Example-Based Priors For Inverse Problems in Image Processing*

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

* Joint work with

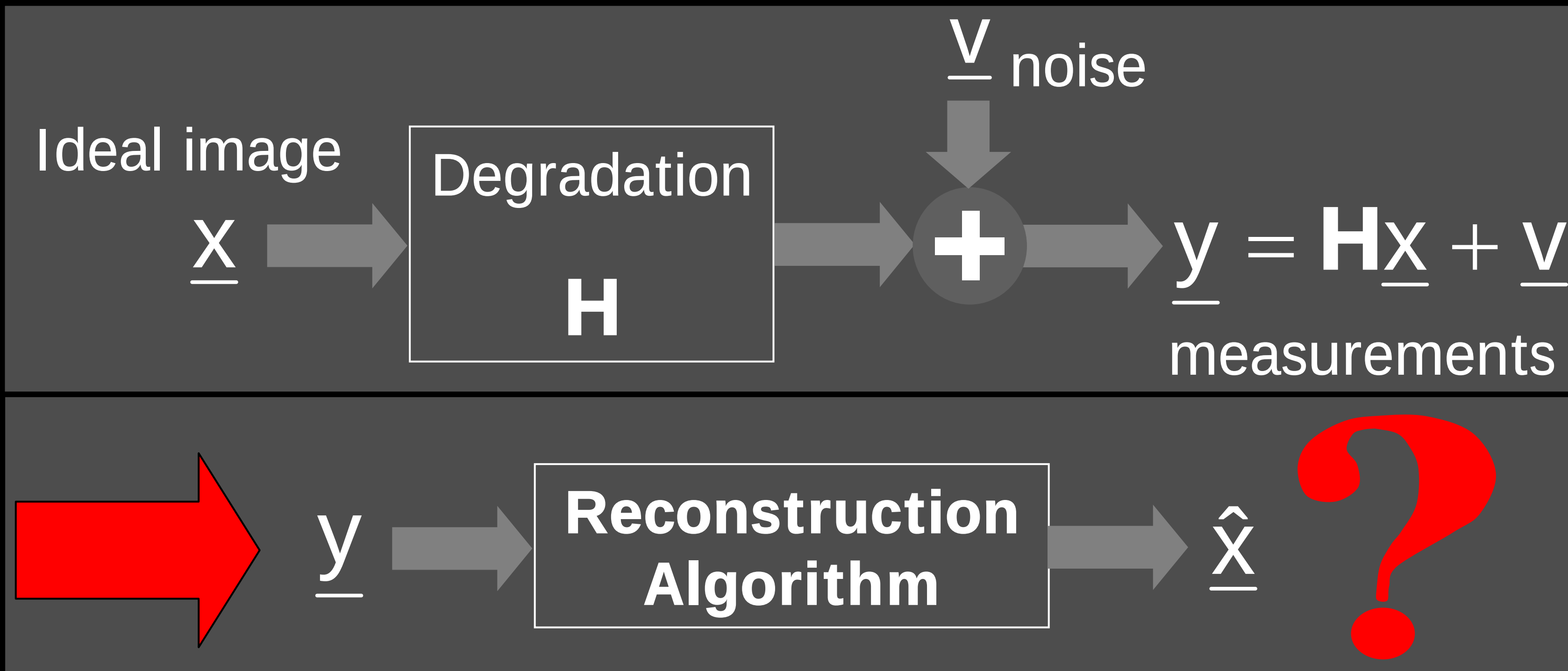
Dmitry Datsenko

International Conference on Super-Resolution Imaging

Hong-Kong, August 29th-31st, 2005

General Motivation

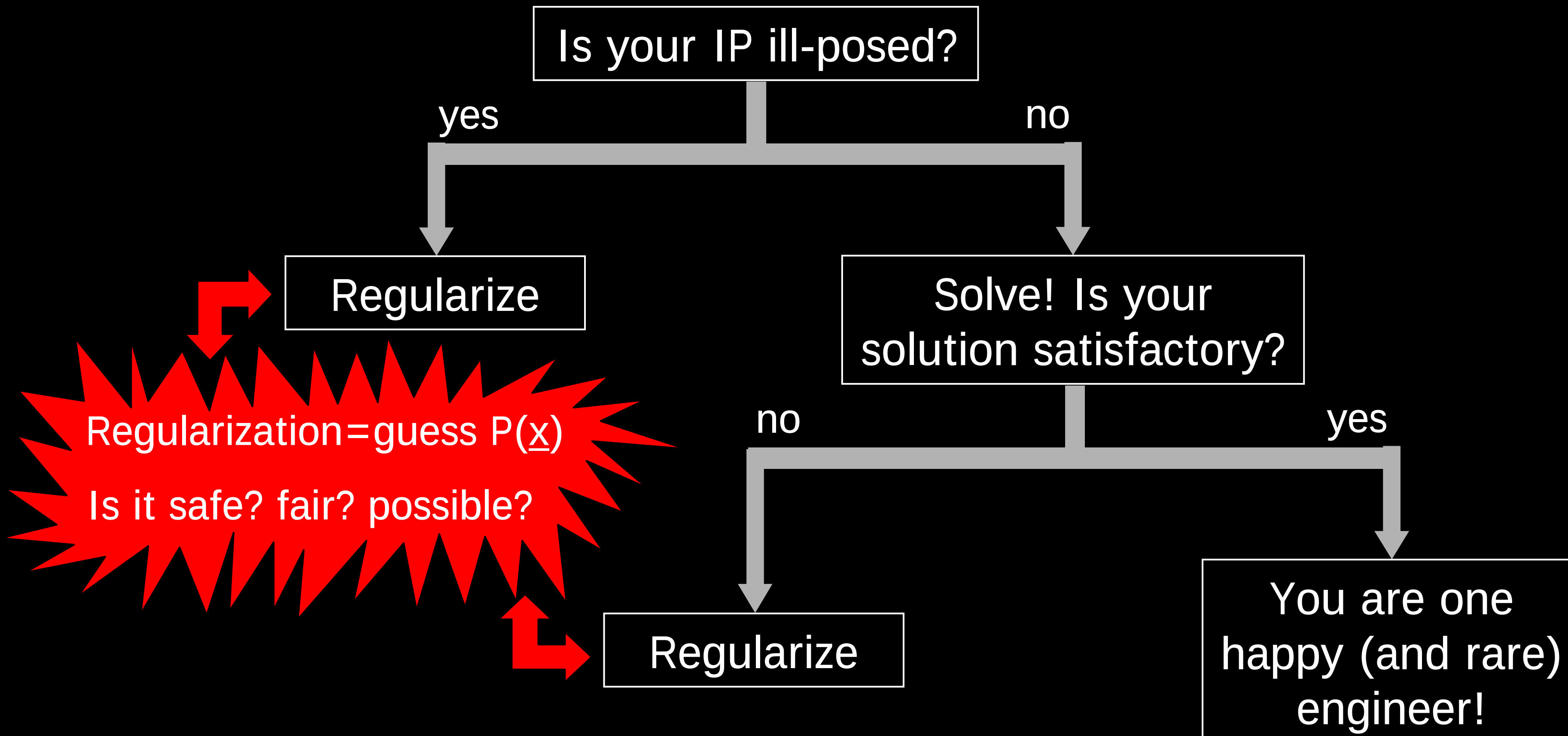
Today we discuss inverse problems



50 years of extensive activity

and there is still long way to go

General Motivation

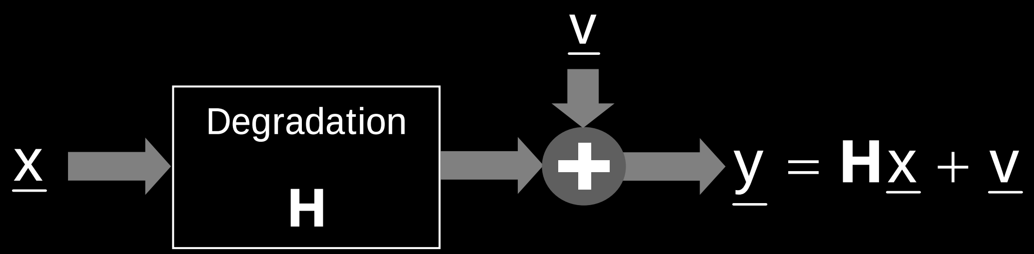


- **Alternative: USE IMAGE EXAMPLES instead of guessing!**
- Today we describe this alternative and its potential, and show new experiments.

