# Sparse Coding for Image and Video Understanding


Jean Ponce

http://www.di.ens.fr/willow/

Willow team, LIENS, UMR 8548

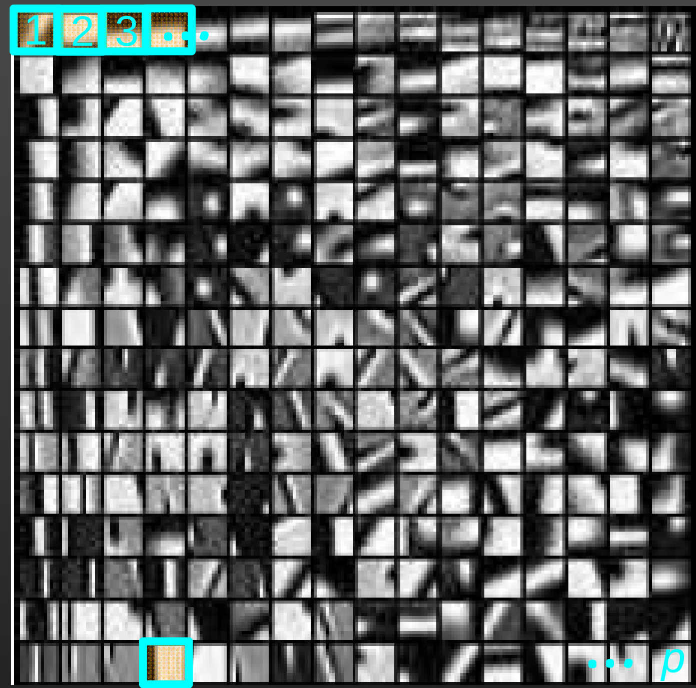
Ecole normale supérieure, Paris


Joint work with Julien Mairal, Francis Bach, Guillermo Sapiro and Andrew Zisserman

# Linear signal models

Signal: $x5\ U^m$

Dictionary: $D=[d_1,\dots,d_p]5\,U^{m\times p}$

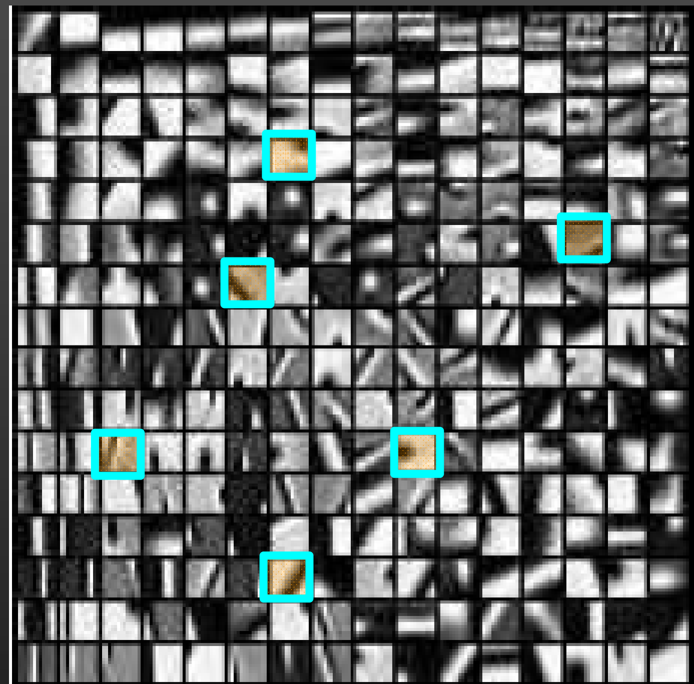



$$x \approx -_1 d_1 + -_2 d_2 + \dots + -_p d_p = D-,\ \text{with}\ -5\,U^p$$

# Sparse linear models

Signal: $x5\ U^m$

Dictionary: $D=[d_1,\ldots,d_p]5\ U^{m\times p}$



$$x \approx -_1 d_1 + -_2 d_2 + \ldots + -_p d_p = D-,\ \text{with } |-|_0 \ll p$$

(Olshausen and Field, 1997; Chen et al., 1999; Mallat, 1999; Elad and Aharon, 2006)
(Kavukcuoglu et al., 2009; Wright et al., 2009; Yang et al., 09; Boureau et al., 2010)

# Sparse coding and dictionary learning: A hierarchy of optimization problems

$$\min\_ \; 1/2 \; | x - D- |_2^2$$

$$\min\_ \; 1/2 \; | x - D- |_2^2 + \gg |-|_0$$

$$\min\_ \; 1/2 \; | x - D- |_2^2 + \gg \,¨(-)$$

$$\min_{D \in C, -_1, \ldots, -_n} \sum_{1 \leq i \leq n} [\, 1/2 \; | x_i - D-_i |_2^2 + \gg \,¨(-_i) \,]$$

$$\min_{D \in C, -_1, \ldots, -_n} \sum_{1 \leq i \leq n} [\, f\,(x_i, D, -_i) + \gg \,¨(-_i) \,]$$

$$\min_{D \in C, -_1, \ldots, -_n} \sum_{1 \leq i \leq n} [\, f\,(x_i, D, -_i) + \gg \sum_{1 \leq k \leq q} ¨(d_k) \,]$$

Least squares

Sparse coding

Dictionary learning

Learning for a task

Learning structures

# Outline

- Sparse linear models of image data
- Unsupervised dictionary learning
- Non-local sparse models for image restoration
- Learning discriminative dictionaries for image classification
- Task-driven dictionary learning and its applications
- Ongoing work

# Dictionary learning

- Given some loss function, e.g.,

$$L(x, D) = 1/2 \,|\, x - D– \,|_2^2 + » \,|–|_1$$

- One usually minimizes, given some data $x_i$, i = 1, ..., n, the empirical risk:

$$\min_D f_n(D) = \sum_{1 \leq i \leq n} L(x_i, D)$$

- But, one would really like to minimize the expected one, that is:

$$\min_D f(D) = H_x [\, L(x, D) \,]$$

(Bottou& Bousquet'08 $ Stochastic gradient descent)

# Learning Algorithms: Standard Framework

- Assumption: examples are drawn independently from an unknown probability distribution $P(x, y)$ that represents the rules of Nature.

**!! Short detour !!**

- In general $f^* \notin \mathcal{F}$.
- The best we can have is $f^*_{\mathcal{F}} \in \mathcal{F}$ that minimizes $E(f)$ inside $\mathcal{F}$.
- But $P(x, y)$ is unknown by definition.
- Instead we compute $f_n \in \mathcal{F}$ that minimizes $E_n(f)$.
  Vapnik-Chervonenkis theory tells us when this can work.

Léon bottou on large-scale learning..

# Learning with Approximate Optimization

Computing $f_n = \arg\min_{f \in \mathcal{F}} E_n(f)$ is often costly.

Since we already make lots of approximations, why should we compute $f_n$ exactly?

Let's assume our optimizer returns $\tilde{f}_n$ such that $E_n(\tilde{f}_n) < E_n(f_n) + \rho$.

For instance, one could stop an iterative optimization algorithm long before its convergence.

## Decomposition of the Error (i)

$$
\begin{aligned}
E(\tilde{f}_n) - E(f^*) &= E(f^*_{\mathcal{F}}) - E(f^*) && \text{Approximation error} \\
&+ E(f_n) - E(f^*_{\mathcal{F}}) && \text{Estimation error} \\
&+ E(\tilde{f}_n) - E(f_n) && \text{Optimization error}
\end{aligned}
$$

Problem:

Choose $\mathcal{F}$, $n$, and $\rho$ to make this as small as possible,

subject to budget constraints $\left\{ \begin{array}{l} \text{maximal number of examples } n \\ \text{maximal computing time } T \end{array} \right.$

Léon bottou on large-scale learning..

# Decomposition of the Error (ii)

**Approximation error bound:** (Approximation theory)
– decreases when $\mathcal{F}$ gets larger.

**Estimation error bound:** (Vapnik-Chervonenkis theory)
– decreases when $n$ gets larger.
– increases when $\mathcal{F}$ gets larger.

**Optimization error bound:** (Vapnik-Chervonenkis theory plus tricks)
– increases with $\rho$.

**Computing time $T$:** (Algorithm dependent)
– decreases with $\rho$
– increases with $n$
– increases with $\mathcal{F}$

Léon bottou on large-scale learning..

## Small-scale vs. Large-scale Learning

We can give *rigorous definitions.*

- **Definition 1:**
  We have a **small-scale learning** problem when the **active budget constraint is the number of examples $n$.**

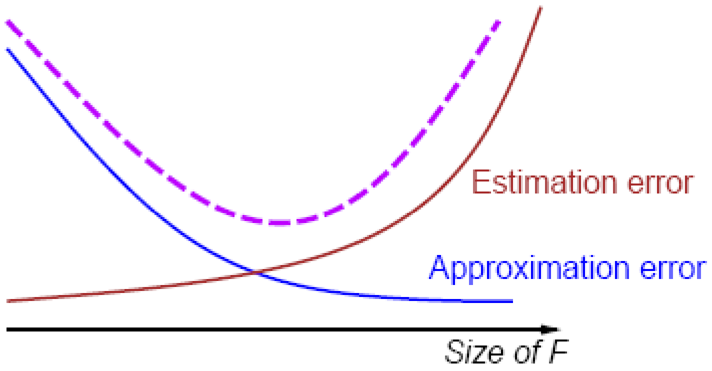
- **Definition 2:**
  We have a **large-scale learning** problem when the **active budget constraint is the computing time $T$.**

Léon bottou on large-scale learning..

# Small-scale Learning

The active budget constraint is the number of examples.

- To reduce the estimation error, take $n$ as large as the budget allows.
- To reduce the optimization error to zero, take $\rho = 0$.
- We need to adjust the size of $\mathcal{F}$.




Léon bottou on large-scale learning..

## Large-scale Learning

The active budget constraint is the computing time.

- More complicated tradeoffs.
  The computing time depends on the three variables: $\mathcal{F}$, $n$, and $\rho$.

- Example.
  If we choose $\rho$ small, we decrease the optimization error. But we must also decrease $\mathcal{F}$ and/or $n$ with adverse effects on the estimation and approximation errors.

- The exact tradeoff depends on the optimization algorithm.

- We can compare optimization algorithms rigorously.

Léon bottou on large-scale learning..

# Online sparse matrix factorization
(Mairal, Bach, Ponce, Sapiro, ICML'09, JMLR'10)

## Problem:

$$\min_{D \in C, -_4, \ldots, -_q} \sum_{1 \leq i \leq n} [\, 1/2 \mid x_] - D-_] \mid_2^2 + \gg \mid -_] \mid_1 \,]$$

$$\min_{D \in C,\, A} [\, 1/2 \mid X - DA \mid_F^2 + \gg \mid A \mid_1 \,]$$

## Algorithm:

Iteratively draw one random training sample $x_t$ and minimize the quadratic surrogate function:

$$g_t ( D ) = 1/t \sum_{1 \leq i \leq t} [\, 1/2 \mid x_] - D-_] \mid_2^2 + \gg \mid -_] \mid_1 \,]$$

(Lars/Lasso for sparse coding, block-coordinate descent with warm restarts for dictionary updates, mini-batch extensions, etc.)

