Single Image Super-Resolution Using Sparse Representation*

Michael Elad

The Computer Science Department

The Technion – Israel Institute of technology

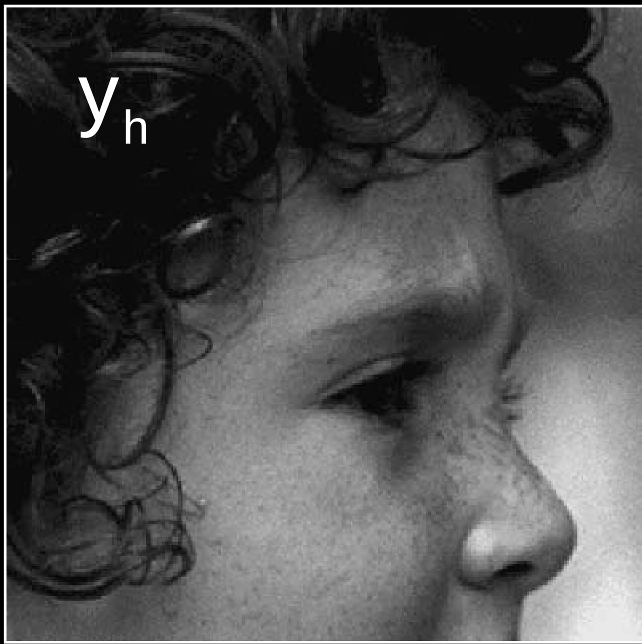
Haifa 32000, Israel

* Joint work with Roman Zeyde and Matan Protter

MS45: Recent Advances in Sparse and Non-local Image Regularization - Part III of III

Wednesday, April 14, 2010

The Super-Resolution Problem



WGN

v

Blur (**H**)
and
Decimation (**S**)

z_ℓ

$$z_\ell = \mathbf{SH}y_h + v$$

Our Task: Reverse the process – recover the high-resolution image from the low-resolution one

Single Image Super-Resolution

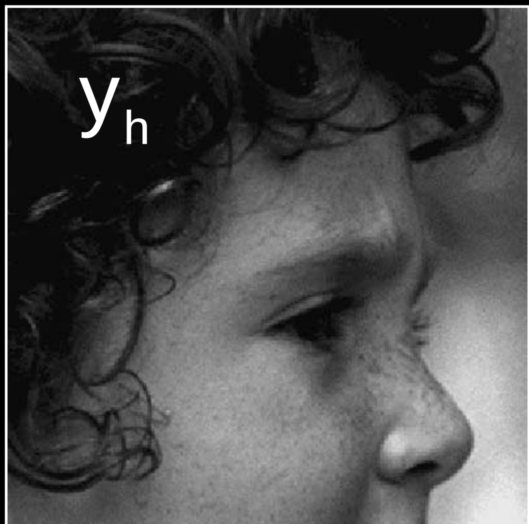
The reconstruction process should rely on:

- The given low-resolution image
- The knowledge of **S**, **H**, and statistical properties of v, and
- Image behavior (prior).

In our work:

- We use patch-based sparse and redundant representation based prior, and
- We follow the work by Yang, Wright, Huang, and Ma [CVPR 2008, IEEE-TIP – to appear], proposing an improved algorithm.

Core Idea (1) - Work on Patches



y_ℓ

We interpolate the low-res. Image in order to align the coordinate systems

Interp.
Q

z_ℓ

$$y_\ell = \mathbf{Q} z_\ell$$

Every patch from y_ℓ should go through a process of resolution enhancement. The improved patches are then merged together into the final image (by averaging).

Enhance Resolution

Core Idea (2) – Learning [Yang et. al. '08]

Enhance Resolution ?

We shall perform a sparse decomposition of the low-res. patch, w.r.t. a learned dictionary $\mathbf{A}_\ell$

We shall construct the high res. patch using the same sparse representation, imposed on a dictionary $\mathbf{A}_h$

and now, lets go into the details ...

The Sparse-Land Prior [Aharon & Elad, '06]

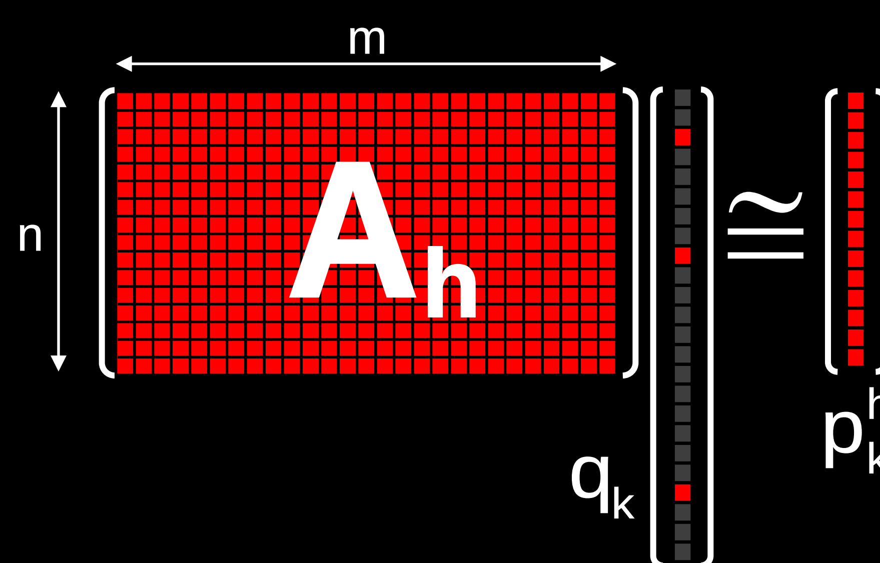
Extraction of a patch from y_h in location k is performed by