Agenda

1. **Regularization – Brief Review**
   Introducing Stability to Inverse Problems
2. Regularization via Examples
   Using Examples? How?
3. Our Recent Experiments
   Simple and general scheme

Maximum–Likelihood (ML) Solution



Assuming zero-mean white Gaussian noise

$$\hat{\underline{x}}_{ML} = \underset{\underline{x}}{\mathrm{Arg\,max}}\, P\left(\underline{y}/\underline{x}\right) = \underset{\underline{x}}{\mathrm{Arg\,min}} \left\| \underline{y} - \mathbf{H}\underline{x} \right\|_2^2$$

Compute $\mathbf{H}^T\mathbf{H}$

singular

non-singular

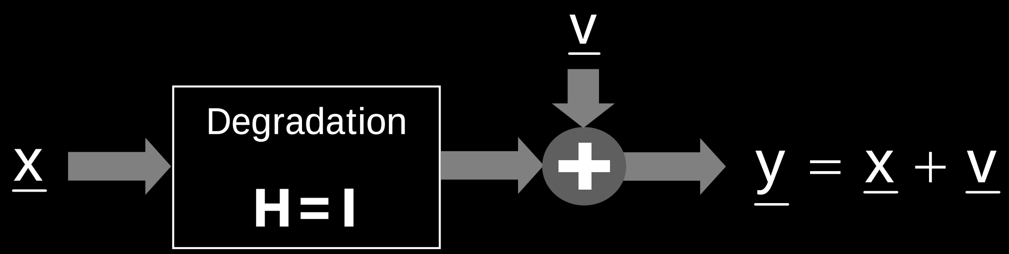
Non-uni … tion!

i.e … ness

Problem

$$\hat{\underline{x}}_{ML} = \left(\mathbf{H}^T\mathbf{H}\right)^{-1}\mathbf{H}^T\underline{y}$$

Well–Posed? It's Not Enough!



Assuming zero-mean white Gaussian noise

$$\hat{\underline{x}}_{ML} = \underset{\underline{x}}{\mathrm{Arg}\max}\, P\left(\underline{y}/\underline{x}\right) = \underset{\underline{x}}{\mathrm{Arg}\min} \left\| \underline{y} - \underline{x} \right\|_2^2$$

Simple but useless result

$$\hat{\underline{x}}_{ML} = \underline{y} \;?$$

Even if the problem is well-posed ($\mathbf{H}^T\mathbf{H} = \mathbf{I}$ – it does not get better!), it does not mean that the ML solution is actually good.

Regularization – Algebraic View

If $\mathbf{H}^T\mathbf{H}$ is singular, "regularize" it [Tikhonov & Arsenin 1977]
by a positive-definite matrix **C** and a small scalar λ

$$\hat{\underline{x}}_{REG} = \left(\mathbf{H}^T\mathbf{H} + \lambda\mathbf{C}\right)^{-1}\mathbf{H}^T\underline{y}$$

This is equivalent to solving

$$\hat{\underline{x}}_{REG} = \underset{\underline{x}}{\mathrm{Arg\,min}} \left\|\mathbf{H}\underline{x} - \underline{y}\right\|_2^2 + \lambda\underline{x}^T\mathbf{C}\underline{x}$$

Why would we want $\underline{x}^T\mathbf{C}\underline{x}$ to be small? Why this expression in particular? Which **C** should we use?

Regularization – Bayesian View

- ML estimation: $\hat{\underline{x}}_{ML} = \underset{\underline{x}}{\mathrm{Arg\,max}}\, P(\underline{y}/\underline{x})$
- Alternative: The Bayesian approach considers the posterior PDF $P(\underline{x}/\underline{y})$ instead.
- In exploring $P(\underline{x}/\underline{y})$ we consider the probable $\underline{x}$, given that $\underline{y}$ was measured. Thus, $\underline{x}$ is also considered random.
- Why called Bayesian? Because of the Bayes formula:

$$P(\underline{x}/\underline{y}) = \frac{P(\underline{y}/\underline{x})P(\underline{x})}{P(\underline{y})}$$

MAP & MMSE Estimators

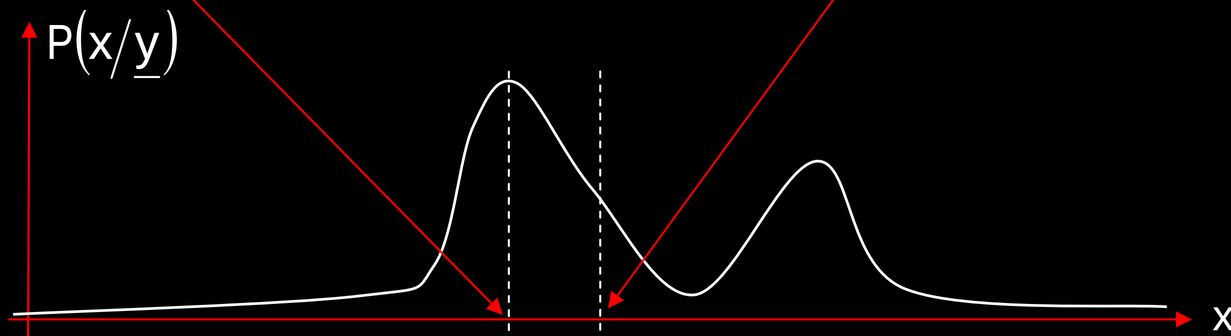
How $P(\underline{x}/\underline{y})$ should be used?

Maximum A-posteriori Probability Estimation

$$\hat{\underline{x}}_{MAP} = \underset{\underline{x}}{\mathrm{Arg\,max}}\, P(\underline{x}/\underline{y})$$

Minimum Mean Squared Error Estimation

$$\hat{\underline{x}}_{MMSE} = E\{\underline{x}/\underline{y}\} = \int \underline{x}\, P(\underline{x}/\underline{y})\, d\underline{x}$$



The MAP Estimator

- While MMSE is considered as a better option, we most often use MAP estimation, because it is much simpler:

$$\hat{\underline{x}}_{MAP} = \underset{\underline{x}}{\mathrm{Arg\,max}}\, P\left(\underline{x}/\underline{y}\right) = \underset{\underline{x}}{\mathrm{Arg\,max}}\, P\left(\underline{y}/\underline{x}\right)P\left(\underline{x}\right)$$

Due to Bayes, and the fact that P(y) can be considered as a constant

- For our inverse problem we have:

$$\hat{\underline{x}}_{MAP} = \underset{\underline{x}}{\mathrm{Arg\,min}}\left\|\mathbf{H}\underline{x} - \underline{y}\right\|_2^2 - \log\left\{P\left(\underline{x}\right)\right\}$$

$$= \underset{\underline{x}}{\mathrm{Arg\,min}}\left\|\mathbf{H}\underline{x} - \underline{y}\right\|_2^2 + f\left(\underline{x}\right)$$

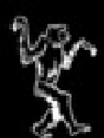
Gibbs distribution

$$P\left(\underline{x}\right) \propto \exp\left\{-f\left(\underline{x}\right)\right\}$$

So, What Shall f(x̲) Be?

During the past several decades we have made all sort of guesses about the functional f(x̲):

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



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

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

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

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

$$f(\underline{x}) = \lambda \|\mathbf{T}\underline{x}\|_1$$

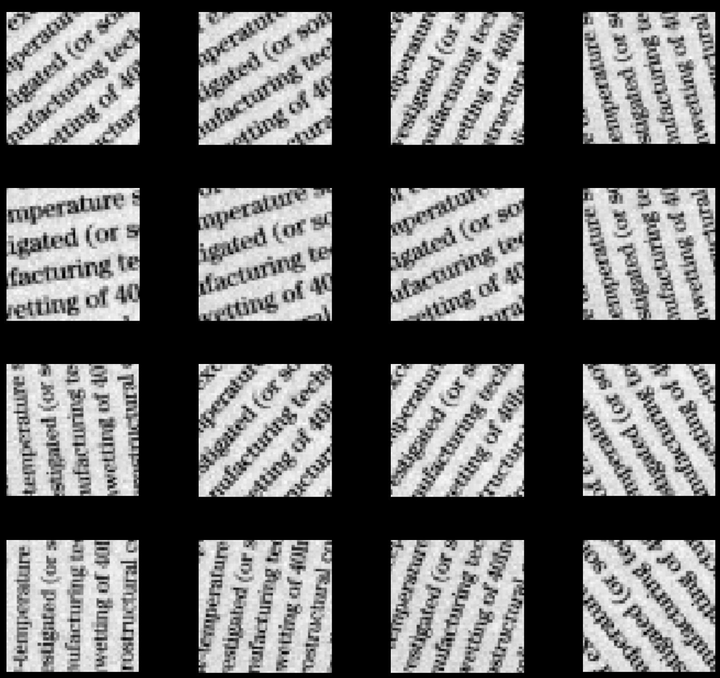
$$f(\underline{x}) = \lambda \|\underline{\alpha}\|_1$$

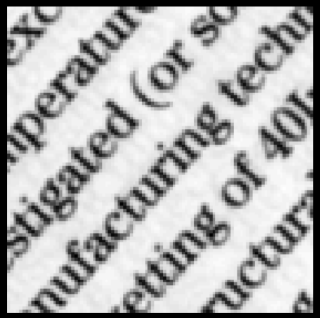
- Mumford & Shah formulation,
- Compression algorithms as priors,
- …

What About Super–Resolution?