# Online sparse matrix factorization

(Mairal, Bach, Ponce, Sapiro, ICML'09, JMLR'10)

**Proposition:**

Under mild assumptions, $D_t$ converges with probability one to a stationary point of the dictionary learning problem.

Proof: Convergence of empirical processes (van der Vaart'98) and, a la Bottou'98, convergence of quasi martingales (Fisk'65).

**Extensions:**

- Non negative matrix factorization (Lee & Seung'01)
- Non negative sparse coding (Hoyer'02)
- Sparse principal component analysis (Jolliffe et al.'03; Zou et al.'06; Zass& Shashua'07; d'Aspremont et al.'08; Witten et al.'09)

# Performance evaluation

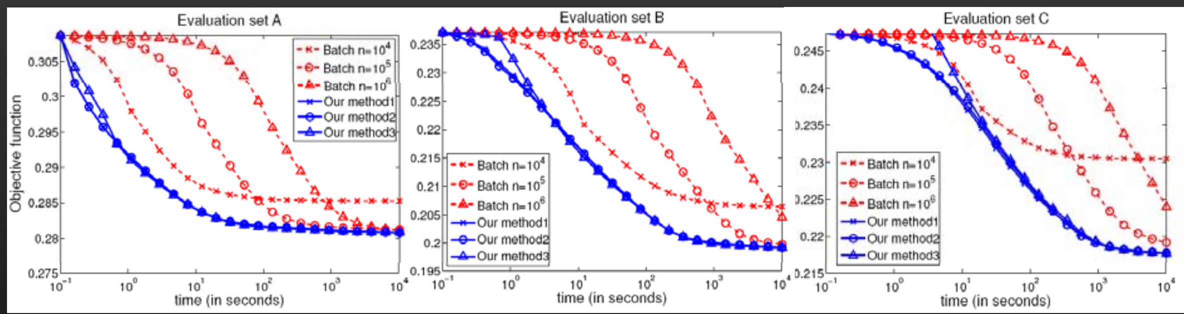
Three datasets constructed from 1,250,000 Pascal'06 patches (1,000,000 for training, 250,000 for testing):

- A: 8§ 8 b&w patches, 256 atoms.
- B: 12§ 16§ 3 color patches, 512 atoms.
- C: 16§ 16 b&w patches, 1024 atoms.

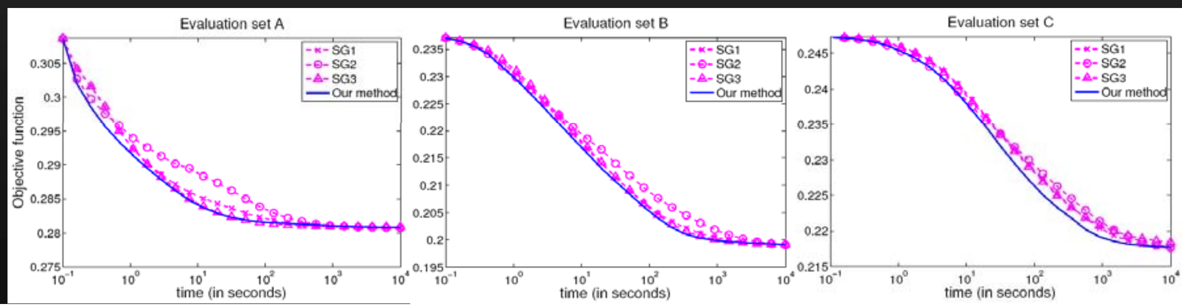
Two variants of our algorithm:

- Online version with different choices of parameters.
- Batch version on different subsets of training data.

Online vs batch




Online vs stochastic gradient descent



# Sparse PCA: Adding sparsity on the atoms

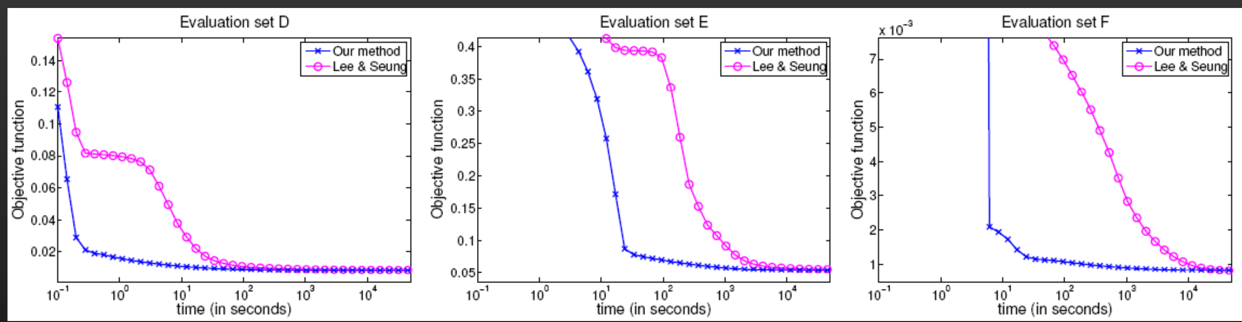
Three datasets:

- D: 2429 19§19 images from MIT-CBCL #1.
- E: 2414 192§168 images from extended Yale B.
- F: 100,000 16§16 patches from Pascal VOC'06.

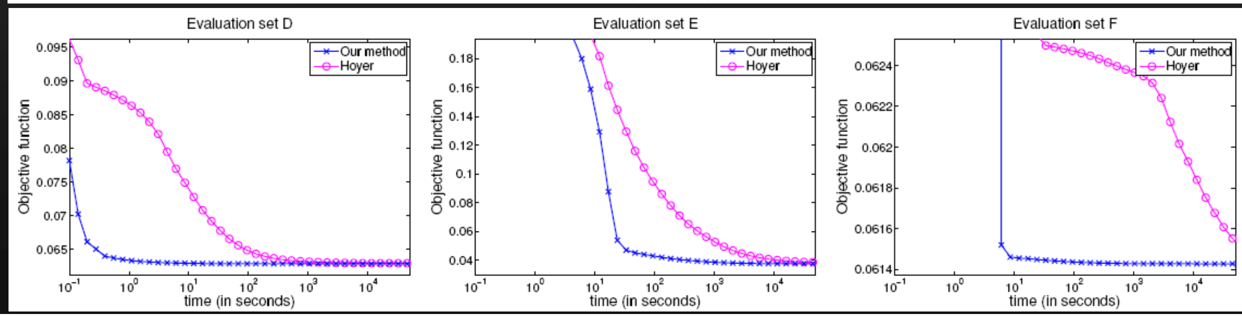
Three implementations:

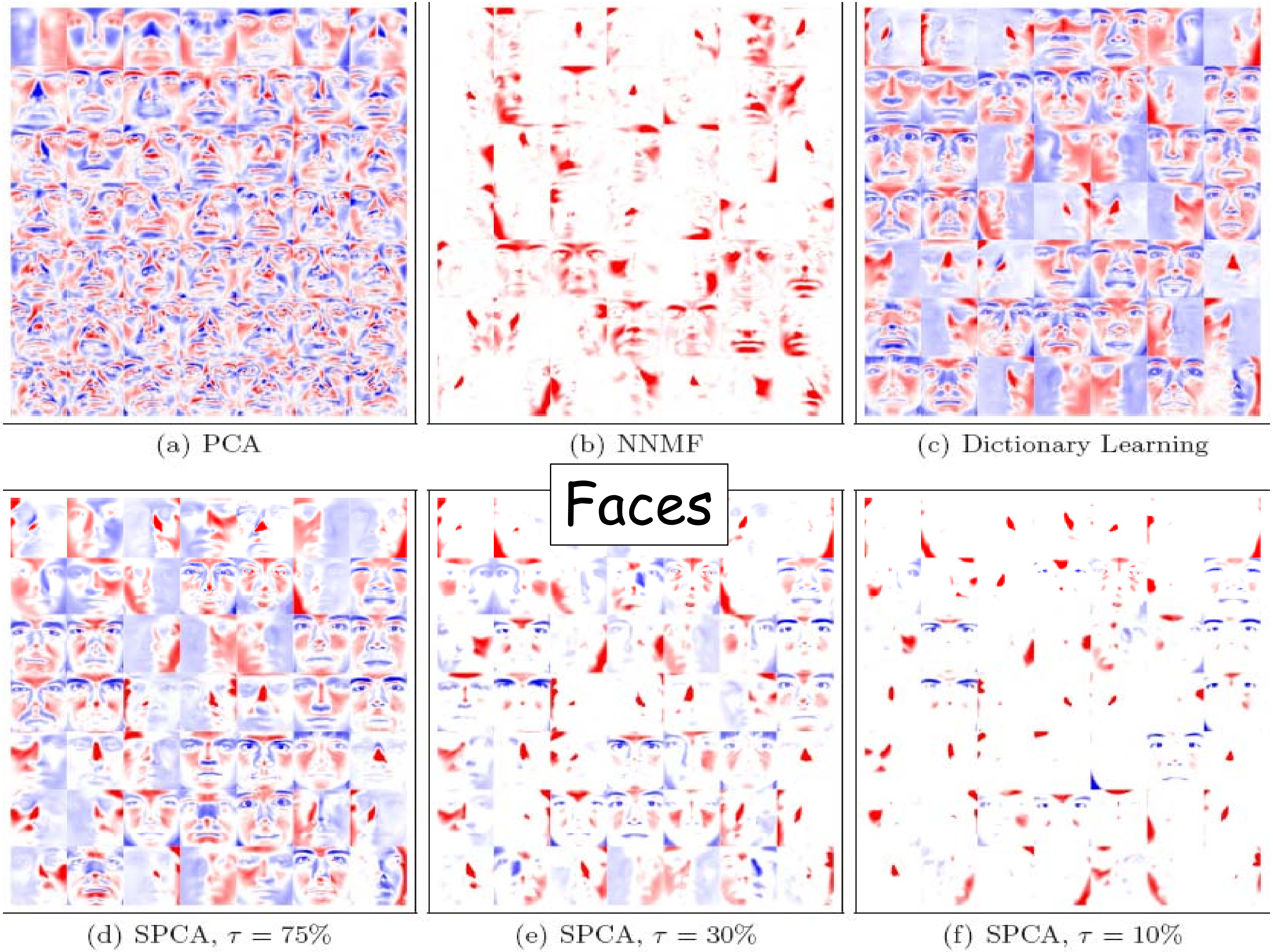
- Hoyer's Matlab implementation of NNMF (Lee & Seung'01).
- Hoyer's Matlab implementation of NNSC (Hoyer'02).
- Our C++/Matlab implementation of SPCA (elastic net on D).