$$p_k^h = \mathbf{R}_k y_h$$

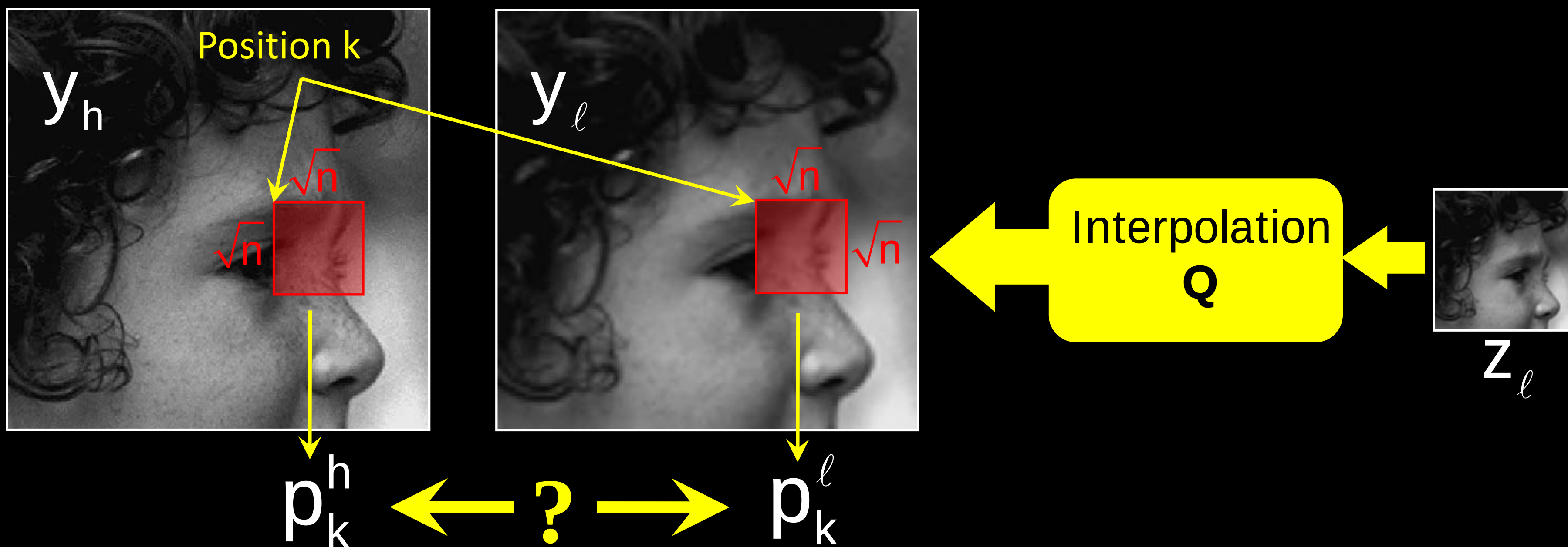
Model Assumption: Every such patch can be represented sparsely over the dictionary $\mathbf{A}_h$:

$$\forall p_k^h \ \exists q_k \quad \text{such that}$$

$$p_k^h \cong \mathbf{A}_h q_k \ \text{ and } \ \|q_k\|_0 << n$$



Low Versus High-Res. Patches

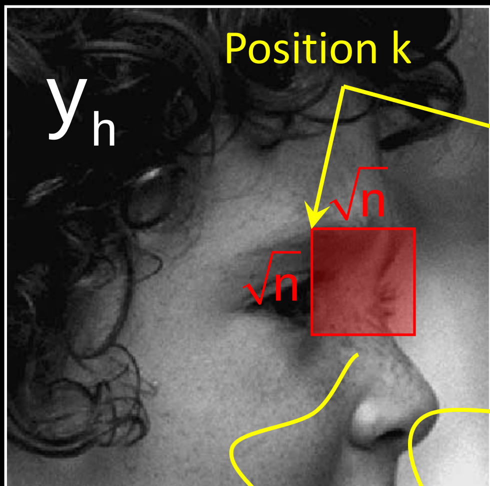


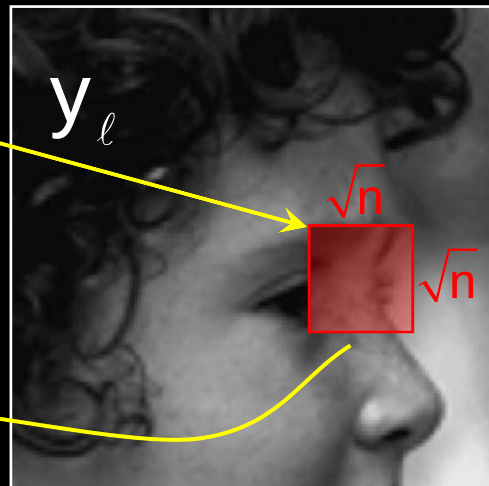
$$z_\ell = \mathbf{SH}y_h + v$$

$$\left\| \mathbf{L}p_k^h - p_k^\ell \right\|_2 \le \varepsilon$$

- **L** = blur + decimation + interpolation.
- This expression relates the low-res. patch to the high-res. one, based on

Low Versus High-Res. Patches





We interpolate the low-res. Image in order to align the coordinate systems

Interpolation

z_ℓ

$$\left\| \mathbf{L} p_k^h - p_k^\ell \right\|_2 \le \varepsilon$$

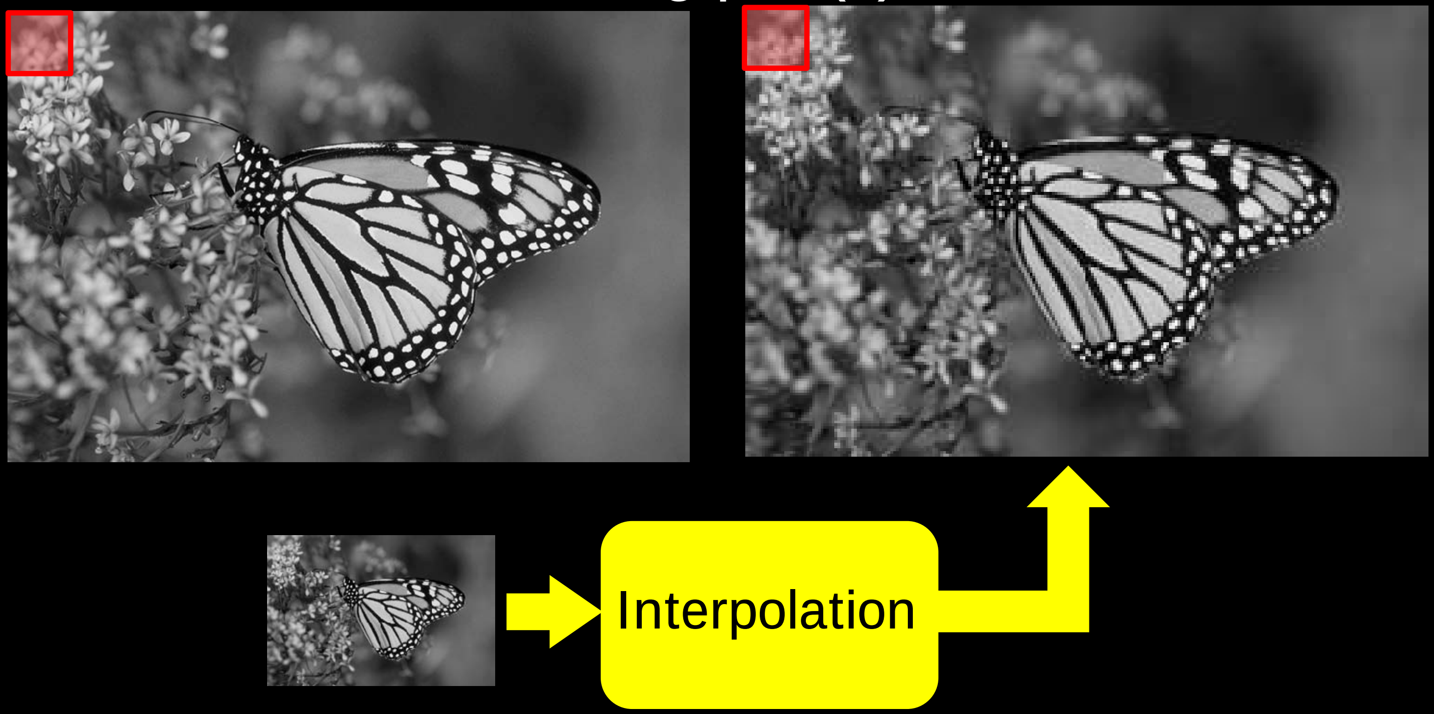
$$p_k^h \cong \mathbf{A}_h q_k$$

$$\left\| \underbrace{\mathbf{L}\mathbf{A}_h}_{\mathbf{A}_\ell} \boxed{q_k} - p_k^\ell \right\|_2 \le \varepsilon$$

q_k is ALSO the sparse representation of the low-resolution patch, with respect to the dictionary $\mathbf{LA}_h$.