Super-Resolution (SR) is an Inverse Problem of the form we have presented

Given: A set of degraded (warped, blurred, decimated, noisy) images:





Required: Fusion of the measurements into a higher resolution image/s

SR is a Regular Inverse Problem

Today we know from several works:

[Baker & Kanade 2001]

[Lin & Shum 2004]

[Milanfar & Robinson 2005]

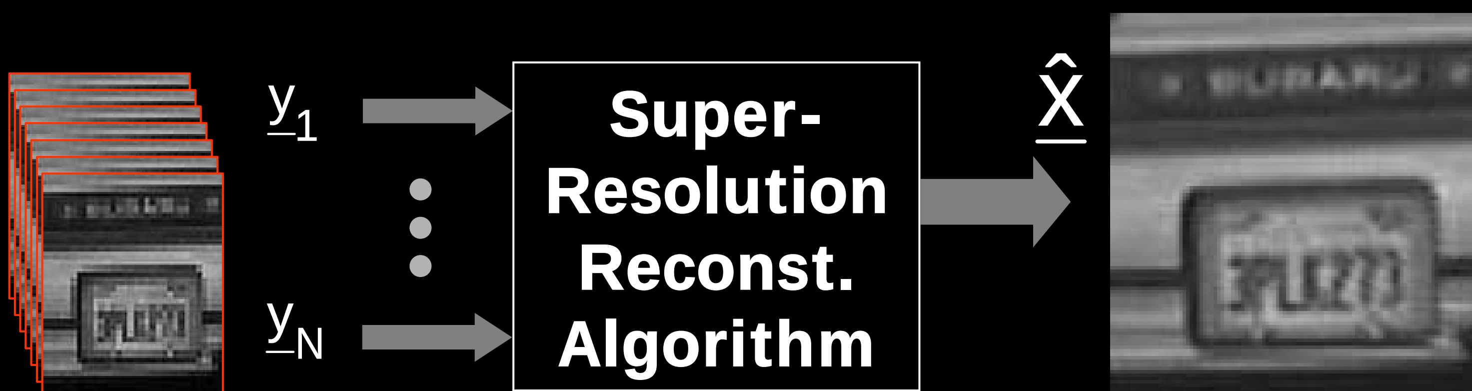
that the SR problem is ill-posed, and adding more measurements cannot fix this fundamental flaw.

THUS: WE NEED REGULARIZATION

Our Experience with SR

For example, in our previous work on
- o robust SR [Farsiu, Elad, Robinson, Milanfar 2003],
- o demosaic+SR [Farsiu, Elad, Milanfar 2004], and
- o dynamic SR [Farsiu, Elad, Milanfar 2005]

we have used the bilateral prior (extension of the TV) to reconstruct super-resolved outcomes



Regularization = Guessing P($\underline{x}$)!!

All these techniques boil down to the fact that we choose f($\underline{x}$) and effectively

GUESS

the prior of the images' PDF:

$$P(\underline{x}) = \frac{1}{z} \cdot \exp\left\{-f(\underline{x})\right\}$$

We do remarkably well, compared to where we stood 30 years ago, but

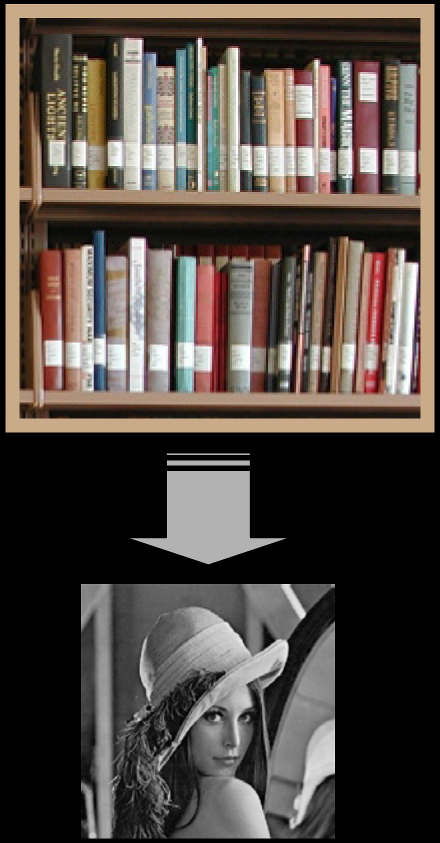
IS IT THE BEST WE CAN DO?

Agenda

1. Regularization Brief Review
   Introducing Stability to Inverse Problems

2. **Regularization via Examples**
   Using Examples? How?

3. Our Recent Experiments
   Simple and general scheme



Main Idea

- Instead of guessing what is P(x) – the PDF of images – why not

  Learn it from image examples? or better yet,

  Use examples to bypass the need for a prior?

- It turns out that these ideas are becoming increasingly appealing in recent years – Here is a partial list of contributions along these lines: [Zhu & Mumford, 1997] [Field & Olshausen 1997] [Simoncelli 1997] [Efros & Leung, 1999] [Engan et. al. 1999] [Wei & Levoy, 2000] [Freeman, Pasztor, & Carmichael, 2000] [Baker & kanade, 2002] [Freeman, Pasztor, & Jones, 2002] [Haber & Tenorio 2003] [Nakagaki & Katsaggelos, 2003] [Cotter et. al. 2003] [Bishop at. al. 2003] [Criminisi et. al. 2004] [Aharon, Elad, and Bruckstein, 2004] [Roth & Black, 2005] [Weissman et. al. 2005].

- Note: when dealing with special & narrow family of images (text, graphics, face images, etc.), this idea can be of much stronger effect.

- We now describe the main ideas behind some of these works.

Examples? How?

How shall we use the examples?

Choose a prior function and **train** its free parameters. Once done, **use** it and its parameters in the inverse prob.

[Zhu & Mumford, 1997]
[Field & Olshausen 1997]
[Simoncelli 1997] [Engan et. al. 1999]
[Baker & kanade, 2002]
[Haber & Tenorio 2003]
[Cotter et. al. 2003] [Aharon, Elad, and Bruckstein, 2004] [Roth & Black, 2005]

Use directly in the reconstruction algorithm, i.e., bypass the prior (and the minimization all together

[Efros & Leung, 1999]
[Wei & Levoy, 2000]
[Freeman, Pasztor, & Carmichael, 2000]
[Freeman, Pasztor, & Jones, 2002]
[Nakagaki & Katsaggelos, 2003] [Bishop et. al. 2003] [Criminisi et. al. 2004]
[Weissman et. al. 2005]

Learning Parameters: MRF

Suppose we are very pleased with the prior

$$f(\underline{x},\underline{\theta}) = \sum_{n=1}^{N} \lambda_n \rho_n \{\mathbf{L}_n \underline{x}\}$$

that employs N linear filters and N robust functions to incorporate their contribution.

Instead of choosing the parameters, we can learn them

using a set of good quality images,

and thus get an overall better prior.

- Two works suggested ideas along the above line: [Zhu & Mumford, 1997] [Roth & Black, 2005].
- The methods vary in the way the training is done. In both, a training phase is required. Once done, the prior is ready to be deployed for any Inverse Problem and any data.

Learning Parameters: MMSE

- Accumulate pairs of images related by the degradation operator:

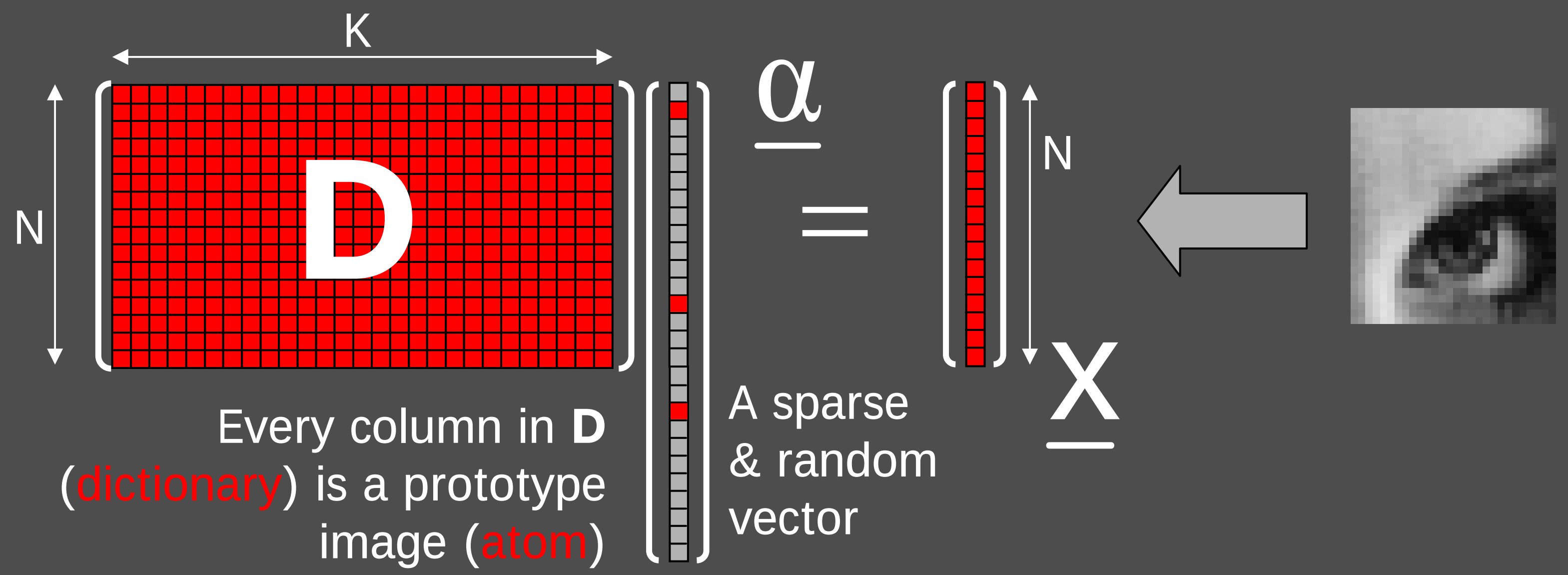
$$\left\{ \underline{x}_k, \underline{y}_k = \mathbf{H}\underline{x}_k + \underline{v} \right\}_{k=1}^{M}$$