SPCA vs NNMF




SPCA vs NNSC






(a) PCA

(b) NNMF

(c) Dictionary Learning

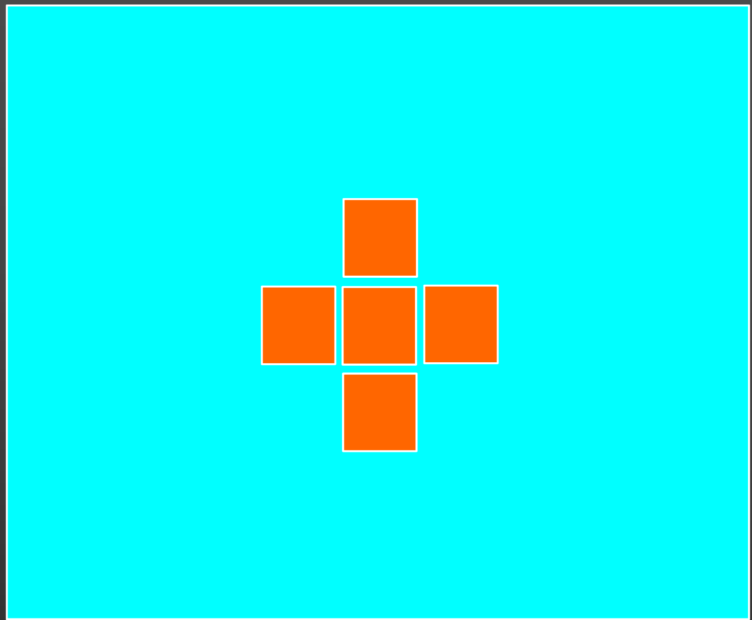
(d) SPCA, $\tau = 75\%$

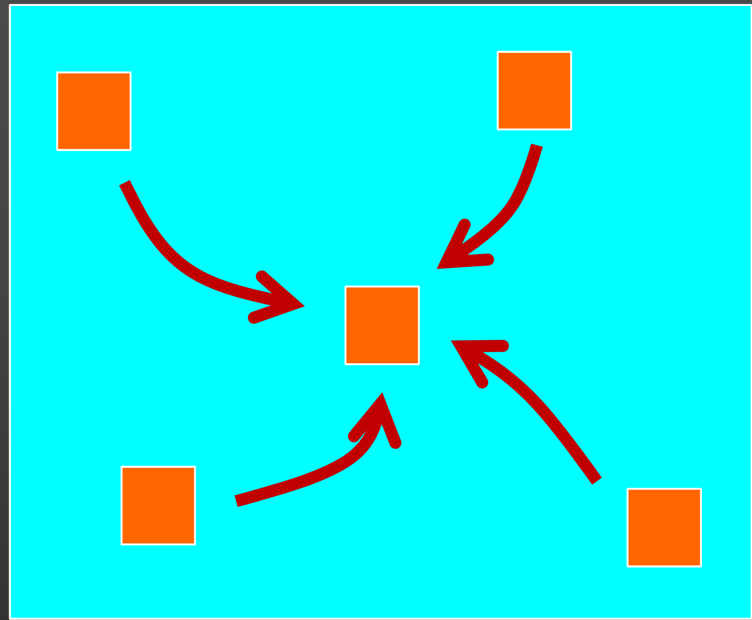
(e) SPCA, $\tau = 30\%$

(f) SPCA, $\tau = 10\%$


Inpainting a 12MP image with a dictionary learned from $7 \times 10^6$ patches (Mairal et al., 2009)
g mountain
alley from th
lf a dread o
orning cam

# State of the art in image denoising





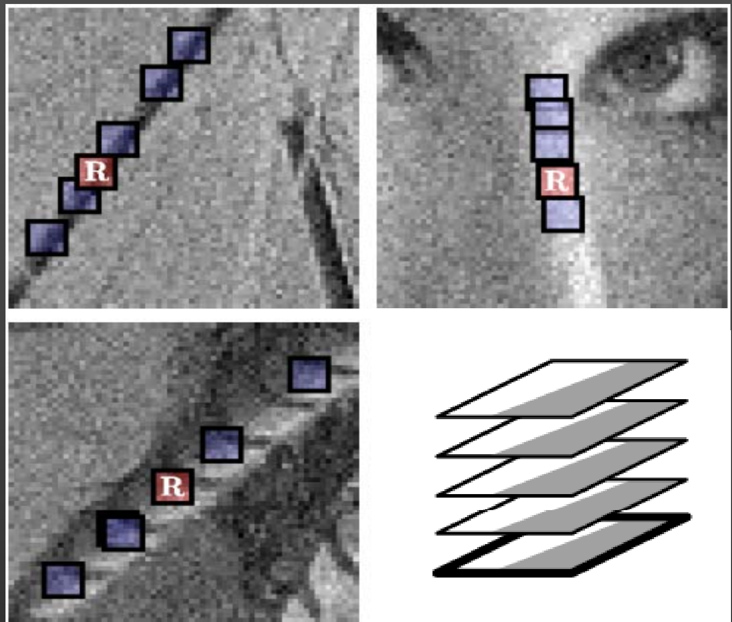
Non-local means filtering (Buades et al.'05)

Dictionary learning for denoising (Elad & Aharon'06; Mairal, Elad & Sapiro'08)

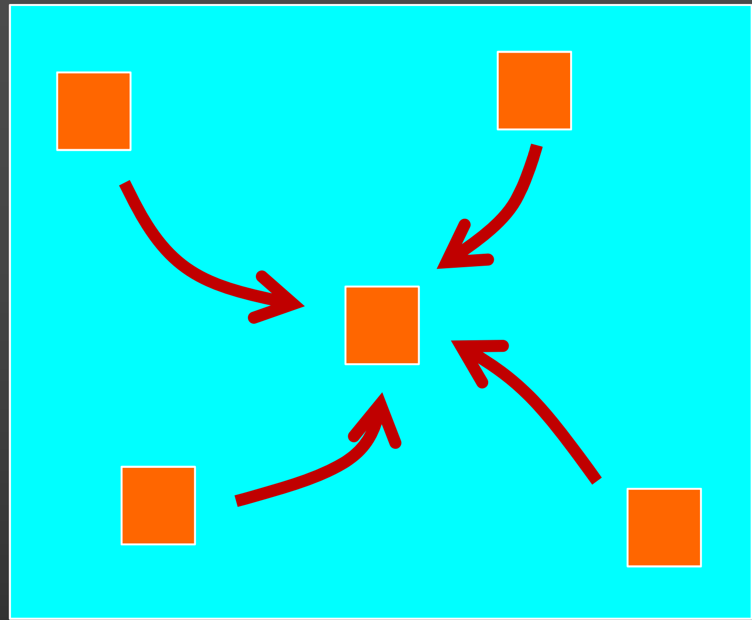
$$\min_{D \in C, \alpha_1, \ldots, \alpha_n} \sum_{1 \le i \le n} \left[ 1/2 \, | x_i - D\alpha_i |_2^2 + \lambda \, |\alpha_i|_1 \right]$$

$$x = 1/n \sum_{1 \le i \le n} R_i D \alpha_i$$

# State of the art in image denoising




BM3D (Dabov et al.'07)



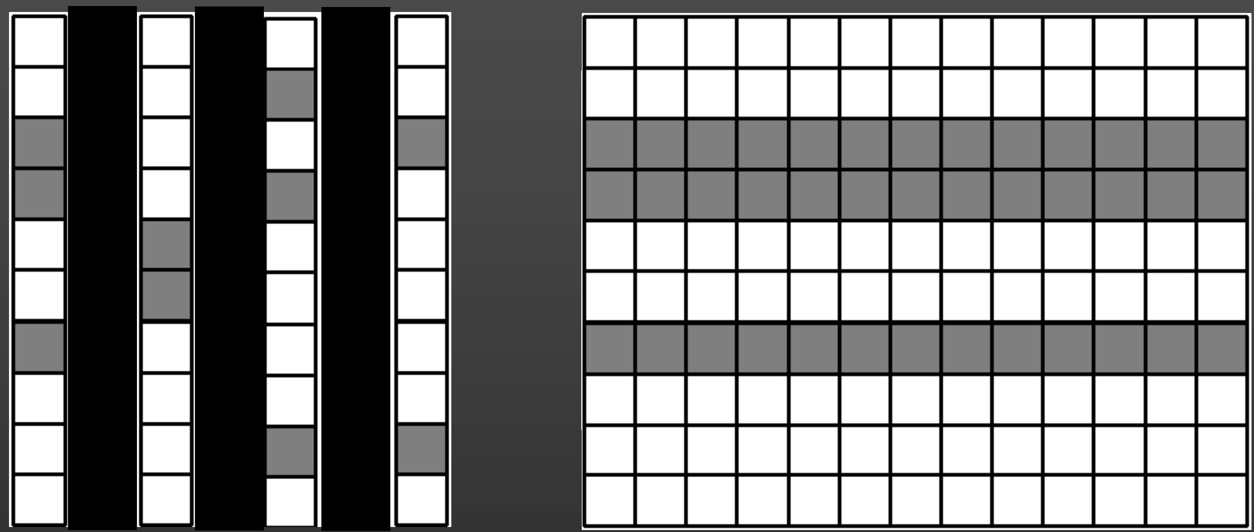
Non-local means filtering (Buades et al.'05)

Dictionary learning for denoising (Elad & Aharon'06; Mairal, Elad & Sapiro'08)

$$\min_{D\in C,-_4,\ldots,-_q} \sum_{1\le i\le n} [\ 1/2\ |\ x_i - D-_1\ |_2^2 + \text{»}\ |-_1|_1\ ]$$

$$x = 1/n \sum_{1\le i\le n} R_i D-_1$$

# Non-local sparse models for image restoration (Mairal, Bach, Ponce, Sapiro, Zisserman, ICCV'09)



Sparsity vs Joint sparsity

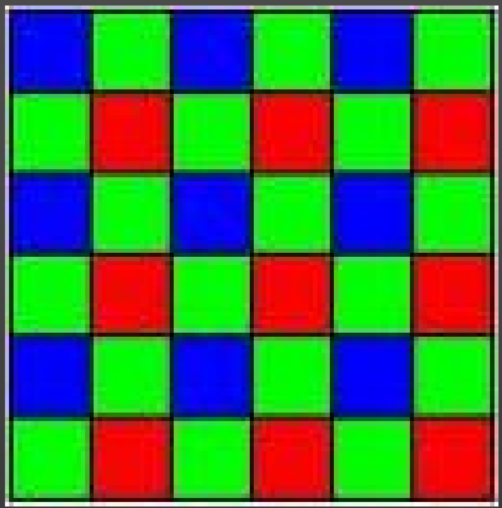
$$\min_{\substack{D \in C \\ A_1,\ldots,A_n}} \sum_i \Big[\sum_{j \in S_i} 1/2 \, \| x_j - D\alpha_{ij} \|_F^2\Big] + \lambda \, \|A_i\|_{p,q}$$

$$\|A\|_{p,q} = \sum_{1 \leq i \leq k} \|\alpha^i\|_q^p \quad (p,q) = (1,2) \text{ or } (0,\infty)$$

| σ | [23] | [25] | [12] | [8] | SC | LSC | LSSC |
|---|---|---|---|---|---|---|---|
| 5 | 37.05 | 37.03 | 37.42 | 37.62 | 37.46 | 37.66 | **37.67** |
| 10 | 33.34 | 33.11 | 33.62 | 34.00 | 33.76 | 33.98 | **34.06** |
| 15 | 31.31 | 30.99 | 31.58 | 32.05 | 31.72 | 31.99 | **32.12** |
| 20 | 29.91 | 29.62 | 30.18 | 30.73 | 30.29 | 30.60 | **30.78** |
| 25 | 28.84 | 28.36 | 29.10 | 29.72 | 29.18 | 29.52 | **29.74** |
| 50 | 25.66 | 24.36 | 25.61 | 26.38 | 25.83 | 26.18 | **26.57** |
| 100 | 22.80 | 21.36 | 22.10 | 23.25 | 22.46 | 22.62 | **23.39** |

PSNR comparison between our method (LSSC) and Portilla et al.'03 [23]; Roth & Black'05 [25]; Elad& Aharon'06 [12]; and Dabov et al.'07 [8].