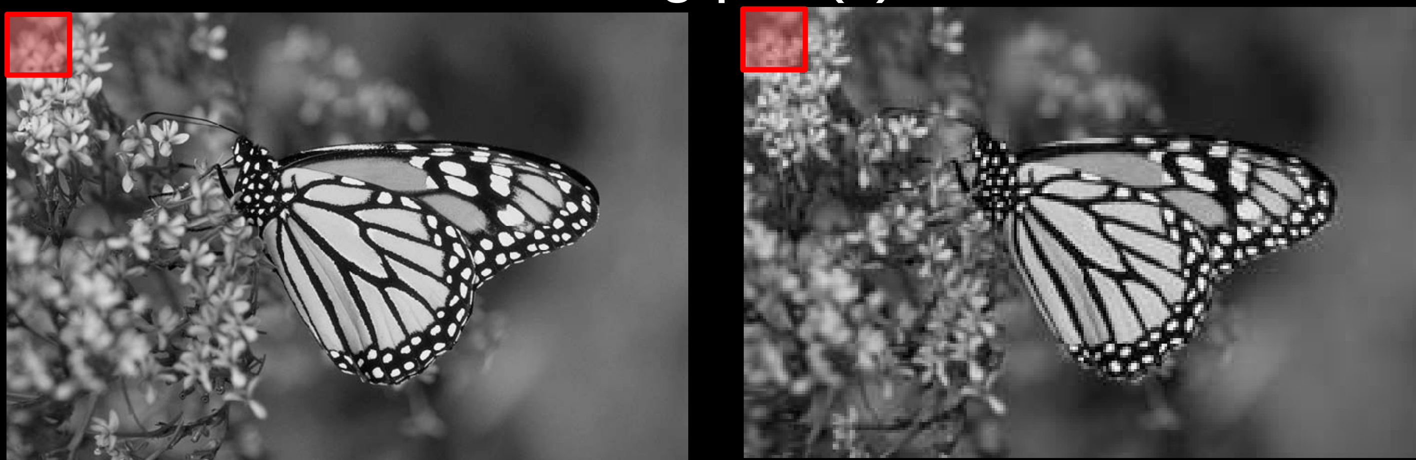
Training the Dictionaries – General

Training pair(s)



Training the Dictionaries – General

Training pair(s)



We obtain a set of matching patch-pairs

$$\left\{ \bar{p}_k^h, \bar{p}_k^\ell \right\}_k$$

Pre-process (low-res) → $\left\{ p_k^h \right\}_k$

Pre-process (high-res) → $\left\{ p_k^\ell \right\}_k$

These will be used for the dictionary training

Alternative: Bootstrapping

The given image to be scaled-up

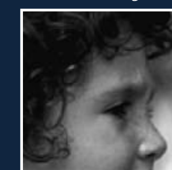
Simulate the degradation process

WGN

v

Blur (**H**) and Decimation (**S**)

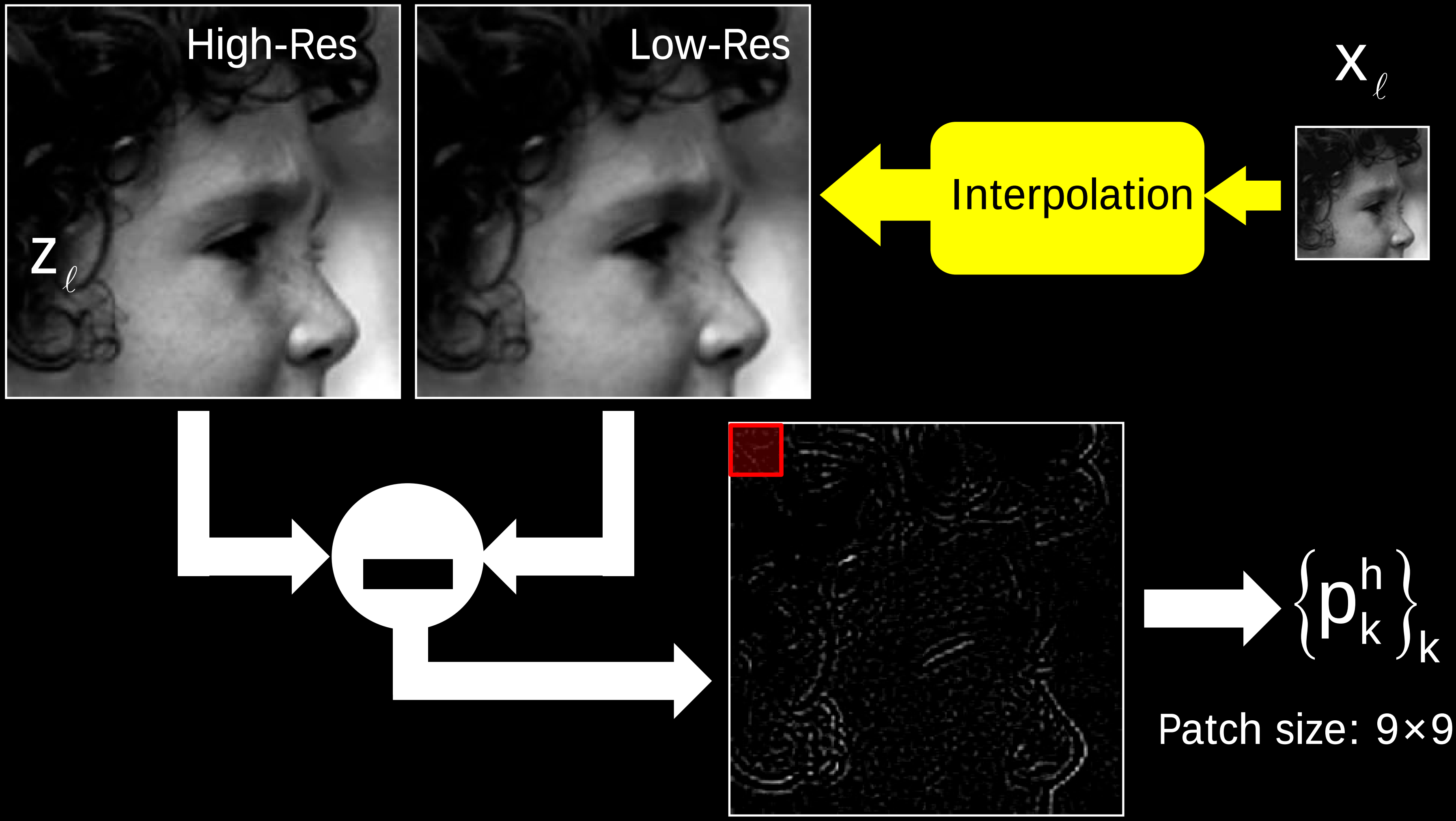
x_ℓ



$$x_\ell = \mathbf{SH}z_h + v$$

Use this pair of images to generate the patches for the training

Pre-Processing High-Res. Patches



Patch size: 9×9

Pre-Processing Low-Res. Patches

Interpolation

x_ℓ

* [0 1 -1]