- Find the parameters that solve:

$$\min_{\underline{\theta}} \sum_{k=1}^{M} \left\| \underline{x}_k - \underline{\hat{x}}_k \right\|_2^2 \quad \text{s.t.} \left\{ \underline{\hat{x}}_k = \min_{\underline{x}} \left\| \mathbf{H}\underline{x} - \underline{y}_k \right\|_2^2 + f(\underline{x}, \underline{\theta}) \right\}_{k=1}^{M}$$

- This method is described in [Haber & Tenorio 2003]. It features:
  - A prior parameters that are tightly coupled with **H**.
  - An interesting MMSE – MAP mixture.
  - Its optimization task is difficult, and thus suitable for low-dimensional cases.

Learning Parameters: Sparsity



Our reconstruction is done by

$$\underset{\underline{\alpha}}{\text{Arg}\min} \left\| \underline{y} - \mathbf{HD}\underline{\alpha} \right\|_2^2 + \lambda \left\| \underline{\alpha} \right\|_1$$

The prior: $\underline{x}$ should have a sparse representation over **D**.

- The prior parameters are **D**.
- Several works study its training [Field & Olshausen 1997] [Engan et. al. 1999] [Cotter et. al. 2003] and [Aharon, Elad, and Bruckstein, 2004]

Examples? How?

How shall we use the examples?

Choose a prior function and train its free parameters. Once done, use it and its parameters in the inverse prob.

[Zhu & Mumford, 1997]
[Field & Olshausen 1997]
[Simoncelli 1997] [Engan et. al. 1999]
[Baker & kanade, 2002]
[Haber & Tenorio 2003]
[Cotter et. al. 2003] [Aharon, Elad, and Bruckstein, 2004] [Roth & Black, 2005]

Use **directly** in the reconstruction algorithm, i.e., bypass the prior (and the minimization all together

[Efros & Leung, 1999]
[Wei & Levoy, 2000]
[Freeman, Pasztor, & Carmichael, 2000]
[Freeman, Pasztor, & Jones, 2002]
[Nakagaki & Katsaggelos, 2003]
[Criminisi et. al. 2004]
[Weissman et. al. 2005]

Direct Use of Examples – General

Gather a set of examples that describe the relation between high-quality (result-like) and low-quality (measurements-like) patches.

Use those pairs to perform the reconstruction directly, without going through an explicit stage of Prior evaluation

- Several works suggested algorithms along these lines:
  - Texture/image synthesis using the corrupted data for examples [Efros & Leung, 1999] [Wei & Levoy, 2000] [Criminisi et. al. 2004].
  - Ziv-Lempel like Denoising [Weissman et. al. 2005].
  - Super-Res. with Bayesian-Belief-Prop. [Freeman, Pasztor, & Carmichael, 2000].
  - Deblurring by adding high-freq. content [Nakagaki & Katsaggelos, 2003].
- All these methods rely on extensive nearest-neighbor searches, and thus the database needs to be efficiently pre-organized (VQ,PCA, ...).

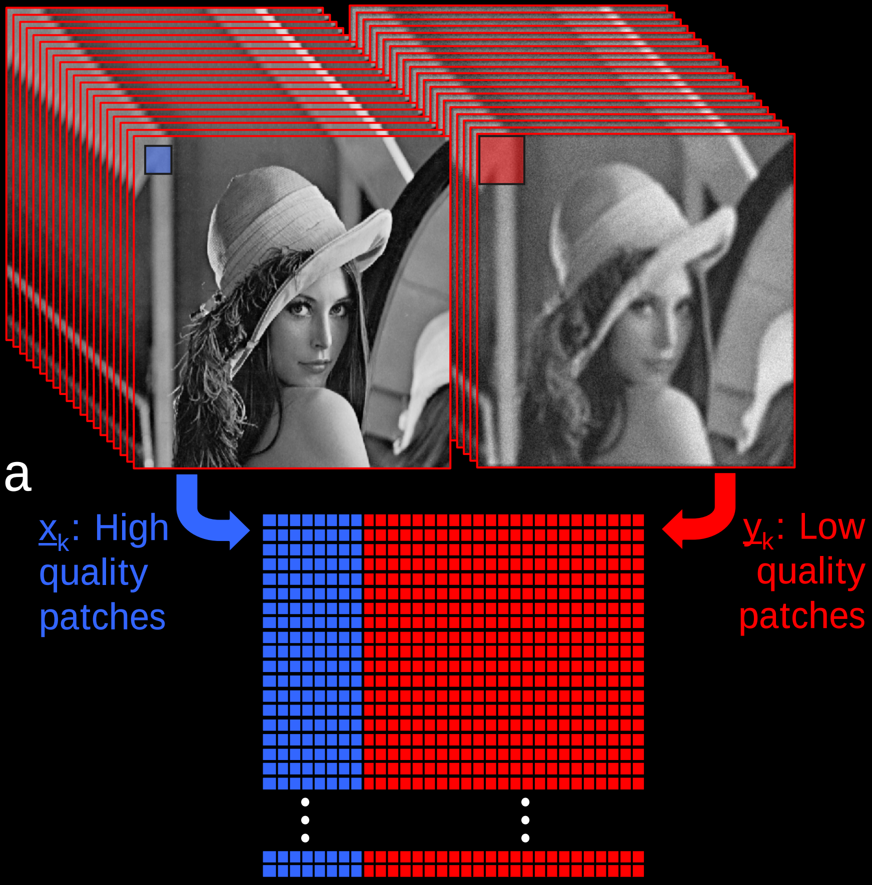
Example for Such an Approach

OFF-LINE STAGE

- Accumulate pairs of image patches related by the degradation operator:

$$\left\{\underline{x}_k, \underline{y}_k = \mathbf{H}\underline{x}_k + \underline{v}\right\}_{k=1}^{M}$$

- Pre-process the database to facilitate a fast search, and better results:
  - Vector quantization on $\left\{\underline{y}_k\right\}_{k=1}^{M}$.
  - PCA for dimensionality reduction.
  - Using informative features (derivatives, high-frequency, multi-scale details, ...).