# Demosaicking experiments



Bayer pattern

LSC

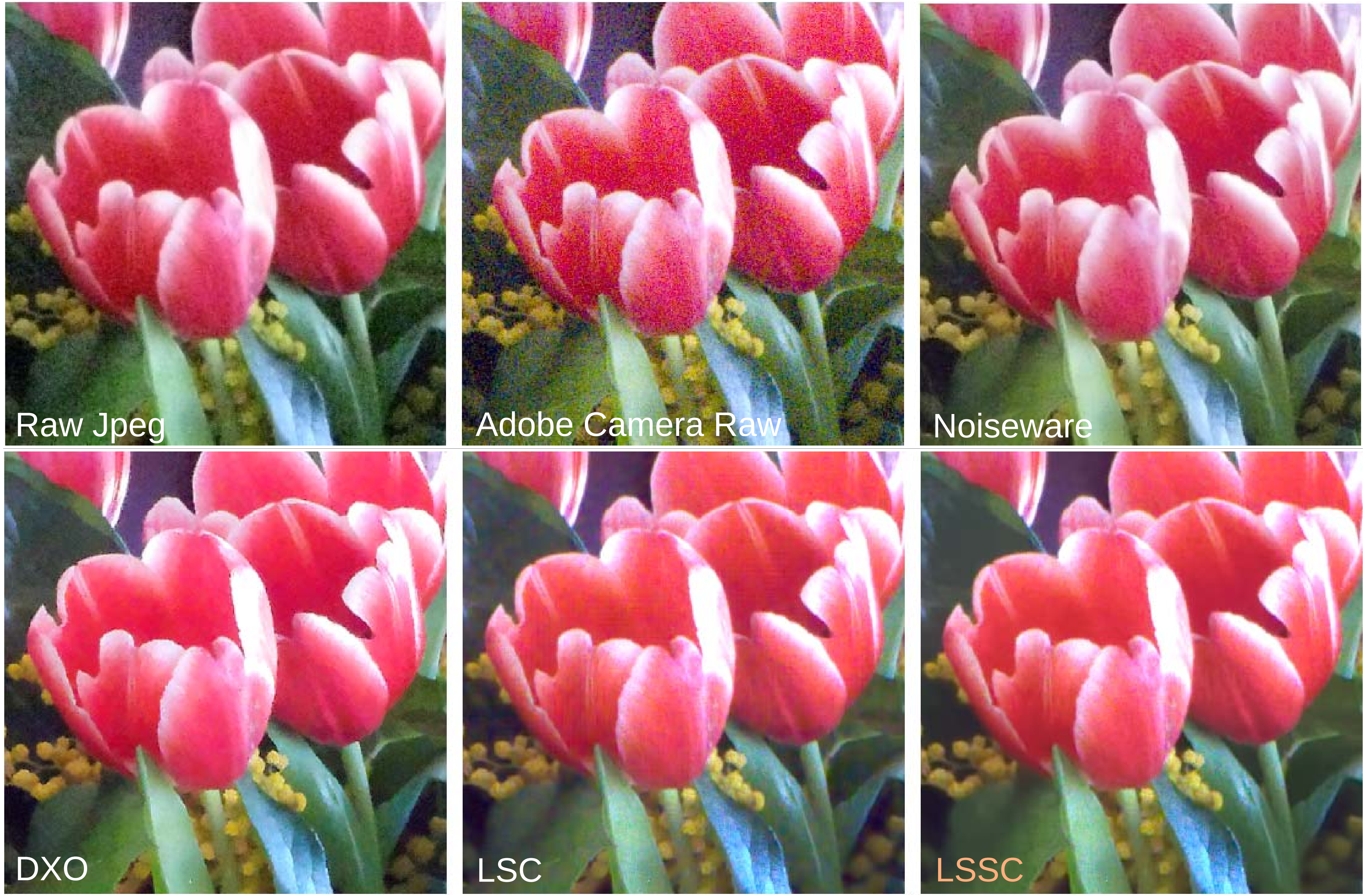
LSSC

| Im. | AP | DL | LPA | SC | LSC | LSSC |
|---|---|---|---|---|---|---|
| 1 | 37.84 | 38.46 | 40.47 | 40.84 | 40.92 | **41.36** |
| 2 | 39.64 | 40.89 | 41.36 | 41.76 | 42.03 | **42.24** |
| 3 | 41.40 | 42.66 | 43.47 | 43.15 | 43.92 | **44.24** |
| ... | ... | ... | ... | ... | ... | ... |
| 23 | 41.93 | 43.22 | 43.92 | 43.47 | 43.93 | **44.34** |
| 24 | 34.74 | 35.55 | 35.44 | 35.59 | 35.85 | **35.89** |
| **Av.** | 39.21 | 40.05 | 40.52 | 40.88 | 41.13 | **41.39** |

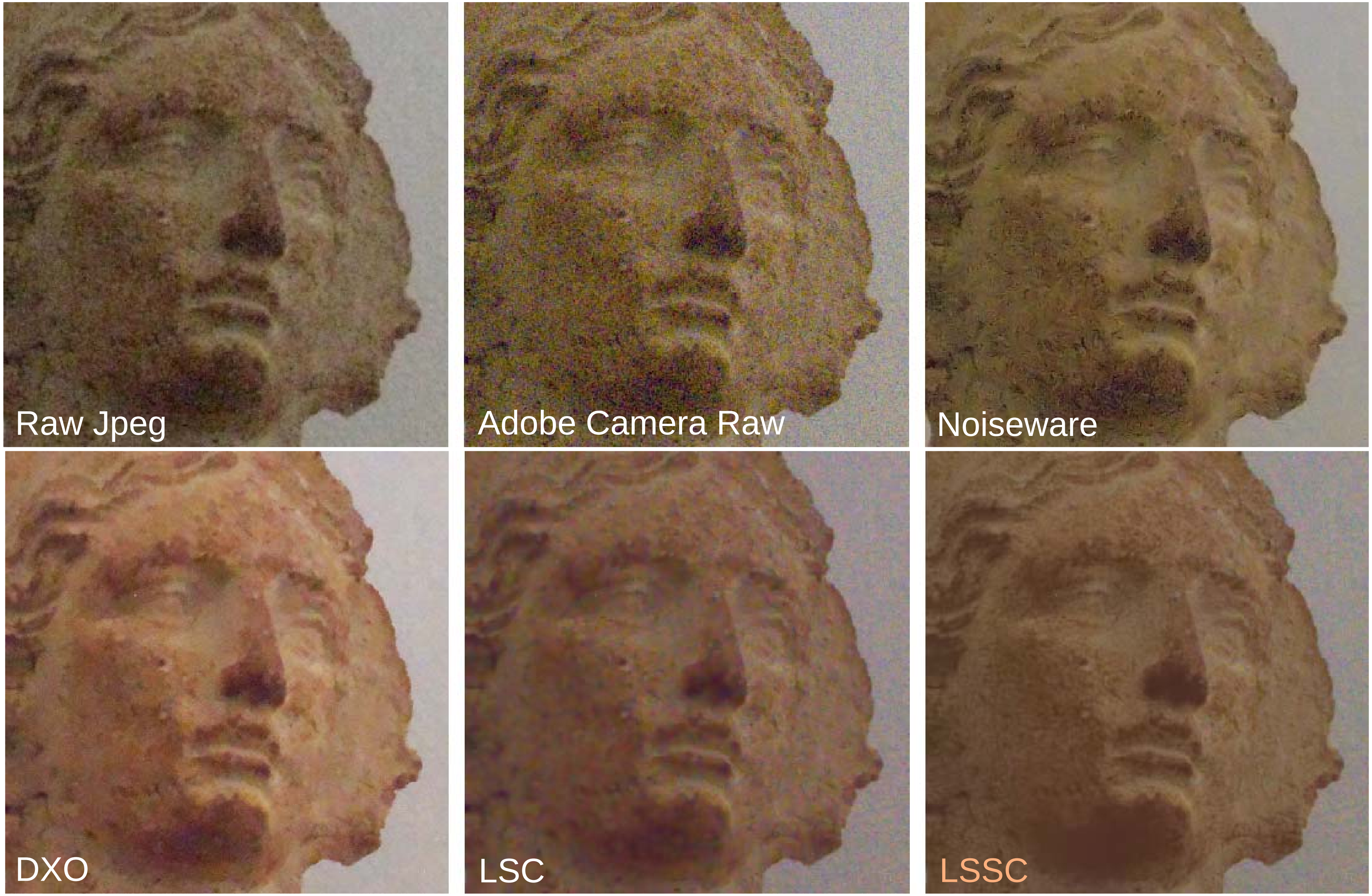
PSNR comparison between our method (LSSC) and Gunturk et al.'02 [AP]; Zhang & Wu'05 [DL]; and Paliy et al.'07 [LPA] on the Kodak PhotoCD data.

# Real noise (Canon Powershot G9, 1600 ISO)

# Real noise (Canon Powershot G9, 1600 ISO)



# Real noise (Canon Powershot G9, 1600 ISO)



# Learning discriminative dictionaries with $L_0$ constraints

(Mairal, Bach, Ponce, Sapiro, Zisserman, CVPR'08)

$$\alpha^*(x,D) = \underset{\alpha}{\text{Argmin}} \; | x - D\alpha |_2^2 \;\; \text{s.t.} \; |\alpha|_0 \leq L$$

$$R^*(x,D) = | x - D\alpha^* |_2^2$$

Orthogonal matching pursuit
(Mallat & Zhang'93, Tropp'04)

Reconstruction (MOD: Engan, Aase, Husoy'99; K-SVD: Aharon, Elad, Bruckstein'06):