* $[0\ 1\ {-1}]^T$

* [-1 2 -1]

* $[-1\ 2\ {-1}]^T$

We extract patches of size 9×9 from each of these images, and concatenate them to form one vector of length 324

$$\left\{ \bar{p}_k^\ell \right\}_k$$

Patch size: ~30

Pre-Processing Low-Res. Patches

Interpolation

x_{ℓ}

We extract patches of size 9×9 from each of these images, and concatenate them to form one vector of length 324

$$\left\{\bar{p}_k^{\ell}\right\}_k \rightarrow \text{Dimensionality Reduction By PCA} \rightarrow \left\{p_k^{\ell}\right\}_k$$

$$p_k^h = \mathbf{B}\bar{p}_k^h$$

Patch size: ~30

Training the Dictionaries: $\mathbf{A}_\ell$

$$\left\{p_k^\ell\right\}_k \Rightarrow \text{K-SVD} \Rightarrow \mathbf{A}_\ell$$

K-SVD

40 iterations, L=3

m=1000

N=30

For an image of size 1000×1000 pixels, there are ~12,000 examples to train on

Remember

Given a low-res. Patch to be scaled-up, we start the resolution enhancement by sparse coding it, to find q_k:

$$\min_{q_k} \left\| p_k^\ell - \mathbf{A}_\ell q_k \right\|_2 \ \text{ s.t. } \ \left\| q_k \right\|_0 \le L$$

Training the Dictionaries: $\mathbf{A}_h$

Remember

Given a low-res. Patch to be scaled-up, we start the resolution enhancement by sparse coding it, to find q_k:

$$\min_{q_k} \left\| p_k^\ell - \mathbf{A}_\ell q_k \right\|_2 \quad \text{s.t.} \quad \left\| q_k \right\|_0 \le L$$

And then, the high-res. Patch is obtained by

$$\hat{p}_k^h = \mathbf{A}_h q_k$$

Thus, $\mathbf{A}_h$ should be designed such that

$$\sum_k \left\| p_k^h - \mathbf{A}_h q_k \right\|_2^2 \xrightarrow[\mathbf{A}_h]{} \min$$

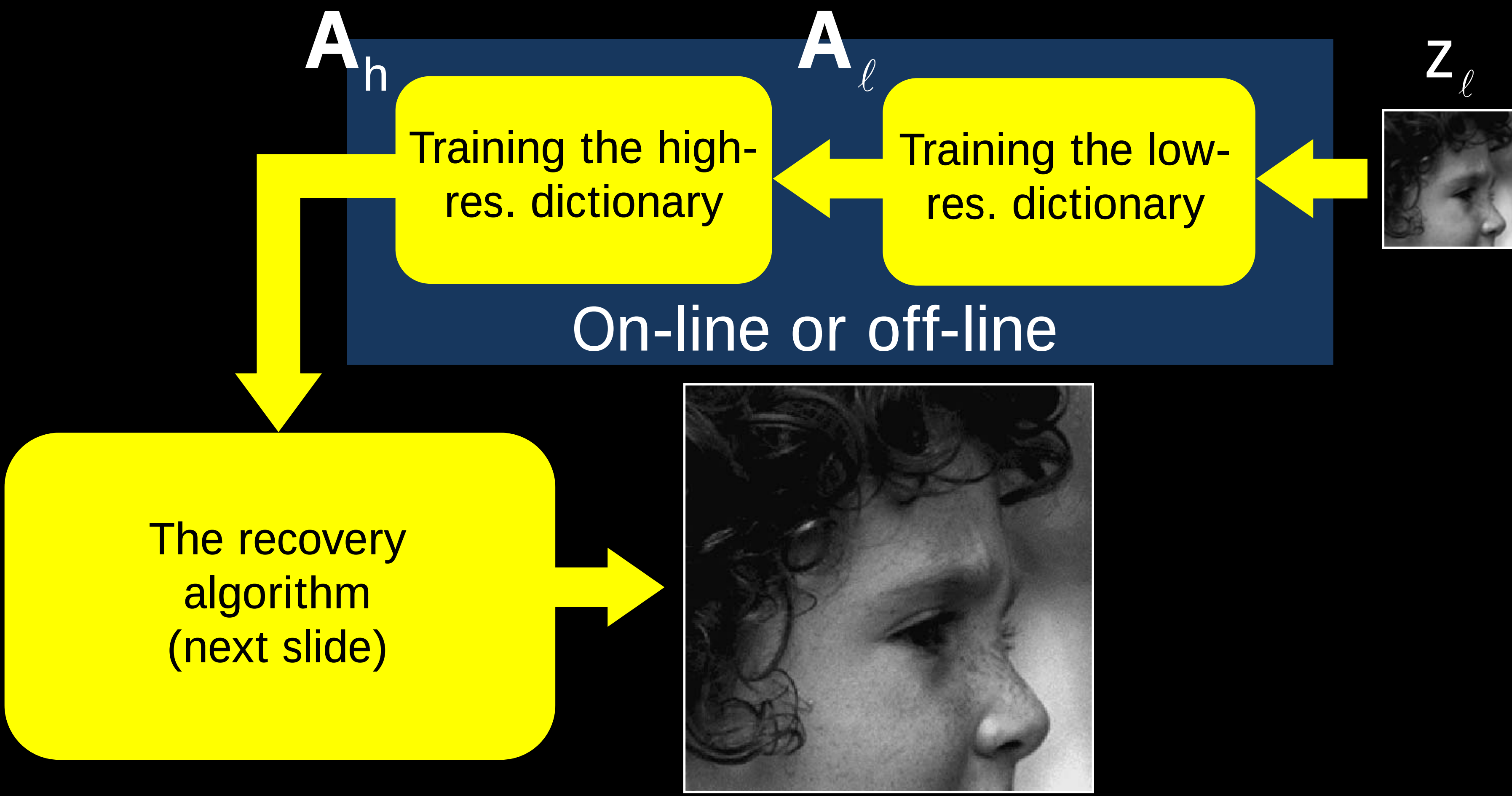
Training the Dictionaries: $\mathbf{A}_h$

$$\sum_k \left\| p_k^h - \mathbf{A}_h q_k \right\|_2^2 \xrightarrow[\mathbf{A}_h]{} \min$$

