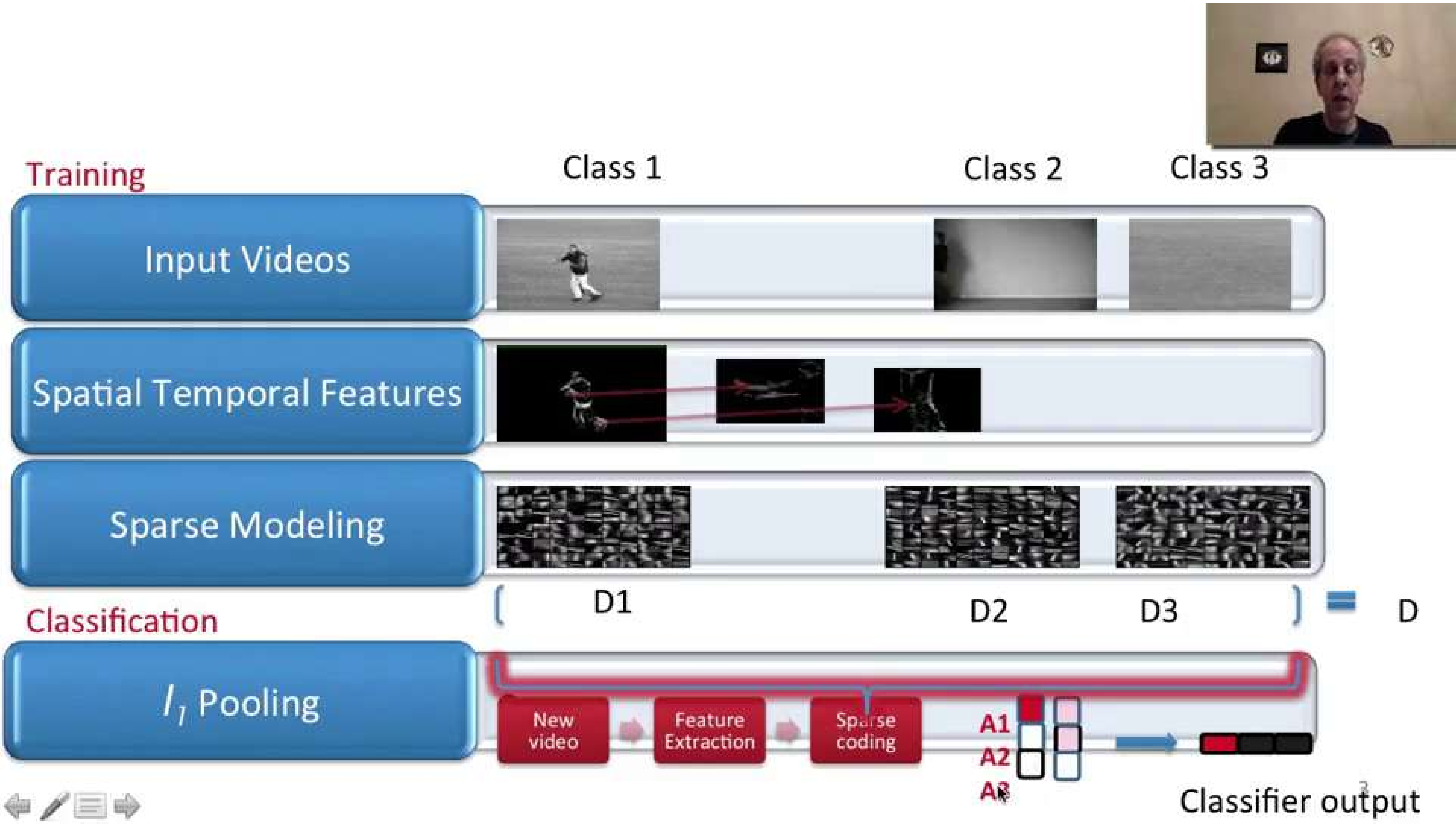


Jogging

Running

Carrying

Jumping



Training

Class 1

Class 2

Class 3

Input Videos

Spatial Temporal Features

Sparse Modeling

[D1 D2 D3] = D

Classification

l_1 Pooling

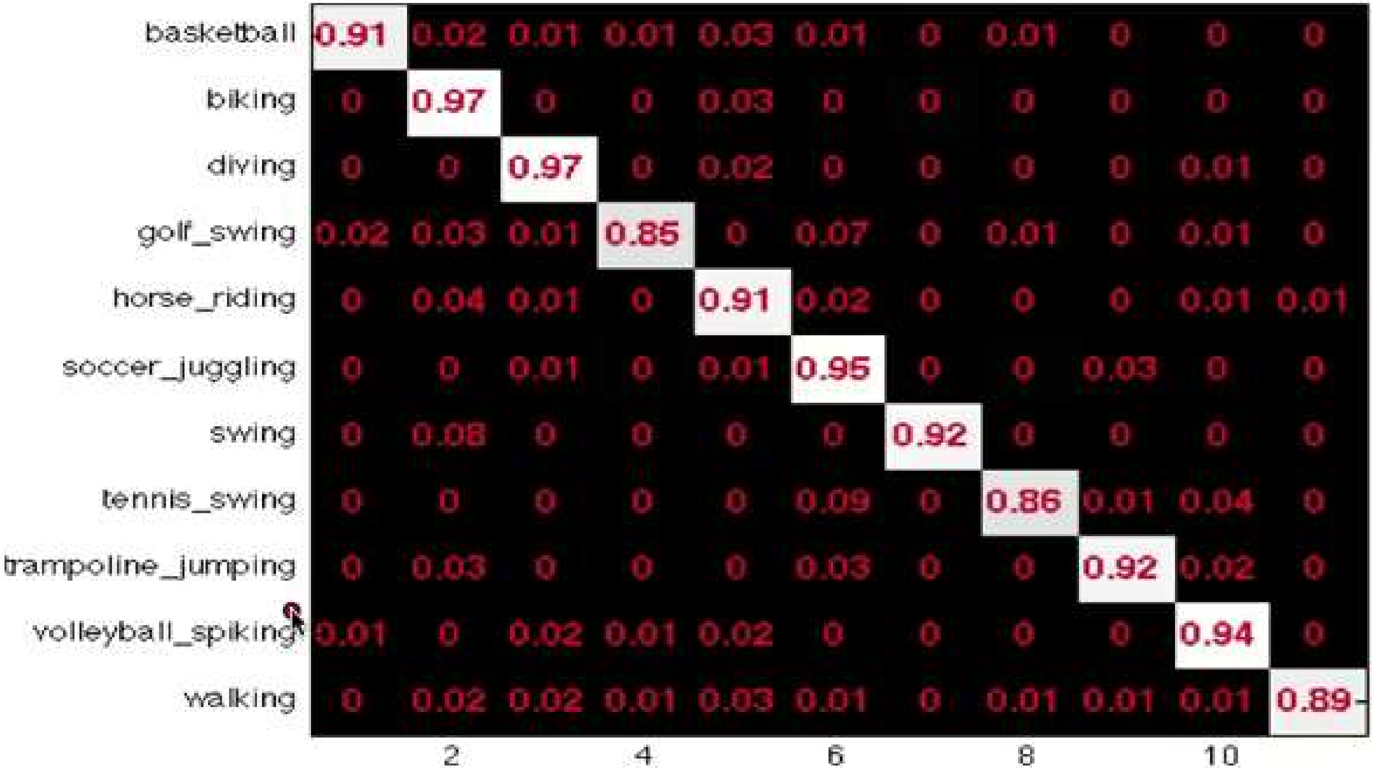
New video

Feature Extraction

Sparse coding

Classifier output

Results: YouTube Action Dataset



basketball	0.91	0.02	0.01	0.01	0.03	0.01	0	0.01	0	0	0
biking	0	0.97	0	0	0.03	0	0	0	0	0	0
diving	0	0	0.97	0	0.02	0	0	0	0	0.01	0
golf_swing	0.02	0.03	0.01	0.85	0	0.07	0	0.01	0	0.01	0
horse_riding	0	0.04	0.01	0	0.91	0.02	0	0	0	0.01	0.01
soccer_juggling	0	0	0.01	0	0.01	0.95	0	0	0.03	0	0
swing	0	0.08	0	0	0	0	0.92	0	0	0	0
tennis_swing	0	0	0	0	0	0.09	0	0.86	0.01	0.04	0
trampoline_jumping	0	0.03	0	0	0	0.03	0	0	0.92	0.02	0
volleyball_spiking	0.01	0	0.02	0.01	0.02	0	0	0	0	0.94	0
walking	0	0.02	0.02	0.01	0.03	0.01	0	0.01	0.01	0.01	0.89
		2		4		6		8		10	


00:00.02