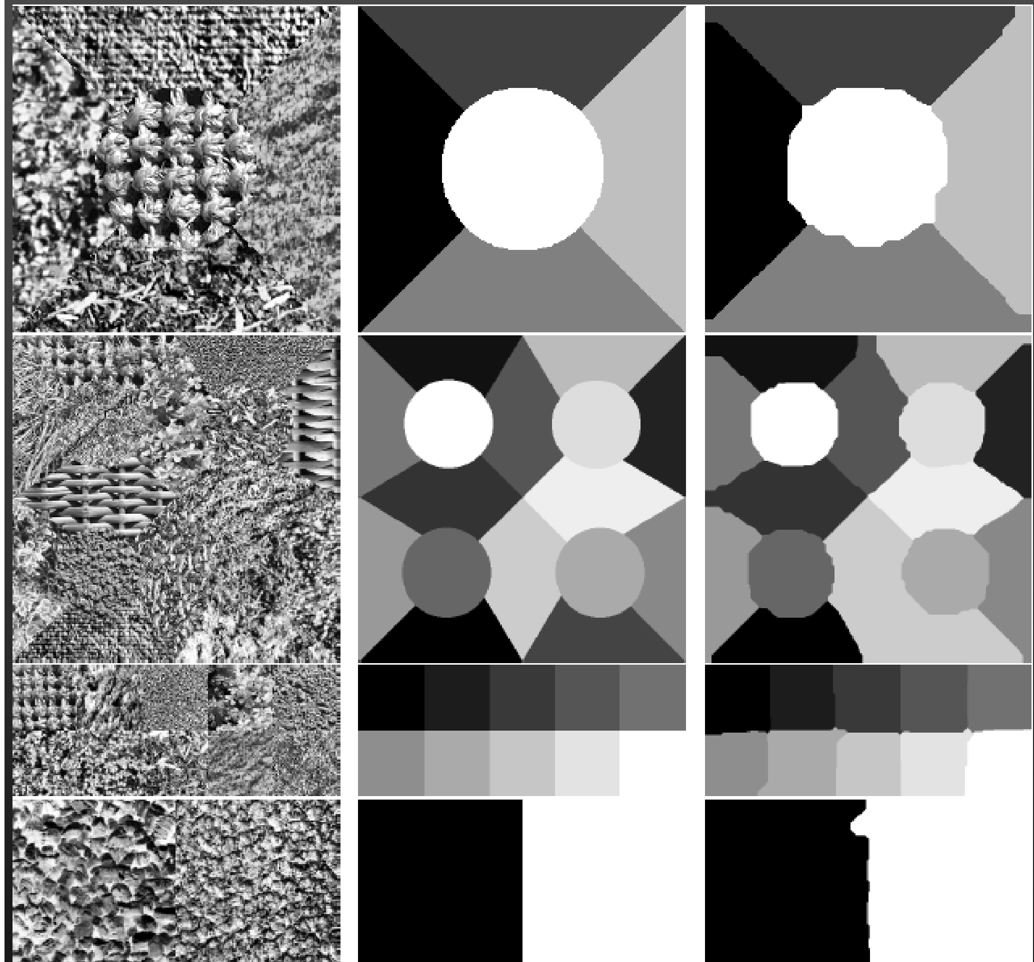
$$\min_{D} \sum_l R^*(x_l,D)$$

Discrimination:

$$\min_{D_1,\ldots,D_n} \sum_{i,l} C_i^{\lambda} [R^*(x_l,D_1),\ldots,R^*(x_l,D_n)] + \lambda\gamma \, R^*(x_l,D_i)$$

(Both MOD and K-SVD versions with truncated Newton iterations.)

# Texture classification results

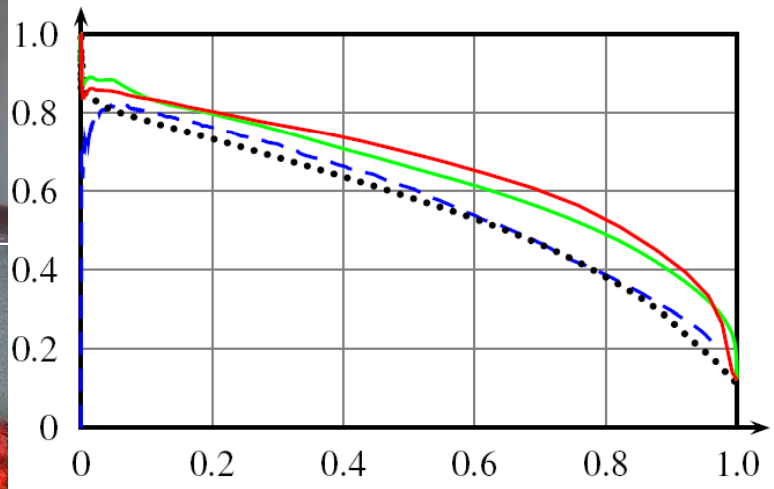


| ♯ | [28] | [17] | [34] | [16] | R1 | R2 | D1 | D2 |
|---|---|---|---|---|---|---|---|---|
| 1 | 7.2 | 6.7 | 5.5 | 3.37 | 2.22 | 1.69 | 1.89 | **1.61** |
| 2 | 18.9 | 14.3 | **7.3** | 16.05 | 24.66 | 36.5 | 16.38 | 16.42 |
| 3 | 20.6 | 10.2 | 13.2 | 13.03 | 10.20 | 5.49 | 9.11 | **4.15** |
| 4 | 16.8 | 9.1 | 5.6 | 6.62 | 6.66 | 4.60 | 3.79 | **3.67** |
| 5 | 17.2 | 8.0 | 10.5 | 8.15 | 5.26 | **4.32** | 5.10 | 4.58 |
| 6 | 34.7 | 15.3 | 17.1 | 18.66 | 16.88 | 15.50 | 12.91 | **9.04** |
| 7 | 41.7 | 20.7 | 17.2 | 21.67 | 19.32 | 21.89 | 11.44 | **8.80** |
| 8 | 32.3 | 18.1 | 18.9 | 21.96 | 13.27 | 11.80 | 14.77 | **2.24** |
| 9 | 27.8 | 21.4 | 21.4 | 9.61 | 18.85 | 21.88 | 10.12 | **2.04** |
| 10 | 0.7 | 0.4 | NA | 0.36 | 0.35 | **0.17** | 0.20 | **0.17** |
| 11 | **0.2** | 0.8 | NA | 1.33 | 0.58 | 0.73 | 0.41 | 0.60 |
| 12 | 2.5 | 5.3 | NA | 1.14 | 1.36 | **0.37** | 1.97 | 0.78 |
| **Av.** | 18.4 | 10.9 | NA | 10.16 | 9.97 | 10.41 | 7.34 | **4.50** |

# Pixel-level classification results

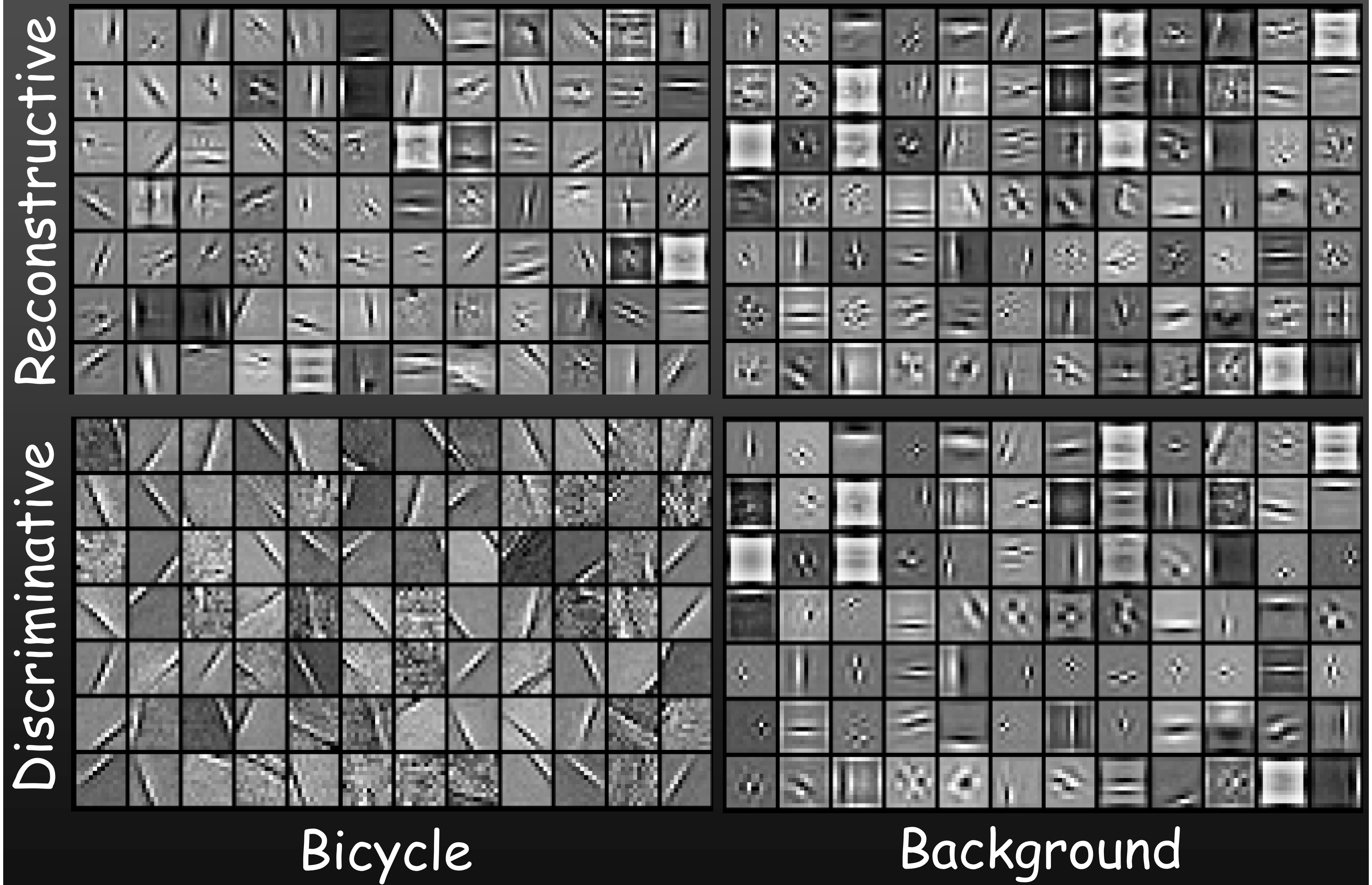
Qualitative results, Graz 02 data

Quantitative results




Comparaison with Pantofaru et al. (2006) and Tuytelaars & Schmid (2007).