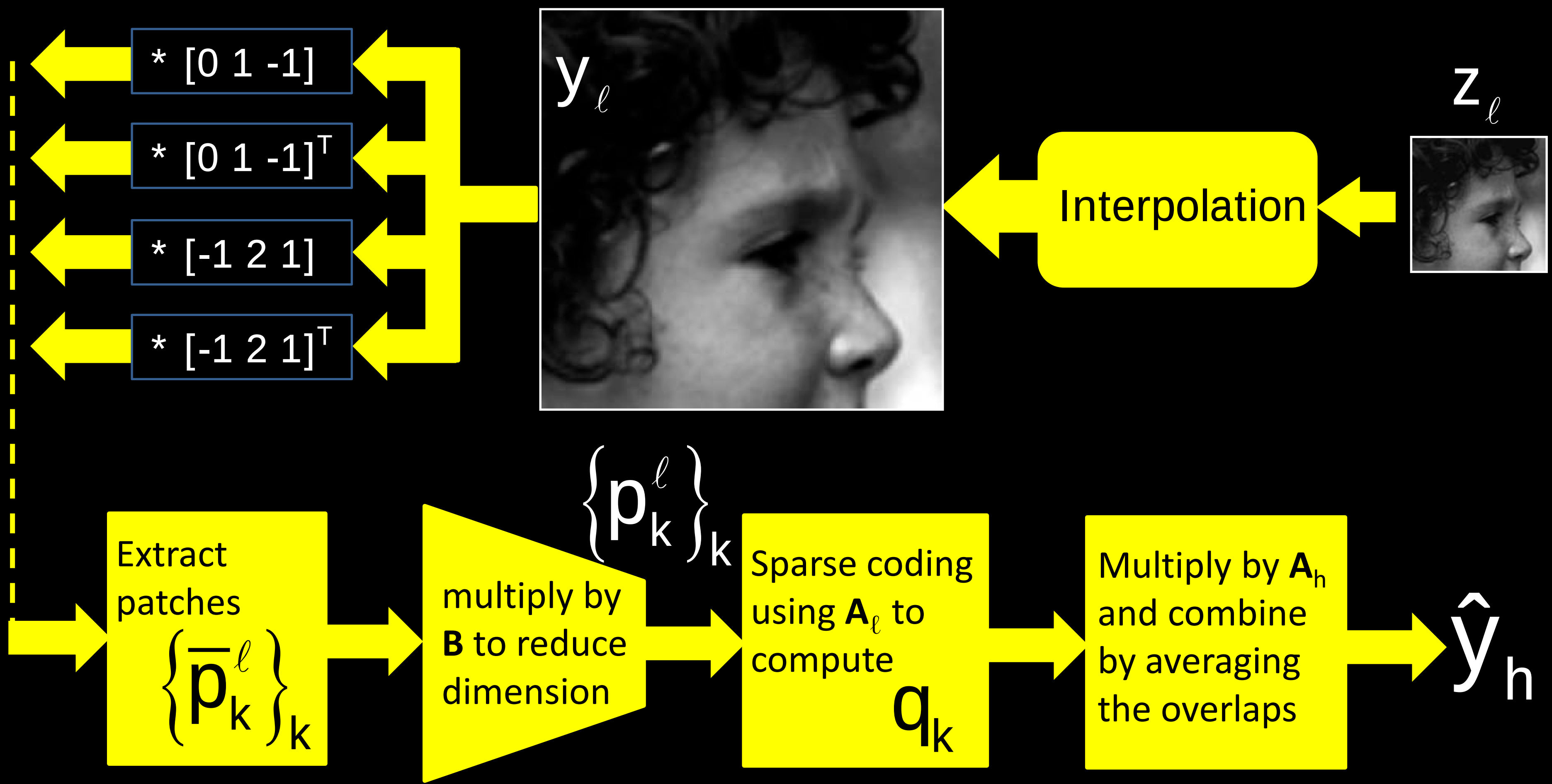
- However, this approach disregards the fact that the resulting high-res. patches are not used directly as the final result, but averaged due to overlaps between them.
- A better method (leading to better final scaled-up image) would be – Find $\mathbf{A}_h$ such that the following error is minimized:

$$\min_{\mathbf{A}_h} \left\| y_h - \hat{y}_h \right\|_2^2 = \min_{\mathbf{A}_h} \left\| y_h - \underbrace{y_\ell - \mathbf{W} \cdot \sum_k \mathbf{R}_k^T \mathbf{A}_h q_k}_{\text{The constructed image}} \right\|_2^2$$

Overall Block-Diagram



The Super-Resolution Algorithm



Relation to the Work by Yang et. al.

We use a direct approach to the training of $\mathbf{A}_h$

$\mathbf{A}_h$

Training the high-res. dictionary

$\mathbf{A}_\ell$

We interpolate to avoid coordinate ambiguity

Training the low-res. dictionary

$\mathbf{z}_\ell$

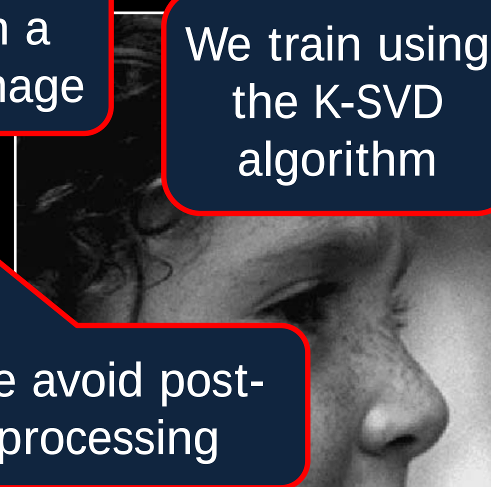
We use OMP for sparse-coding

We train on a difference-image

We train using the K-SVD algorithm

We can train on the given image or a training set

The recovery algorithm (previous slide)



We avoid post-processing

We reduce dimension prior to training

Relation to the Work by Yang et. al.