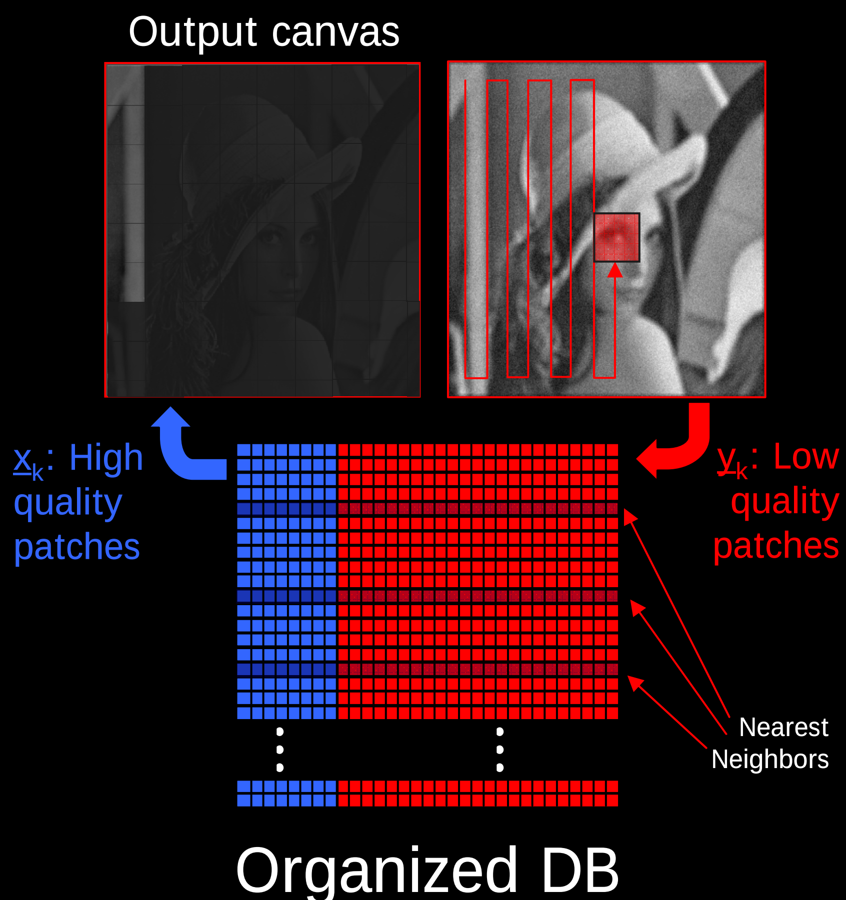


Example for Such an Approach

ON-LINE STAGE

- ❑ Sweep through the given (low-quality) image, patch by patch (with overlaps).
- ❑ Per the patch $\underline{y}_k$, find its J-nearest neighbors in the DB.
- ❑ Refer to these NN matching high-quality part $\underline{x}_k^1, \underline{x}_k^2, \ldots, \underline{x}_k^J$.
- ❑ Use those to update the result by
  - (weighted) averaging, or
  - Bayesian Belief Propagation.

[Freeman, Pasztor, & Carmichael, 2000]
[Freeman, Pasztor, & Jones, 2002]
[Nakagaki & Katsaggelos, 2003]



Major Questions to Consider …

Which examples to use?

high

high & low

low

How to use the examples?

- Train to set Prior param.
  - MRF
  - sparse
- Use directly in the alg.
  - Ave.
  - BBP

Which estimator to target?

- MMSE
- other
- MAP

Other important questions:

- ❑ How to represent examples?
- ❑ How to organize the examples?
- ❑ Which IP to target?

Agenda

geometric image
processing laboratory

1. Regularization – Brief Review

   Introducing Stability to Inverse Problems

2. Regularization via Examples

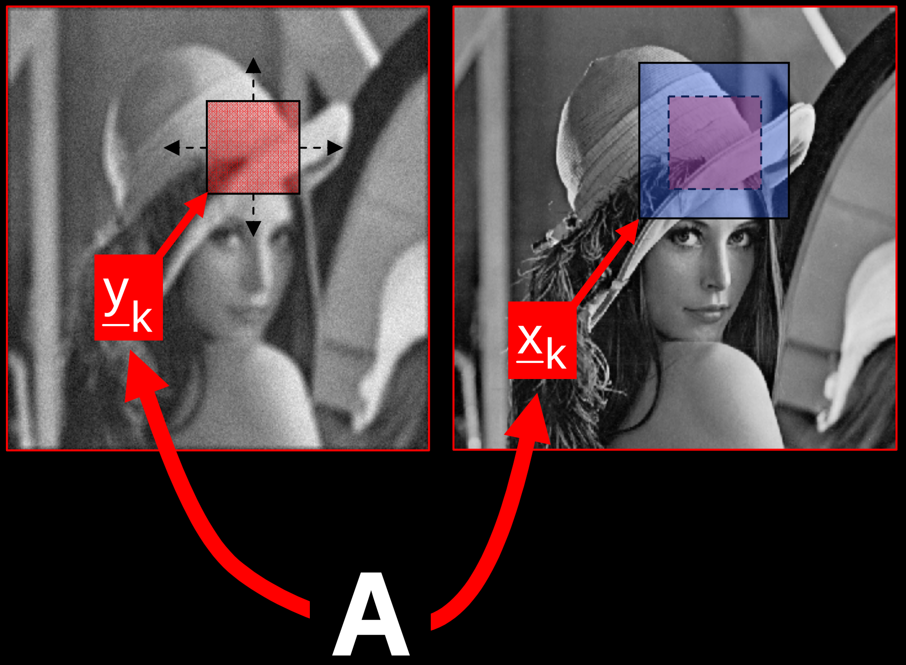
   Using Examples? How?

3. **Our Recent Experiments**

   Simple and general scheme

Definition: The Influence Zone

- Given a degraded image $\underline{Y}$, we consider working with fully overlapping patches of size $N \times N$ pixels. Denote such patch as $\underline{y}_k$.
- Knowing **H**, and assuming locality, we can define a corresponding patch $\underline{x}_k$ in $\underline{X}$ (of size $L \times L$, $L \geq N$) that contains all the pixels that generate $\underline{y}_k$. This is the influence zone.



- We define an $N^2 \times L^2$ operator **A** that relates the two (T is a function of the noise):

$$\left\| \mathbf{A}\underline{x}_k - \underline{y}_k \right\|_2 \leq T$$

Examples for $\underline{y}_k : 3 \times 3$

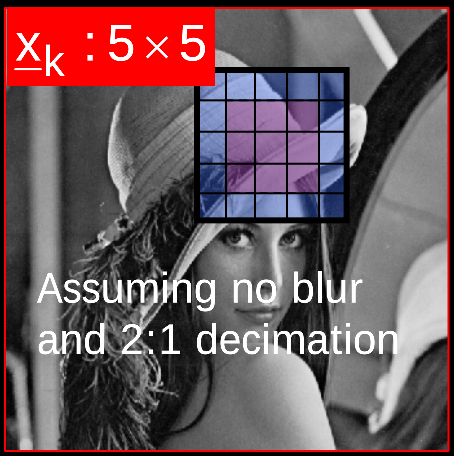
Denoising

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} = \mathbf{I}_9$$

Deblurring

$$A_1 = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 \end{bmatrix}, \ \mathbf{A} = A_1 \otimes A_1$$

Up-Scaling

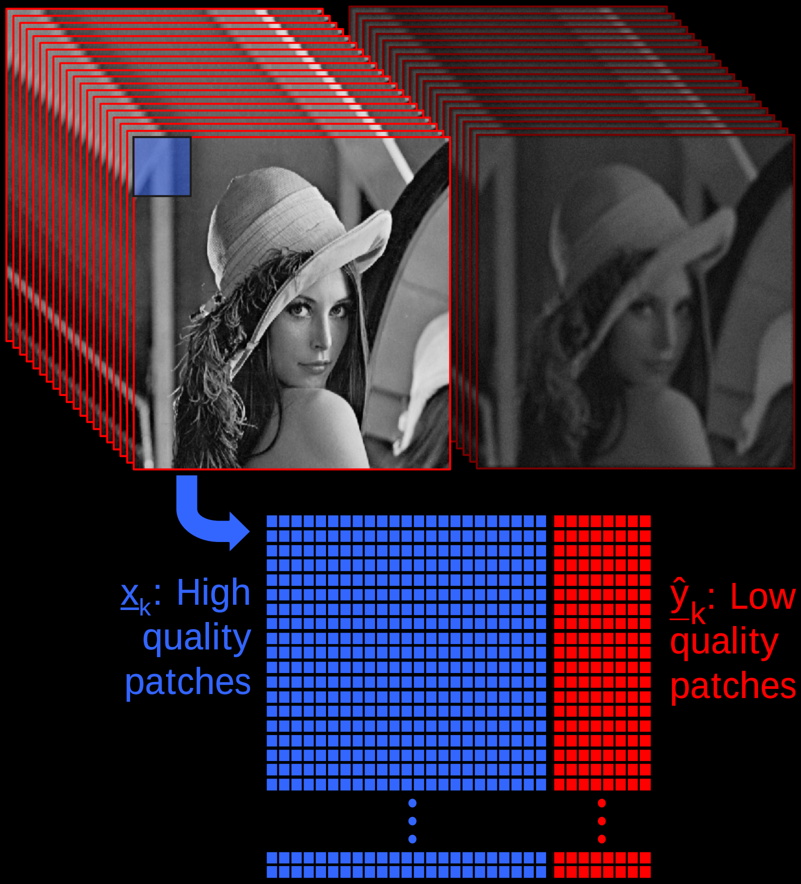


$$A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}, \ \mathbf{A} = A_1 \otimes A_1$$

The Basics: Off–Line

- Suppose that we have a database containing many $\underline{x}_k$ patches of the desired (L×L) size.
  Question 1: How to organize it?

But we will not do that !!



- This database (that has ONLY high quality image patches) will be used in the reconstruction for any known **H**.

The Basics: On–Line

- Given a low-quality image (with known local degradation), we sweep through it in overlapping blocks $\underline{y}_k$.