# Reconstructive vs discriminative dictionaries



# Learning discriminative dictionaries with $L_1$ constraints

(Mairal, Leordeanu, Bach, Hebert, Ponce, ECCV'08)

$$\alpha^*(x,D) = \underset{\alpha}{\text{Argmin}}\ | x - D\alpha |_2^2 \text{ s.t. } |\alpha|_1 \leq L$$

$$R^*(x,D) = | x - D\alpha^* |_2^2$$

Lasso: Convex optimization
(LARS: Efron et al.'04)

Reconstruction (Lee, Battle, Rajat, Ng'07):

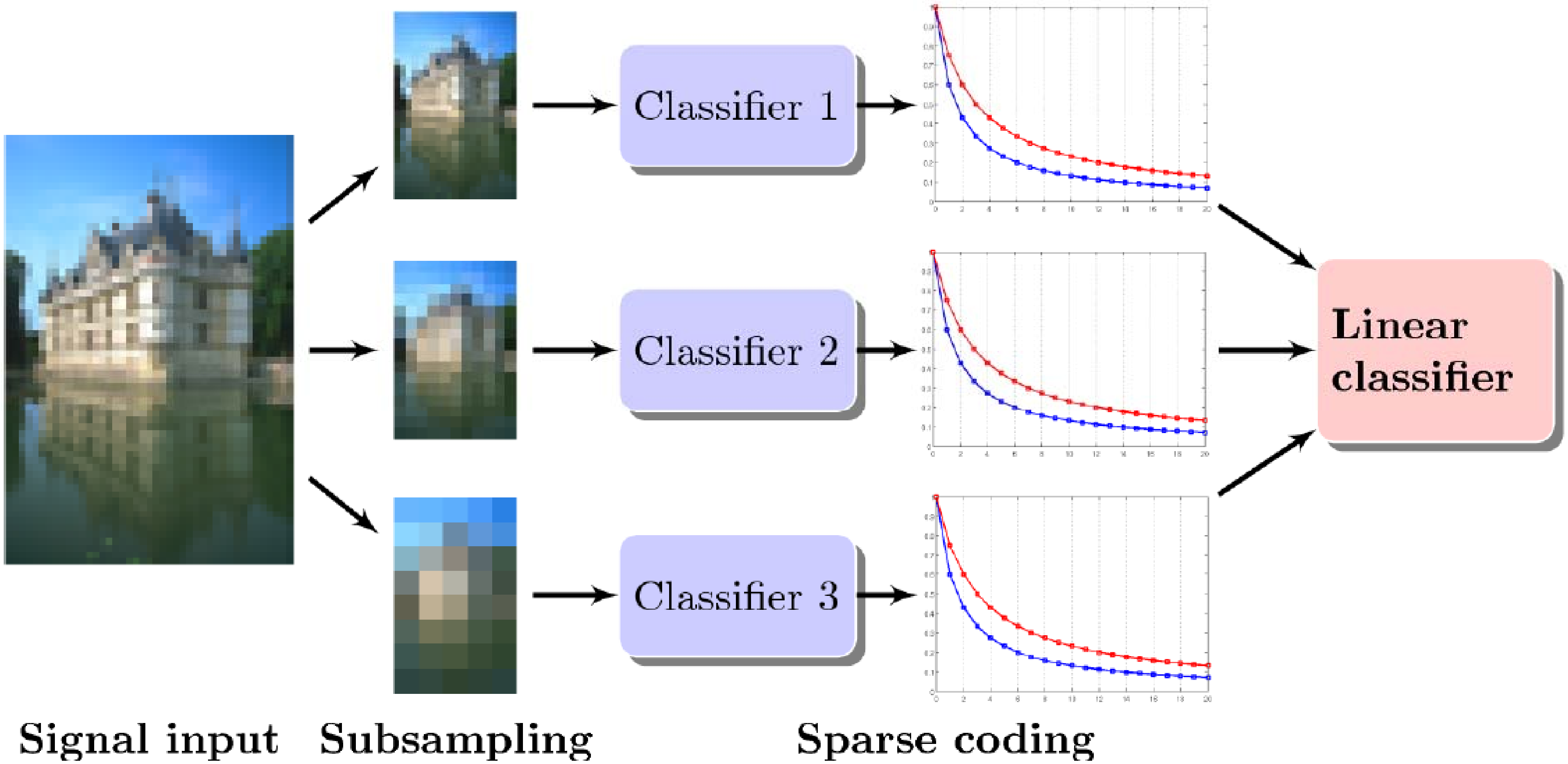
$$\min_D \sum_l R^*(x_l,D)$$

Discrimination:

$$\min_{D_1,\ldots,D_n} \sum_{i,l} C_i^{\lambda} [R^*(x_l,D_1),\ldots,R^*(x_l,D_n)] + \lambda\gamma R^*(x_l,D_i)$$

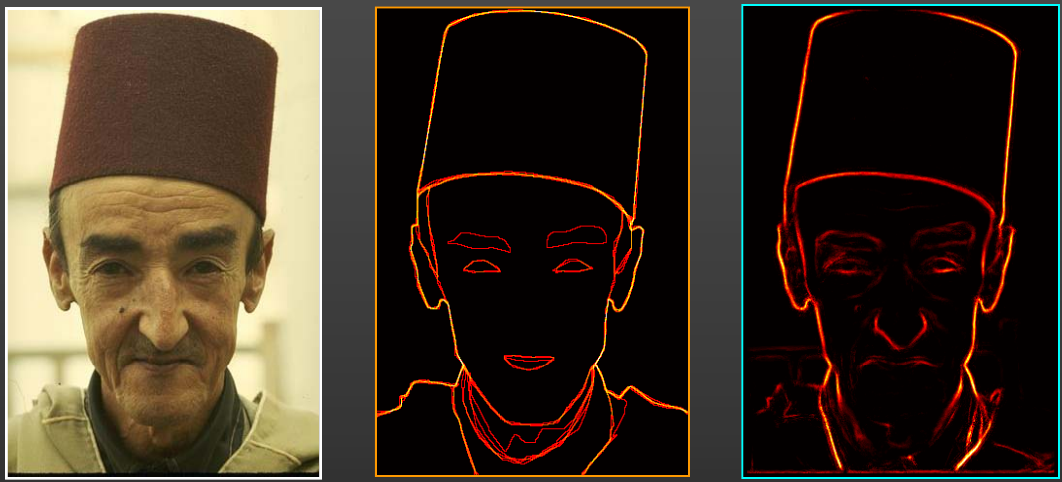
(Partial dictionary update with Newtown iterations on the dual problem; partial fast sparse coding with projected gradient descent.)

# Patch classification with learned dictionaries

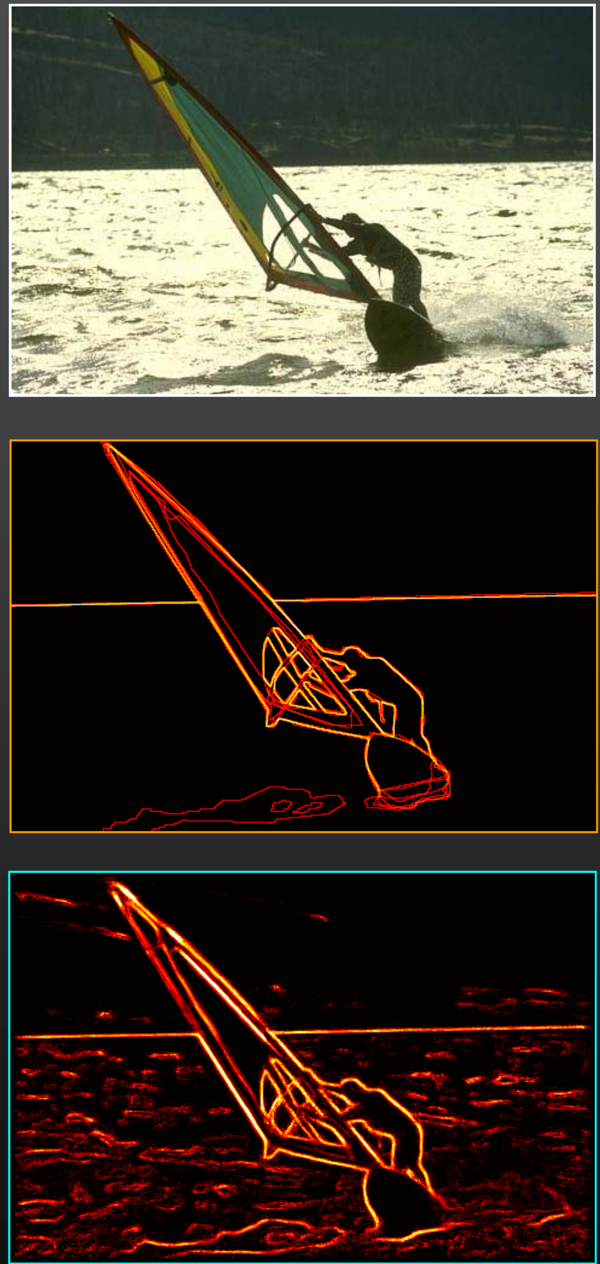


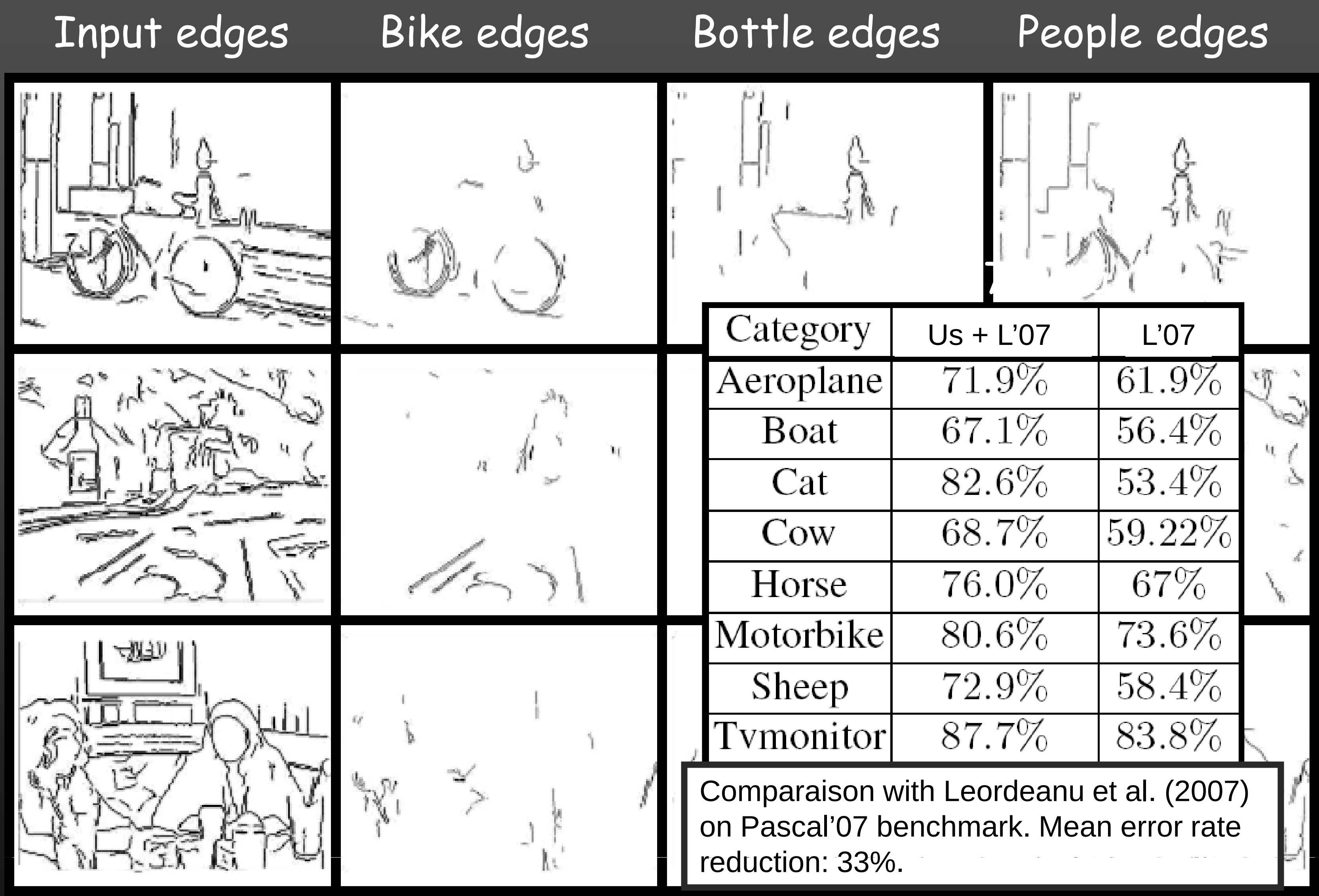
# Edge detection results

Quantitative results on the Berkeley segmentation dataset and benchmark (Martin et al., ICCV'01)



| Rank | Score | Algorithm |
|---|---|---|
| 0 | 0.79 | Human labeling |
| 1 | 0.70 | (Maire et al., 2008) |
| 2 | 0.67 | (Aerbelaez, 2006) |
| 3 | 0.66 | (Dollar et al., 2006) |
| 3 | 0.66 | Us – no post-processing |
| 4 | 0.65 | (Martin et al., 2001) |
| 5 | 0.57 | Color gradient |
| 6 | 0.43 | Random |






| Category | Us + L'07 | L'07 |
|---|---|---|
| Aeroplane | 71.9% | 61.9% |
| Boat | 67.1% | 56.4% |
| Cat | 82.6% | 53.4% |
| Cow | 68.7% | 59.22% |
| Horse | 76.0% | 67% |
| Motorbike | 80.6% | 73.6% |
| Sheep | 72.9% | 58.4% |
| Tvmonitor | 87.7% | 83.8% |

Comparaison with Leordeanu et al. (2007) on Pascal'07 benchmark. Mean error rate reduction: 33%.