We use a direct approach to the training of $\mathbf{A}_h$

$\mathbf{A}_h$

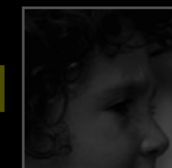
Training the high-res. dictionary

$\mathbf{A}_\ell$

We interpolate to avoid coordinate ambiguity

Training the low-res. dictionary

$\mathbf{z}_\ell$



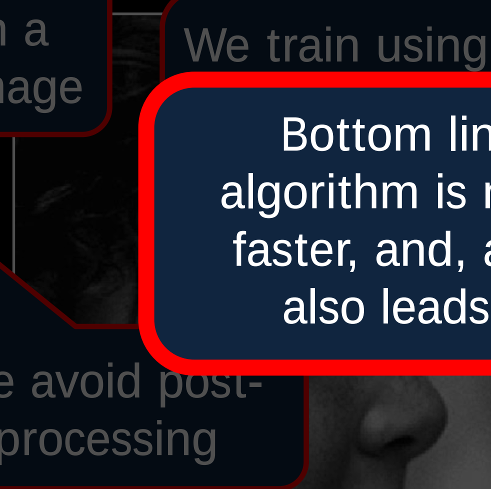
We use OMP for sparse-coding

We train on a difference-image

We train using

We can train on the given image

The recovery algorithm (previous slide)

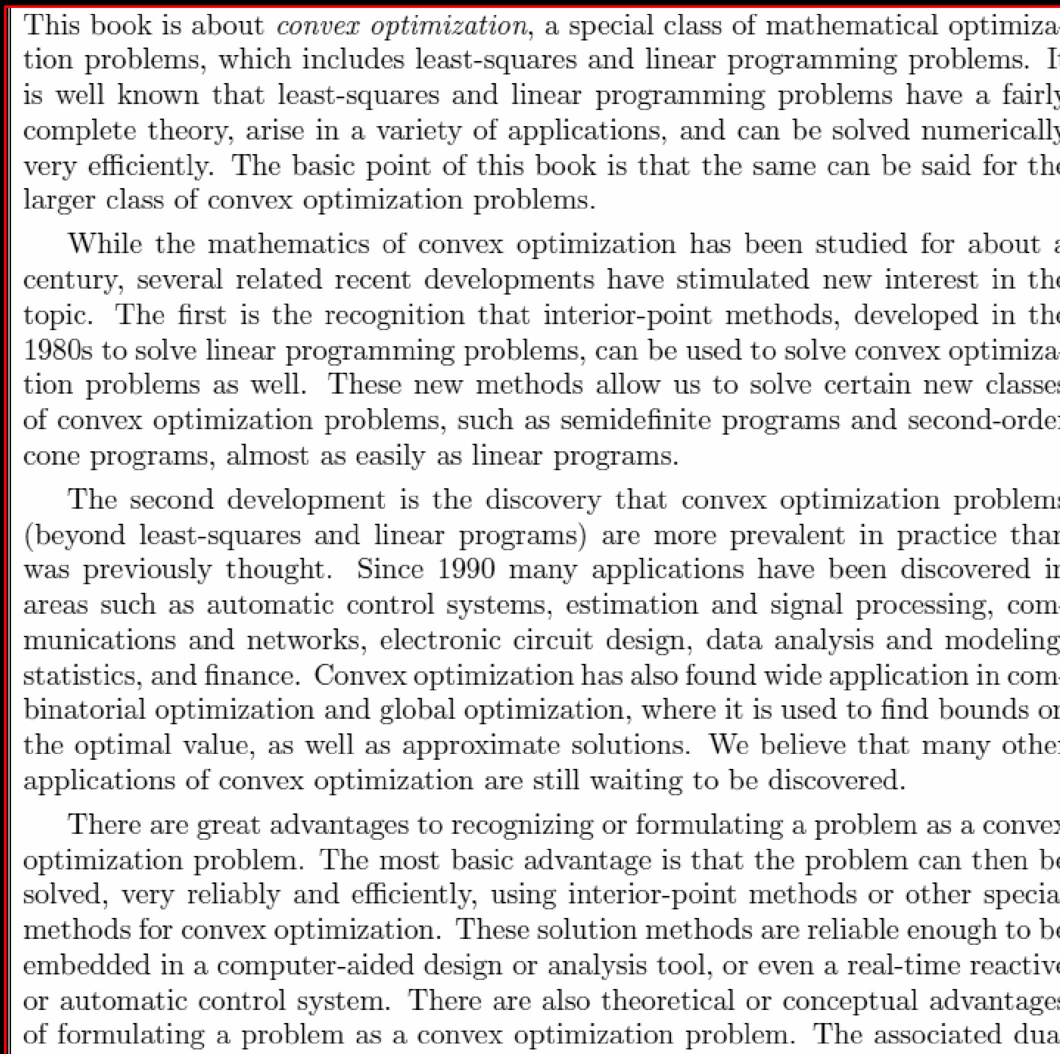


We avoid post-processing

to training

Bottom line: The proposed algorithm is much simpler, much faster, and, as we show next, it also leads to better results

Results (1) – Off-Line Training


This book is about *convex optimization*, a special class of mathematical optimization problems, which includes least-squares and linear programming problems. It is well known that least-squares and linear programming problems have a fairly complete theory, arise in a variety of applications, and can be solved numerically very efficiently. The basic point of this book is that the same can be said for the larger class of convex optimization problems.

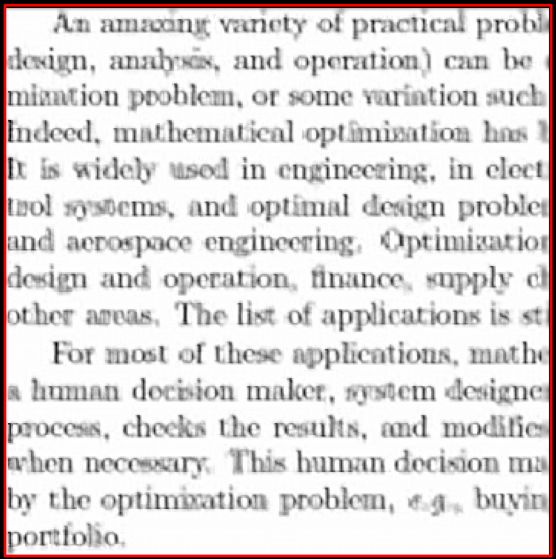
While the mathematics of convex optimization has been studied for about a century, several related recent developments have stimulated new interest in the topic. The first is the recognition that interior-point methods, developed in the 1980s to solve linear programming problems, can be used to solve convex optimization problems as well. These new methods allow us to solve certain new classes of convex optimization problems, such as semidefinite programs and second-order cone programs, almost as easily as linear programs.

The second development is the discovery that convex optimization problems (beyond least-squares and linear programs) are more prevalent in practice than was previously thought. Since 1990 many applications have been discovered in areas such as automatic control systems, estimation and signal processing, communications and networks, electronic circuit design, data analysis and modeling, statistics, and finance. Convex optimization has also found wide application in combinatorial optimization and global optimization, where it is used to find bounds on the optimal value, as well as approximate solutions. We believe that many other applications of convex optimization are still waiting to be discovered.