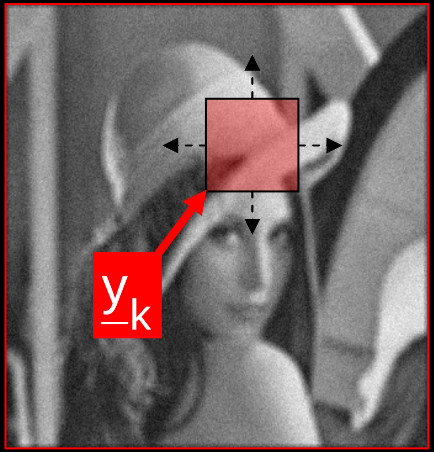
- Per each $\underline{y}_k$ we search the DB for its J_k (may vary) nearest-neighbors, satisfying

$$\left\| \hat{\underline{y}}_k - \underline{y}_k \right\|_2^2 = \left\| \mathbf{A}\underline{x}_k - \underline{y}_k \right\|_2^2 \leq T$$

Question 2: How to find the NN?

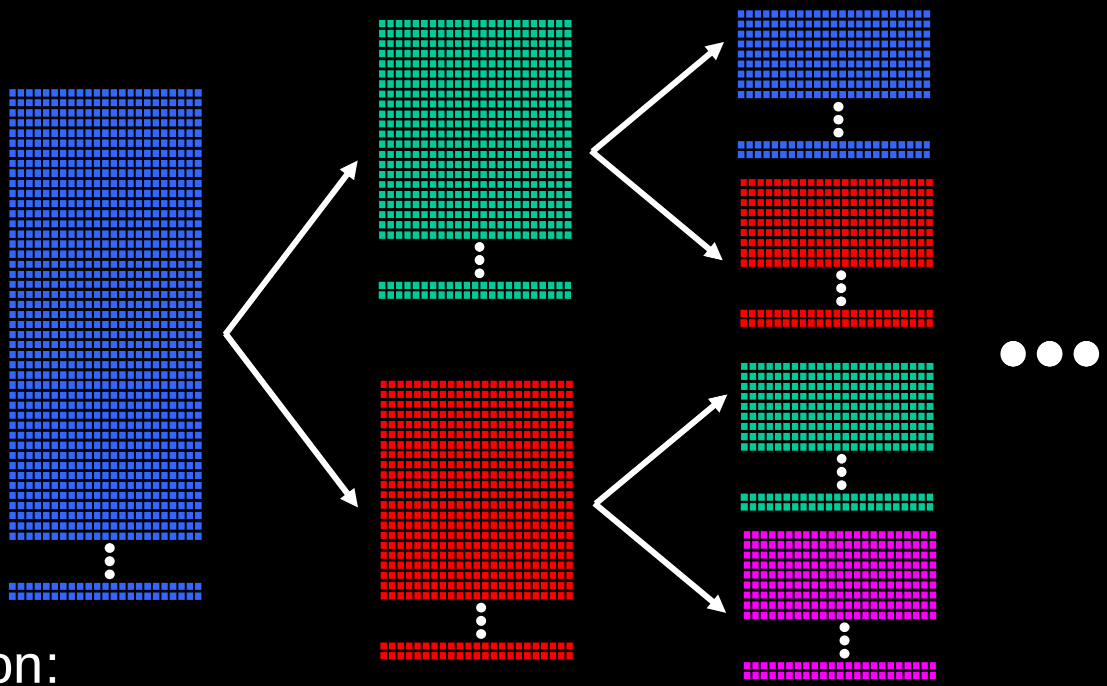
- We mark the found neighbors as $\left\{ \underline{x}_k^j \right\}_{j=1}^{J_k}$. These will be used for the reconstruction.

Question 3: How to use those examples in the recovery?



Q1: How to Organize the DB?

- The database $\{\underline{x}_k\}_{k=1}^{M}$ should be decomposed hierarchically in a tree structure (k-d-Tree for box-like clusters or Tree-K-Means for spheres). Its leafs are the actual examples.



- Every cluster holds its smallest bounding information:
  - **Box**: its corners; or
  - **Sphere**: center + radius,
- Instead of applying the operator **A** on all the DB, apply it only on the bounding features (center or corners) upon need.

Q2: How to Find the NN?

Our goal: given $\underline{y}_k$, find its J_k nearest-neighbors, satisfying

$$\left\| \mathbf{A}\underline{x}_k - \underline{y}_k \right\|_2^2 \leq T$$

Brute-force

"Blur" the DB

$$\left\{\underline{x}_k\right\}_{k=1}^{M} \xrightarrow{\mathbf{A}} \left\{\underline{\hat{y}}_k\right\}_{k=1}^{M}$$

Search directly NN

$$\left\| \underline{\hat{y}}_k - \underline{y}_k \right\|_2^2 \leq T$$

Fast

A fast algorithm is possible because

- of the DB tree structure,
- Only "centers" are "blurred",
- Cross-distances are pre-computed,
- A rejection-based search is applied,
- Option: a multi-resolution search.

The Fast Search (1)

Core tool: Given a cluster C_0 with center $\underline{x}_0$ and radius r_0, and given a candidate patch, $\underline{y}_k$, we should solve:

$$z_{opt} = \underset{\underline{z}}{\mathrm{Argmin}} \left\| \mathbf{A}\underline{z} - \underline{y}_k \right\|_2^2 \quad \text{s.t.} \quad \left\| \underline{z} - \underline{x}_0 \right\|_2 \le r_0$$

- Closed form solution using SVD.
- Similar structure for k-d-tree, solved with LSQLIN.

$$d\left\{\underline{y}_k, C_0\right\} = \left\| \mathbf{A}\underline{z}_{opt} - \underline{y}_k \right\| \ge T\,?$$

yes

This cluster and all its content is rejected

no

Still suspect: dive in to the next level

Using $\underline{z}_{opt}$, we can make simple tests to see if it falls inside the sub-clusters, and save computations this way

The Fast Search (2)

Assume that as part of the DB organization, we gather the cross distances between the cluster-centers in each level

We can use a triangle inequality

$$d\{\underline{y}_k, C_1\} \geq d\{\underline{y}_k, C_0\} - \sigma_{max}\{\mathbf{A}\}\left(\left\|\underline{x}_0 - \underline{x}_1\right\| + r_0 + r_1\right)$$

We get a lower-bound on the distance to all other clusters

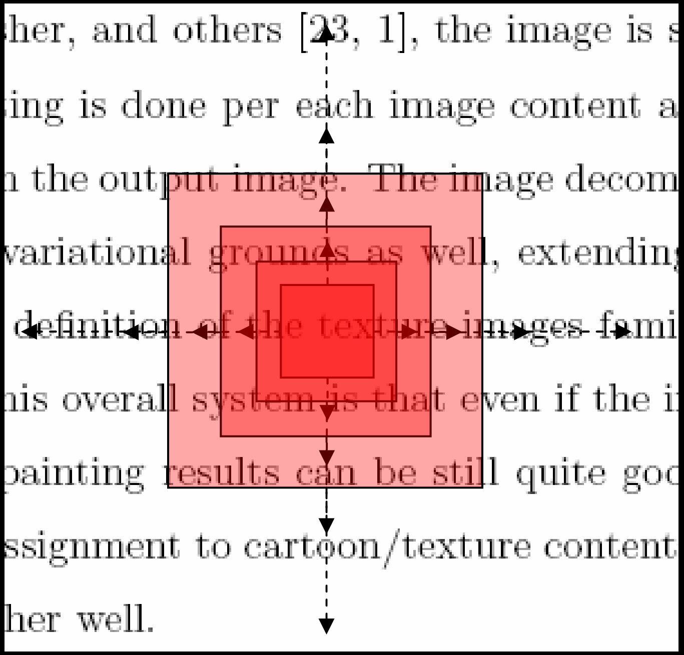
A distance we computed as part of the search

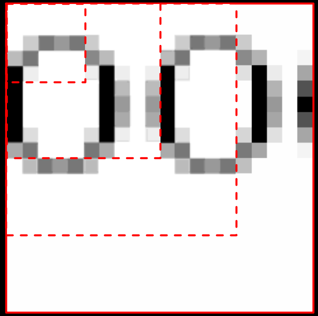
A pre-evaluated distance

Yet another rejection

Role of Multi–Scale?

- ❑ So far we assumed that the patches are of constant size (in x and y).
- ❑ Varying the size can be of worth, both for **speed** and for **accuracy**:
  - Speed: Starting with small patch-size, the tests are cheaper, and only on the non-rejected results we should proceed with a larger patch-size.
  - Accuracy: As the patch-size increases, the number of candidate neighbors drops and sharpens, reducing chances of multi-modal posterior. In fact, a stopping rule to the patch-size increase could be the number of examples found.