# Task-driven dictionary learning

(Mairal, Bach, Ponce, 2010)

$$\min_{W,D} f(W,D) = \mathbb{E}_{x,y}\,[L(y, W, \alpha^*(x, D))] + \nu|W|_F^2$$

$$\text{with } \alpha^*(x,D) = \underset{\alpha}{\text{Argmin}}\ |x - D\alpha|_2^2 + \lambda|\alpha|_1 + \mu|\alpha|_2^2$$

(Mairal et al.'08; Bradley & Bagnell'09; Boureau et al.'10; Yang et al.'10)

**Applications:** Regression, classification, compressed sensing.

**Extensions:** Learning linear transforms of the input data, semi-supervised learning.

**Proposition:** Under mild assumptions, the function f is differentiable, and its gradient can be written in closed form as an expectation.

**Algorithm:** Stochastic gradient descent.

Authentic

Fake

(Mairal, Bach, Ponce, 2010)

Data courtesy of James Hughes & Daniel Rockmore

Authentic

Fake

Fake

Data courtesy of James Hughes & Daniel Rockmore

Authentic

Fake

Authentic

Data courtesy of James Hughes & Daniel Rockmore

# A common architecture for image classification

| | | | |
|---|---|---|---|
| Filtering | SIFT at keypoints | dense gradients | dense SIFT |
| Coding | vector quantization | vector quantization | sparse coding |
| Pooling | whole image, mean | coarse grid, mean | spatial pyramid, max |

Idea: Replace k-means by sparse coding (Yang et al., CVPR'09; Boureau et al., CVPR'10, ICML'10; Yang et al., CVPR'10).

# Learning dictionaries for image classification
## (Boureau, LeCun, Bach, Ponce, CVPR'10)

| Method | Caltech-101, 30 training examples | | 15 Scenes, 100 training examples | |
|---|---|---|---|---|
| | Average Pool | Max Pool | Average Pool | Max Pool |
| | Results with basic features, SIFT extracted each 8 pixels | | | |
| Hard quantization, linear kernel | $51.4 \pm 0.9$ [256] | $64.3 \pm 0.9$ [256] | $73.9 \pm 0.9$ [1024] | $80.1 \pm 0.6$ [1024] |
| Hard quantization, intersection kernel | $64.2 \pm 1.0$ [256] **(1)** | $64.3 \pm 0.9$ [256] | $80.8 \pm 0.4$ [256] **(1)** | $80.1 \pm 0.6$ [1024] |
| Soft quantization, linear kernel | $57.9 \pm 1.5$ [1024] | $69.0 \pm 0.8$ [256] | $75.6 \pm 0.5$ [1024] | $81.4 \pm 0.6$ [1024] |
| Soft quantization, intersection kernel | $66.1 \pm 1.2$ [512] **(2)** | $70.6 \pm 1.0$ [1024] | $81.2 \pm 0.4$ [1024] **(2)** | $83.0 \pm 0.7$ [1024] |
| Sparse codes, linear kernel | $61.3 \pm 1.3$ [1024] | $71.5 \pm 1.1$ [1024] **(3)** | $76.9 \pm 0.6$ [1024] | $83.1 \pm 0.6$ [1024] **(3)** |
| Sparse codes, intersection kernel | $70.3 \pm 1.3$ [1024] | $71.8 \pm 1.0$ [1024] **(4)** | $83.2 \pm 0.4$ [1024] | $84.1 \pm 0.5$ [1024] **(4)** |

| Single - feature | Method | Caltech 15 tr. | Caltech 30 tr. | Scenes |
|---|---|---|---|---|
| Boiman et al. [3] | Nearest neighbor + spatial correspondence | $65.0 \pm 1.1$ | 70.4 | - |
| Jain et al. [9] | Fast image search for learned metrics | 61.0 | 69.6 | - |
| Lazebnik et al. [12] | (1) SP + hard quantization + kernel SVM | 56.4 | $64.4 \pm 0.8$ | $81.4 \pm 0.5$ |
| van Gemert et al. [27] | (2) SP + soft quantization + kernel SVM | – | $64.1 \pm 1.2$ | $76.7 \pm 0.4$ |
| Yang et al. [31] | (3) SP + sparse codes + max pooling + linear SVM | $67.0 \pm 0.5$ | $73.2 \pm 0.5$ | $80.3 \pm 0.9$ |
| Yang et al. [31] | (4) SP + sparse codes + max pooling + kernel SVM | $60.4 \pm 1.0$ | – | $77.7 \pm 0.7$ |
| Zhang et al. [32] | $k$NN-SVM | $59.1 \pm 0.6$ | $66.2 \pm 0.5$ | - |
| Zhou et al. [33] | SP + Gaussian mixture | – | – | $84.1 \pm 0.5$ |

Scenes, supervised dictionary learning

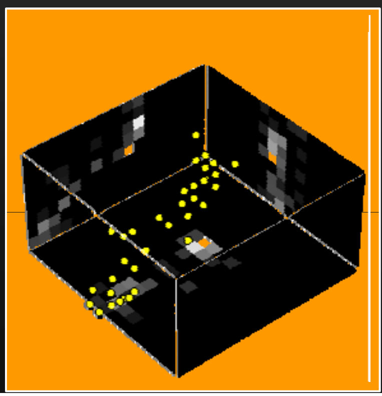
| | Unsup | Discr [1024] | Unsup | Discr [2048] |
|---|---|---|---|---|
| Linear | $83.6 \pm 0.4$ | $84.9 \pm 0.3$ | $84.2 \pm 0.3$ | $85.6 \pm 0.2$ |
| Intersect | $84.3 \pm 0.5$ | $84.7 \pm 0.4$ | $84.6 \pm 0.4$ | $85.1 \pm 0.5$ |

# Non-blind deblurring (Couzinie-Devy, Mairal, Bach, Ponce, 2010)

| | *Cameraman* | | | | *Lena* | | | |
|---|---|---|---|---|---|---|---|---|
| PSNR input image | 20.76 | 22.35 | 22.29 | 24.7 | 25.84 | 27.57 | 27.35 | 29.00 |
| Richardson-Lucy [27] | 4.47 | 5.53 | 3.58 | 0.50 | 4.80 | 5.29 | 2.71 | -0.07 |
| Guerrero-Colon et al.[21] | 7.33 | 7.45 | 5.55 | 2.73 | NA | NA | NA | NA |
| SA-DCT [16] | 8.55 | 8.11 | 6.33 | 3.37 | 7.79 | 7.55 | 6.10 | 4.49 |
| Dabov et al. [20] | 8.34 | 8.19 | 6.40 | 3.34 | 7.97 | 7.95 | 6.53 | 4.81 |
| Ours, $\gamma = 0$ | 4.49 | 8.11 | 6.61 | 3.11 | 4.41 | 8.10 | 6.78 | 5.16 |
| Ours, $\gamma \neq 0$ + denoising | 6.39 | 8.14 | 6.64 | 3.14 | 6.30 | 8.24 | 6.9 | 5.31 |

# Non-uniform blind deblurring

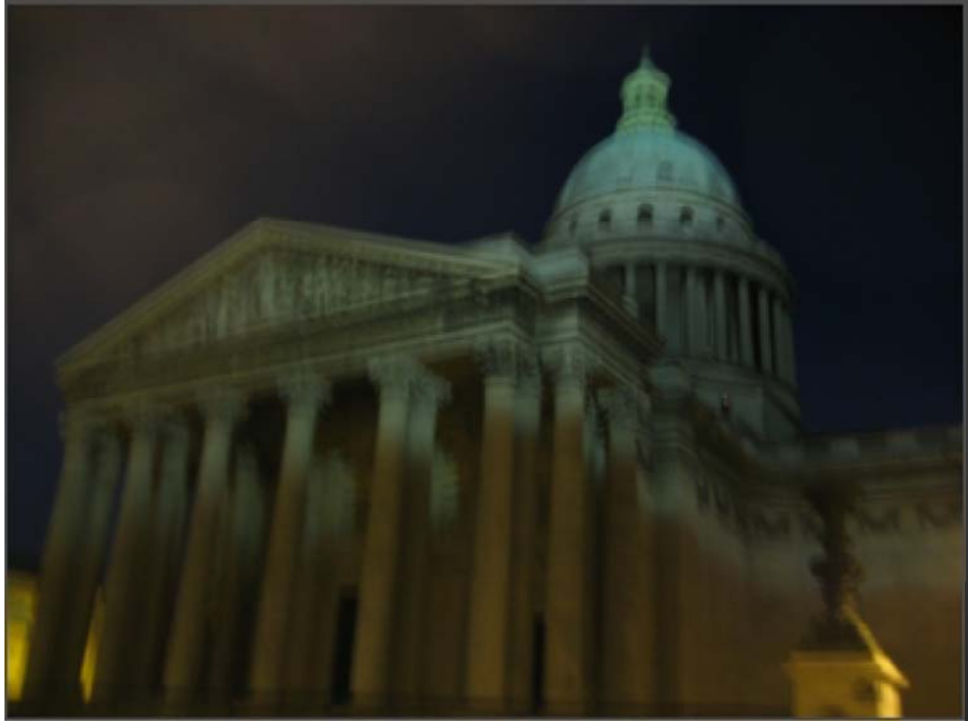
(Whyte et al., CVPR'10)

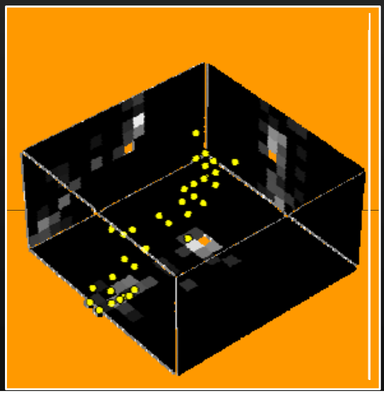



# !! Short detour !!

# Non-uniform blind deblurring

(Whyte et al., CVPR'10)