There are great advantages to recognizing or formulating a problem as a convex optimization problem. The most basic advantage is that the problem can then be solved, very reliably and efficiently, using interior-point methods or other special methods for convex optimization. These solution methods are reliable enough to be embedded in a computer-aided design or analysis tool, or even a real-time reactive or automatic control system. There are also theoretical or conceptual advantages of formulating a problem as a convex optimization problem. The associated dual

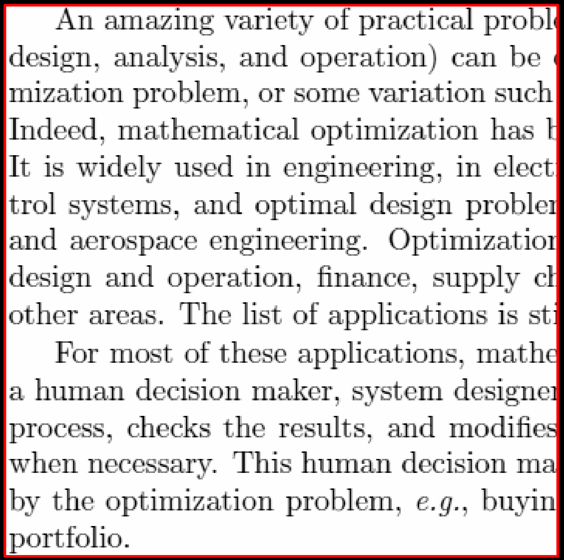
The training image: 717×717 pixels, providing a set of 54,289 training patch-pairs.

Results (1) – Off-Line Training


An amazing variety of practical probl
design, analysis, and operation) can be
mization problem, or some variation such
Indeed, mathematical optimization has b
It is widely used in engineering, in elect
trol systems, and optimal design probler
and aerospace engineering. Optimization
design and operation, finance, supply ch
other areas. The list of applications is sti
For most of these applications, mathe
a human decision maker, system designer
process, checks the results, and modifies
when necessary. This human decision ma
by the optimization problem, *e.g.*, buyin
portfolio.

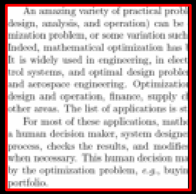
SR Result
PSNR=16.95dB

Ideal Image


An amazing variety of practical probl
design, analysis, and operation) can be
mization problem, or some variation such
Indeed, mathematical optimization has b
It is widely used in engineering, in elect
trol systems, and optimal design probler
and aerospace engineering. Optimization
design and operation, finance, supply ch
other areas. The list of applications is sti
For most of these applications, mathe
a human decision maker, system designer
process, checks the results, and modifies
when necessary. This human decision ma
by the optimization problem, *e.g.*, buyin
portfolio.


An amazing variety of practical probl
design, analysis, and operation) can be
mization problem, or some variation such
Indeed, mathematical optimization has b
It is widely used in engineering, in elect
trol systems, and optimal design probler
and aerospace engineering. Optimization
design and operation, finance, supply ch
other areas. The list of applications is sti
For most of these applications, mathe
a human decision maker, system designer
process, checks the results, and modifies
when necessary. This human decision ma
by the optimization problem, *e.g.*, buyin
portfolio.

Bicubic interpolation
PSNR=14.68dB


An amazing variety of practical probl
design, analysis, and operation) can be
mization problem, or some variation such
Indeed, mathematical optimization has b
It is widely used in engineering, in elect
trol systems, and optimal design probler
and aerospace engineering. Optimization
design and operation, finance, supply ch
other areas. The list of applications is sti
For most of these applications, mathe
a human decision maker, system designer
process, checks the results, and modifies
when necessary. This human decision ma
by the optimization problem, *e.g.*, buyin
portfolio.

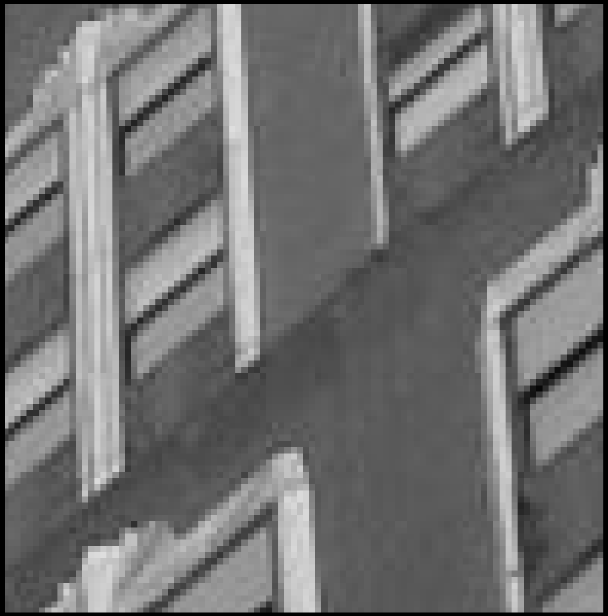
Given Image

Results (2) – On-Line Training

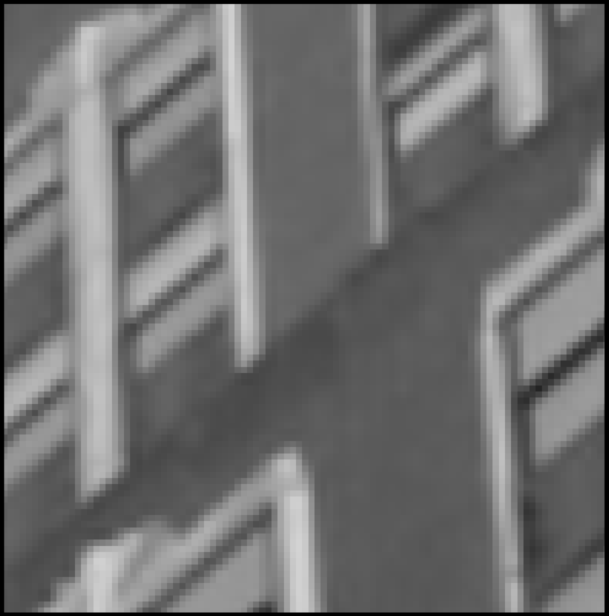
Given image

Scaled-Up (factor 2:1) using the proposed algorithm, PSNR=29.32dB (3.32dB improvement over bicubic)