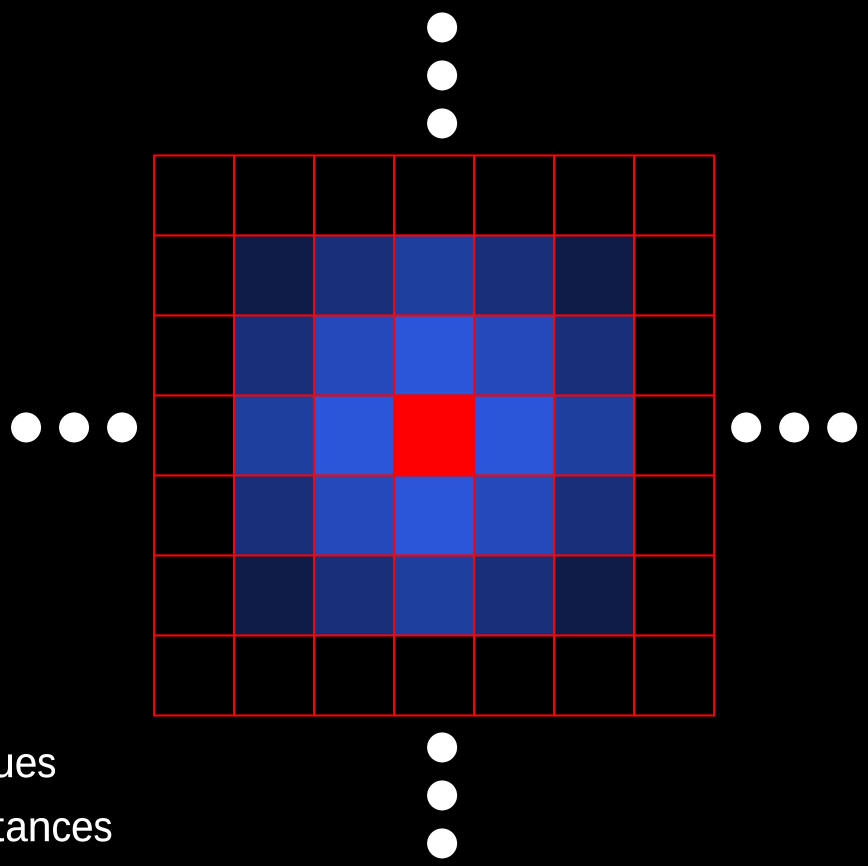




Q3: How to Reconstruct?

- Consider the output image ... and refer to one specific pixel in it.
- Suppose we work with blocks of 3×3 in x̲ (i.e., L=3) and see how they contribute to the chosen pixel.
- Per every such location, we have a group (J_k) of candidate NN examples, with a distance attribute.
- Thus we have an array of the form:

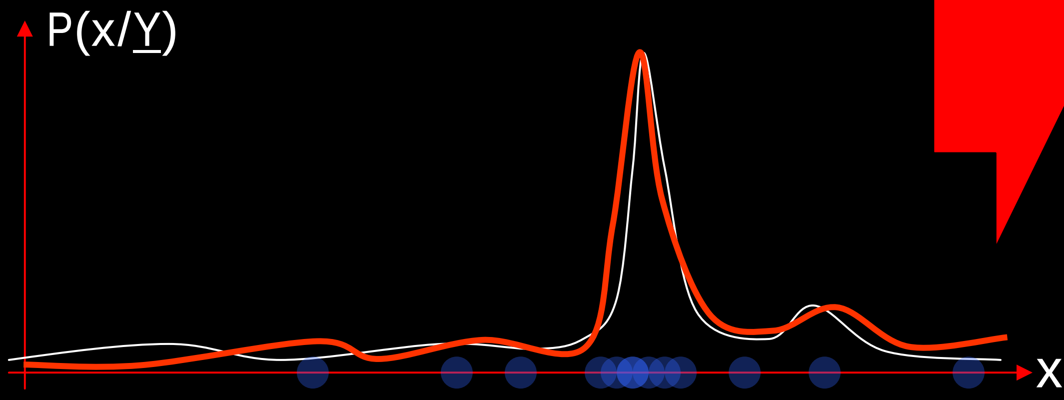
$$\begin{array}{cc} \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & \cdots & x_n \\ \varepsilon_1 & \varepsilon_2 & \varepsilon_3 & \varepsilon_4 & \varepsilon_5 & \varepsilon_6 & \varepsilon_7 & \varepsilon_8 & & \varepsilon_n \end{bmatrix} & \begin{array}{l} \text{Values} \\ \text{Distances} \end{array} \end{array}$$



The Reconstruction

- We have the array: $\begin{bmatrix} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_8 & \cdots & x_n \\ \varepsilon_1 & \varepsilon_2 & \varepsilon_3 & \varepsilon_4 & \varepsilon_5 & \varepsilon_6 & \varepsilon_7 & \varepsilon_8 & & \varepsilon_n \end{bmatrix}$ Values / Distances

- Assumption: these points stand for sampling from the scalar posterior, $P(x/\underline{Y})$, where:
  x: the pixel in question
  $\underline{Y}$: the zone covered by all these patches



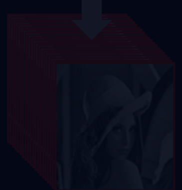
- A histogram of these examples leads to the posterior: MAP and MMSE are within reach.
- The distances are helpful in weighting those samples, and robustifying against the choice of T.

To Summarize

These features make the proposed algorithm close in spirit to the work reported in
[Freeman et. al. 2000,2002]
[Nakagaki & Katsaggelos, 2003]
but different in several important aspects.

Which examples to use?

high



How to use the examples?

Use directly in the alg.

Ave.

BBP

MMSE

other

MAP

Other important questions:

- How to represent examples? Plain & MultiScale
- How to organize the examples? Special way
- Which IP to target? Several at once

Few Text Experiments

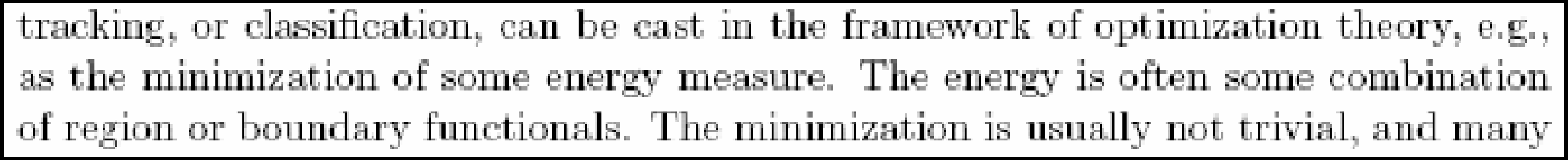
- We will now present few reconstruction examples:
  - Denoising,
  - Deblurring, and
  - Up-Scaling.
- These text experiments use training patches (~60,000) taken from:

> with the method of active contours. Active contours have been introduced by Kass, Witkin and Terzopoulos [34] and were originally boundary methods. Snakes [34], balloons [10], or geodesic active contours [4] are driven towards the edges of an image through the minimization of a boundary integral of functions of features depending on edges. Active contours driven by region functionals in addition to boundary functionals have appeared later. Introduced by [11] and [43], they have been further developed

- These experiments are in line with those done by Baker & Kanade.

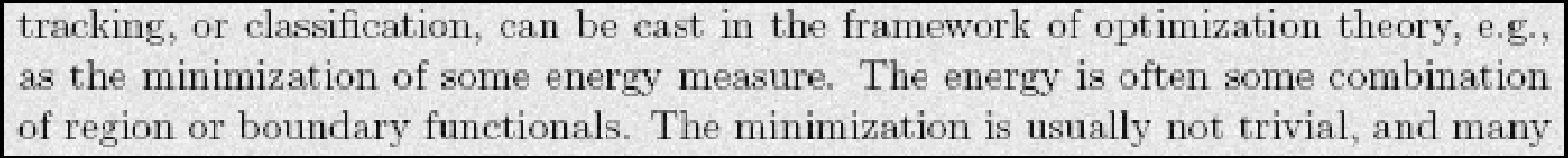
Results: Denoising

Original Image $\underline{x}$



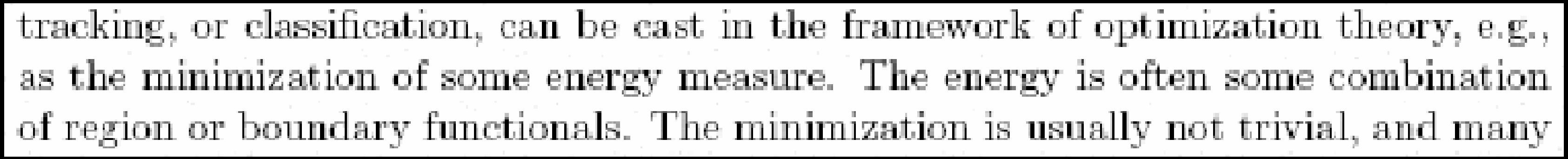
tracking, or classification, can be cast in the framework of optimization theory, e.g., as the minimization of some energy measure. The energy is often some combination of region or boundary functionals. The minimization is usually not trivial, and many

Measured image $\underline{y}$: Additive Gaussian noise $\mathcal{N}(0,8\cdot\mathbf{I})$



tracking, or classification, can be cast in the framework of optimization theory, e.g., as the minimization of some energy measure. The energy is often some combination of region or boundary functionals. The minimization is usually not trivial, and many

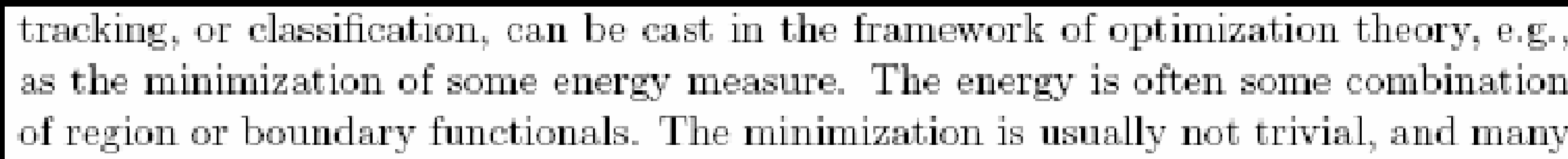
Reconstructed [5×5 patches] RMSE=2.45



tracking, or classification, can be cast in the framework of optimization theory, e.g., as the minimization of some energy measure. The energy is often some combination of region or boundary functionals. The minimization is usually not trivial, and many

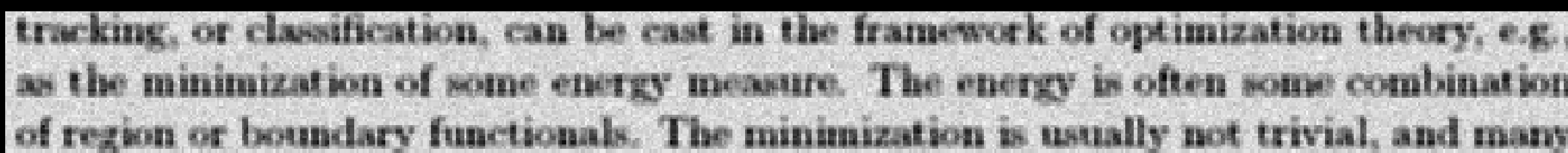
Result: Deblurring

Original Image $\underline{x}$



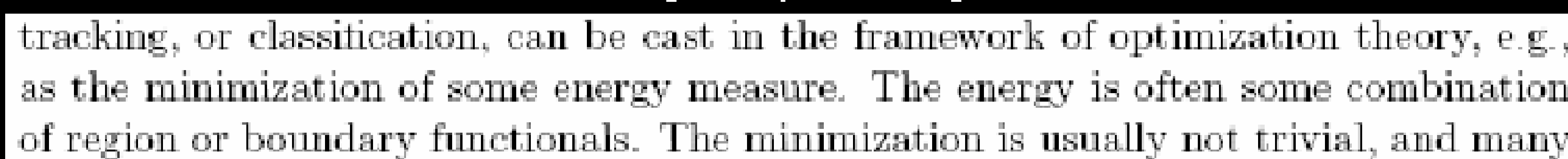
tracking, or classification, can be cast in the framework of optimization theory, e.g., as the minimization of some energy measure. The energy is often some combination of region or boundary functionals. The minimization is usually not trivial, and many

Measured image $\underline{y}$: 3×3 uniform blur & Gaussian noise $\mathcal{N}(0,8\cdot\mathbf{I})$



tracking, or classification, can be cast in the framework of optimization theory, e.g., as the minimization of some energy measure. The energy is often some combination of region or boundary functionals. The minimization is usually not trivial, and many

Reconstructed [5×5 patches]   RMSE=6.4



tracking, or classitication, can be cast in the framework of optimization theory, e.g., as the minimization of some energy measure. The energy is often some combination of region or boundary functionals. The minimization is usually not trivial, and many

Result: Up–Scaling

Original Image $\underline{x}$

tracking, or classification, can be cast in the framework of optimization theory, e.g.,
as the minimization of some energy measure. The energy is often some combination
of region or boundary functionals. The minimization is usually not trivial, and many

Measured image $\underline{y}$: $\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$ blur, 2:1 decimation & noise $\mathcal{N}(0,8\cdot\mathbf{I})$

tracking, or classification, can be cast in the framework of optimization theory, e.g.,
as the minimization of some energy measure. The energy is often some combination
of region or boundary functionals. The minimization is usually not trivial, and many

Reconstructed [8×8 patches] RMSE=10.01

tracking, or classification, can be cast in the framework of optimization theory, e g.,
as the minimization of some energy measure. The energy is often some combination
of region or bourdary functionals. The minimization is usually not trivial. and many

Few Face Experiments

- We now present few preliminary reconstruction examples for 4:1 scaled-up faces.
- The faces (training and testing) are taken from the ORL database.
- These experiments use training patches (~50,000) taken from 7 faces (different people!) from the ORL DB, and allowing offset and scale:

- These experiments are in line with those done by Baker & Kanade.
- Results CAN BE IMPROVED – We are working on this.

Result: Up–Scaling

Original

Measured

Blurred by $\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$

4:1 decimation

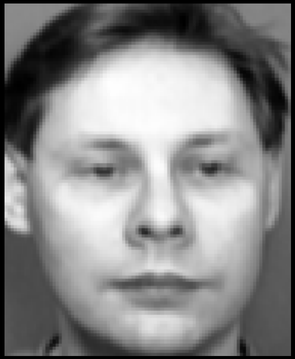
& noise $\mathcal{N}(0,8\cdot\mathbf{I})$

Bilinear Inter.

Reconstructed

Block size 26×26

Result: Up–Scaling



Original

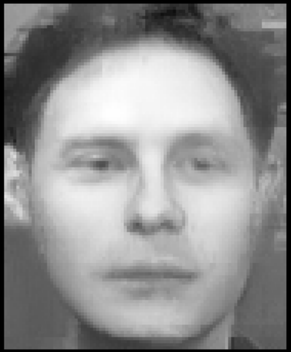
Measured

Blurred by $\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}$

4:1 decimation

& noise $\mathcal{N}(0,8\cdot\mathbf{I})$

Bilinear Inter.



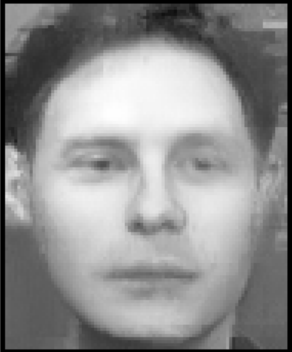
Reconstructed

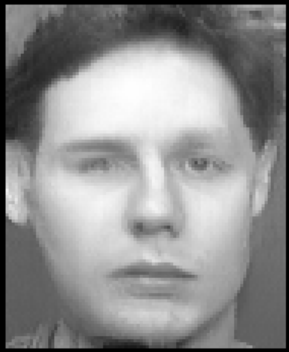
Block size 26×26

Changing the Training Set?

Conclusions:

- Both results are “correct” in the sense that they fit the measurements well, while each draws different information in order to complete the picture.
- Use as many examples as possible and the result will be more reliable.
- Using too many examples complicates the reconstruction algorithm. How much is enough?





Conclusion

Regularization is an extremely important tool in inverse problems

How to use it well?

The Bayesian approach gives us a constructive answer

But this means that we have to guess $P(\underline{x})$?

For 50 years we have been guessing $P(\underline{x})$ based on intuition: varying degrees of success

Can we do things differently?

Yes! WE CAN USE EXAMPLES

How?

Today we have seen some of the recent activity on this front

What next?

This idea has huge potential:
Relation to learning theory, probability sampling, efficient DB search, sensitivity to model errors, ...

Warning: Examples May Mislead

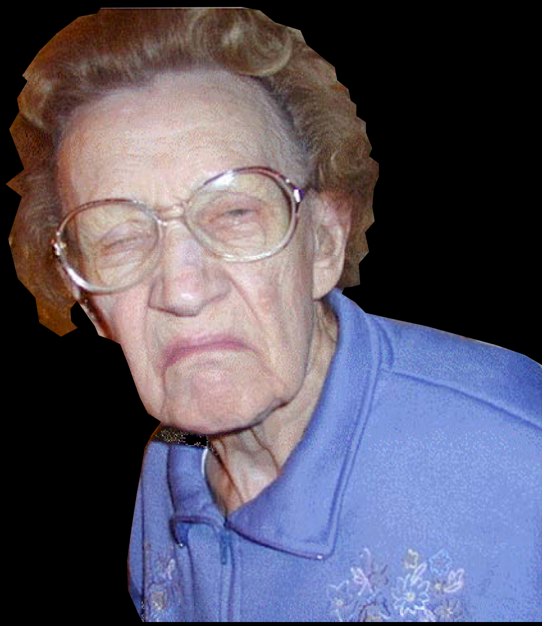
Too few examples can mislead,

Or … at least so we hope !!!!

Have you found your true love? Are you about to pop the question?

Ever wondered how will she look like and behave 40 years from now?

Should we use examples? How about looking at her mom?



[Kimmel, 2005]