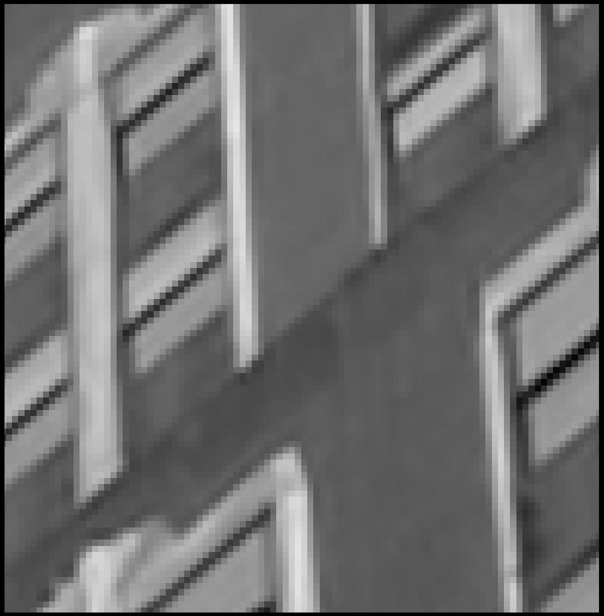
Results (2) – On-Line Training



The Original



Bicubic Interpolation



SR result

Results (2) – On-Line Training

The Original

Bicubic Interpolation

SR result

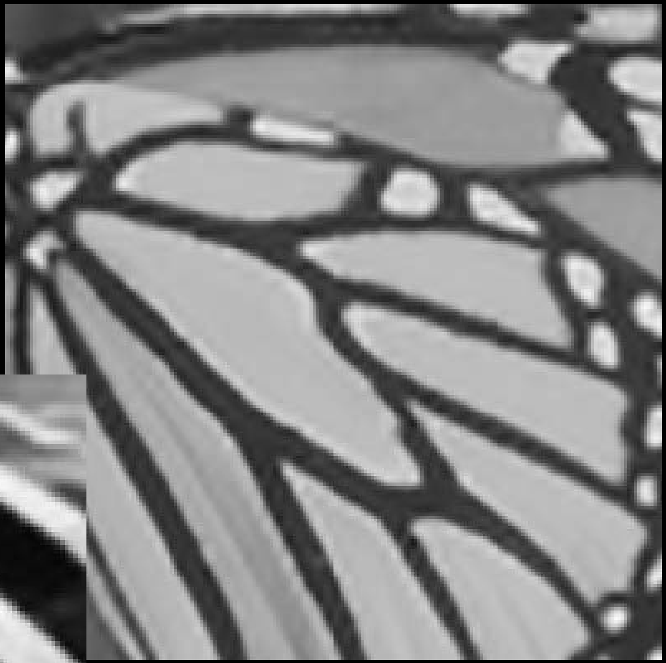
Comparative Results – Off-Line

	Bicubic		Yang et. al.		Our alg.	
	PNSR	SSIM	PSNR	SSIM	PSNR	SSIM
Barbara	26.24	0.75	26.39	0.76	26.77	0.78
Coastguard	26.55	0.61	27.02	0.64	27.12	0.66
Face	32.82	0.80	33.11	0.80	33.52	0.82
Foreman	31.18	0.91	32.04	0.91	33.19	0.93
Lenna	31.68	0.86	32.64	0.86	33.00	0.88
Man	27.00	0.75	27.76	0.77	27.91	0.79
Monarch	29.43	0.92	30.71	0.93	31.12	0.94
Pepper	32.39	0.87	33.33	0.87	34.05	0.89
PPT3	23.71	0.87	24.98	0.89	25.22	0.91
Zebra	26.63	0.79	27.95	0.83	28.52	0.84
Average	**28.76**	**0.81**	**29.59**	**0.83**	**30.04**	**0.85**

Comparative Results - Example

Yang et. al.

Comparative Results - Example



Ours

Summary

Single-image scale-up – an important problem, with many attempts to solve it in the past decade.

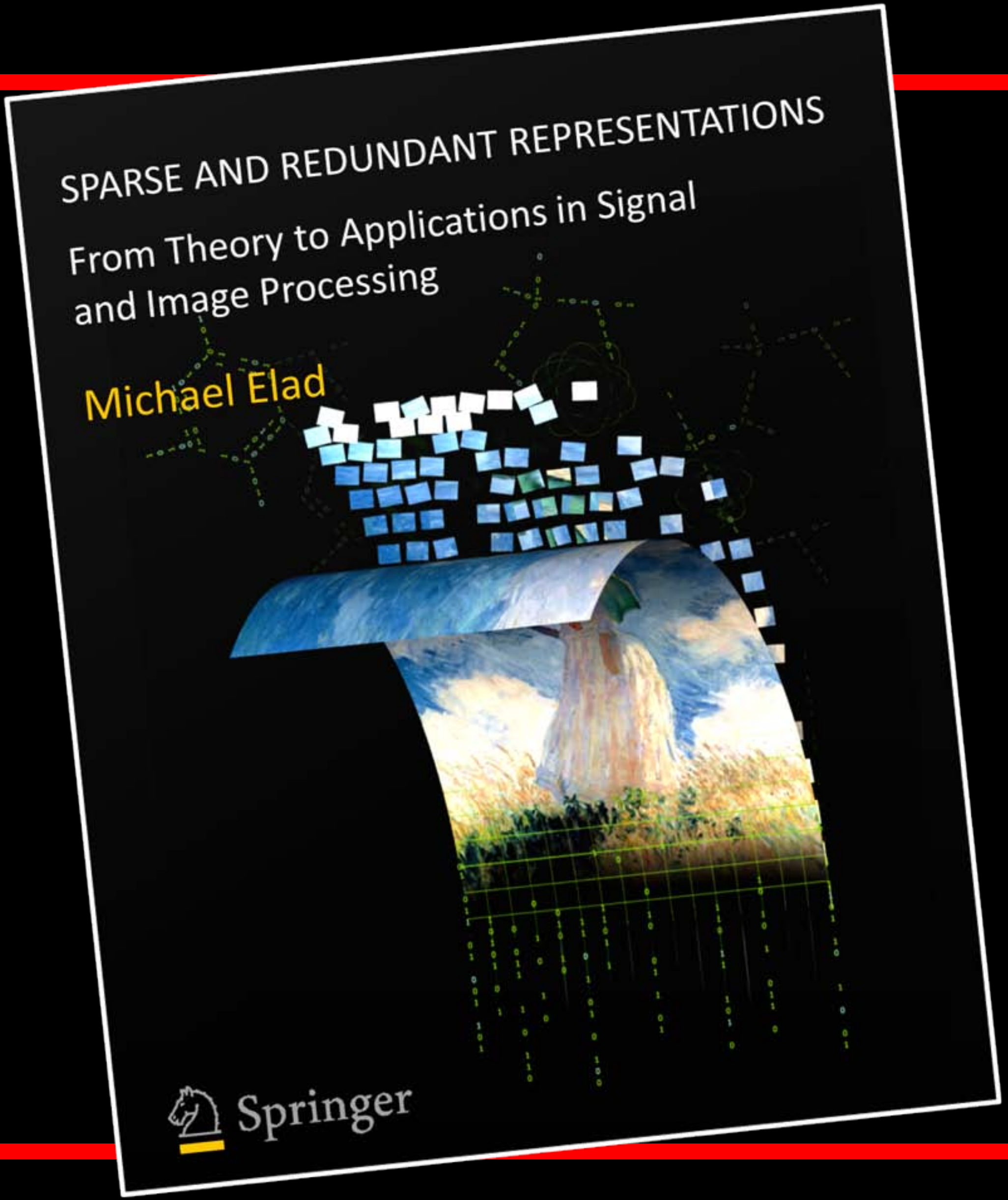
The rules of the game: use the given image, the known degradation, and a sophisticated image prior.

Yang et. al. [2008] – a very elegant way to incorporate sparse & redundant representation prior

We introduced modifications to Yang's work, leading to a simple, more efficient alg. with better results.

More work is required to improve further the results obtained.

A New Book



Thank You Very Much !!