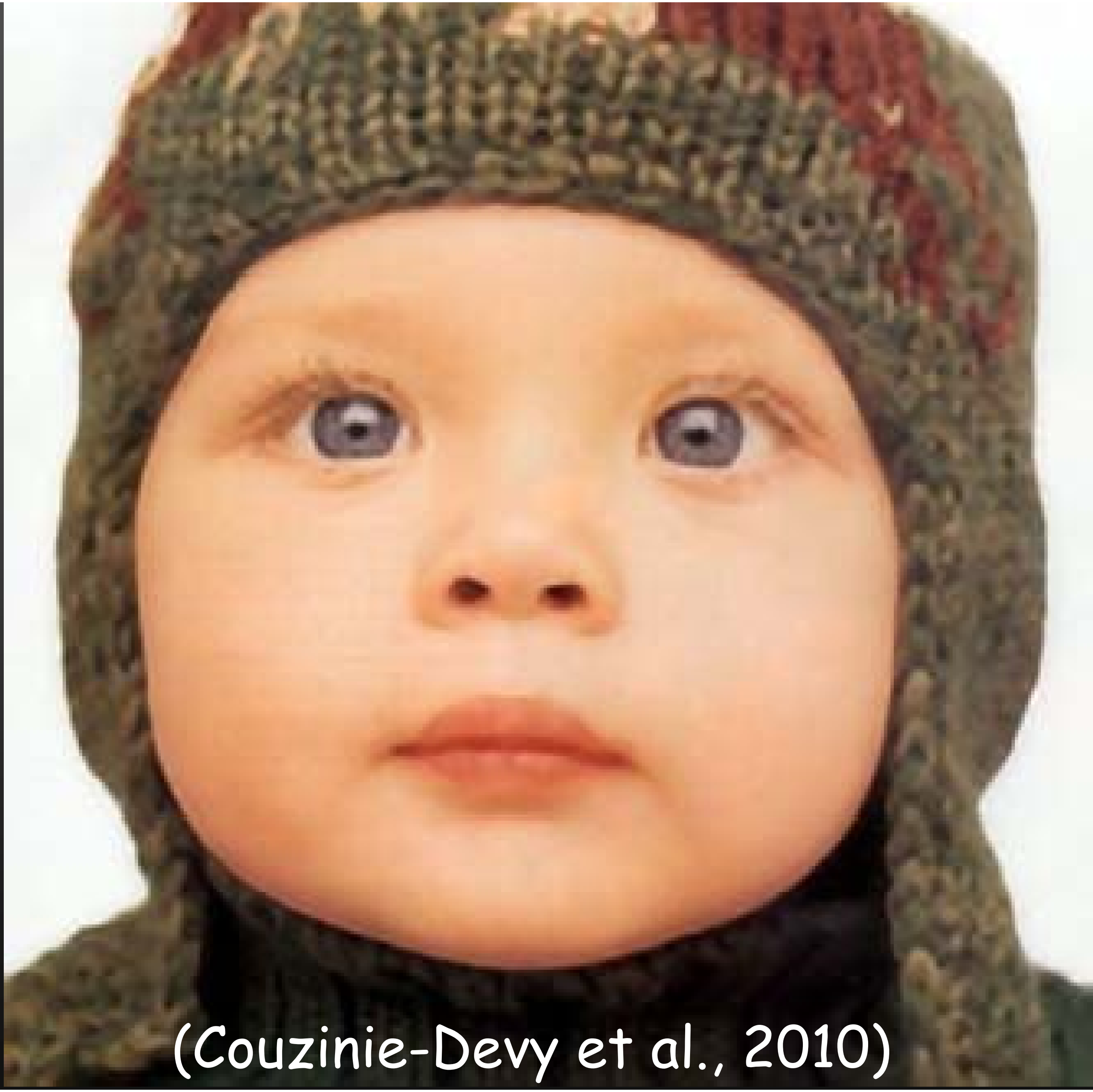




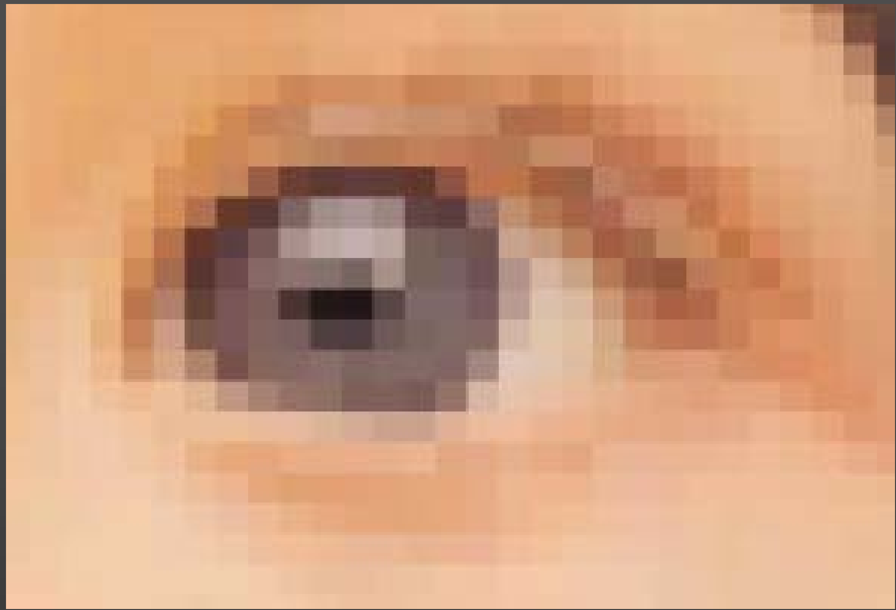

Digital Zoom

(Fattal, 2007)

(Glasner et al., 2009)



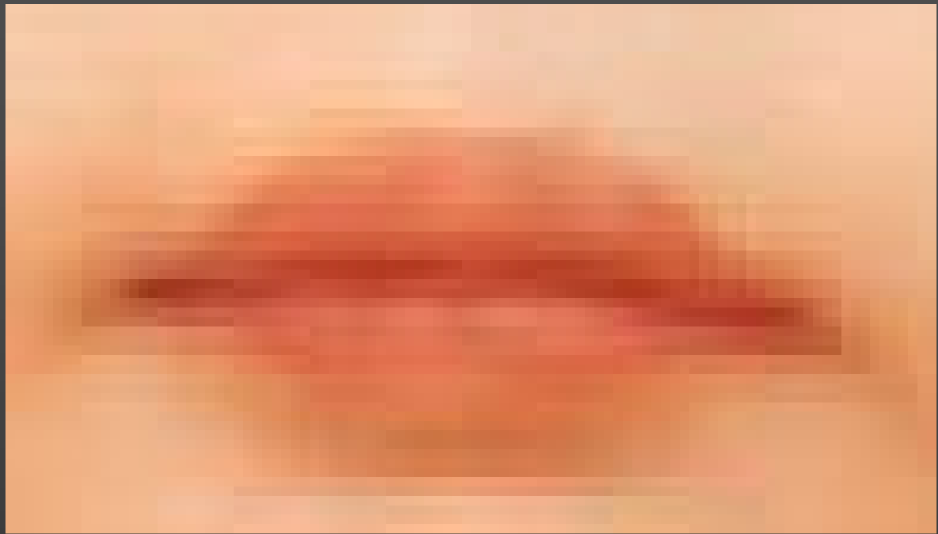
(Couzinie-Devy et al., 2010)

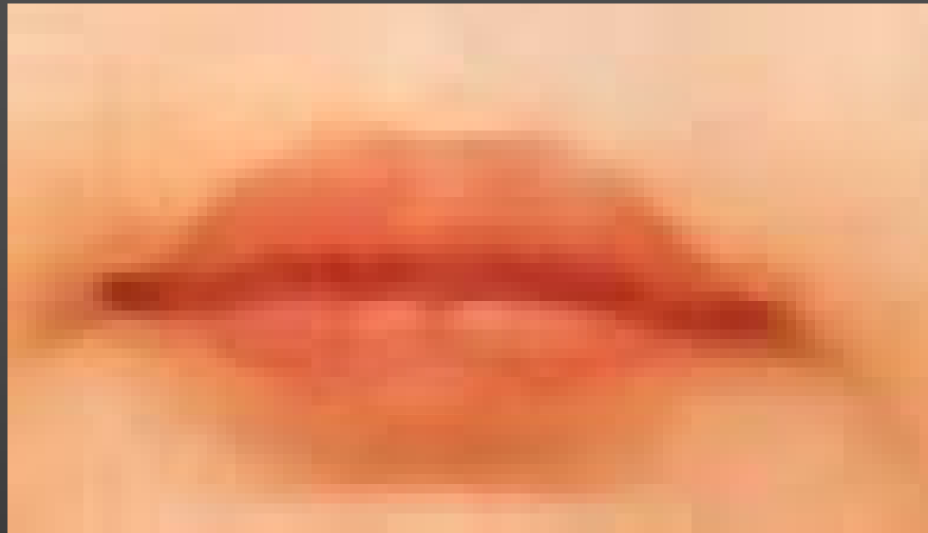


(Fattal, 2007)

(Glasner et al., 2009)

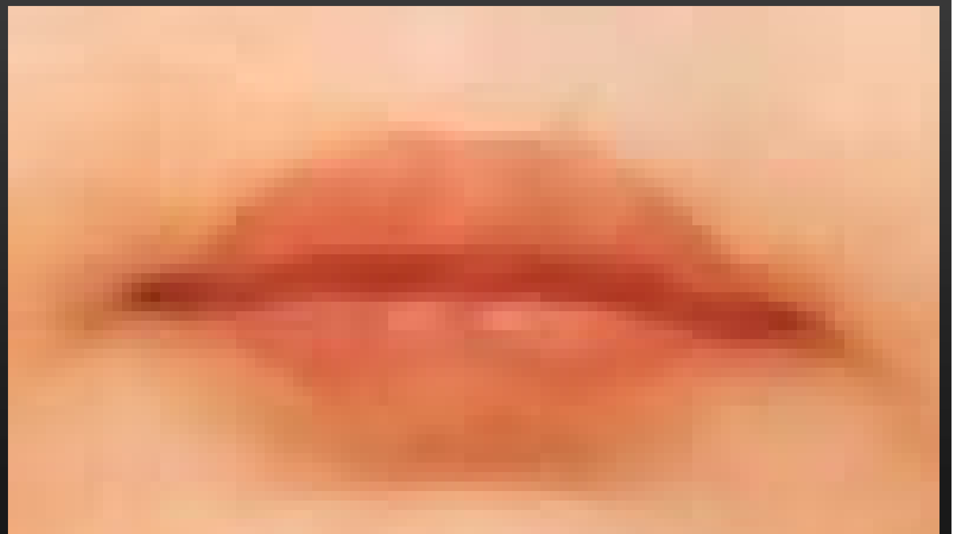
(Couzinie-Devy et al., 2010)



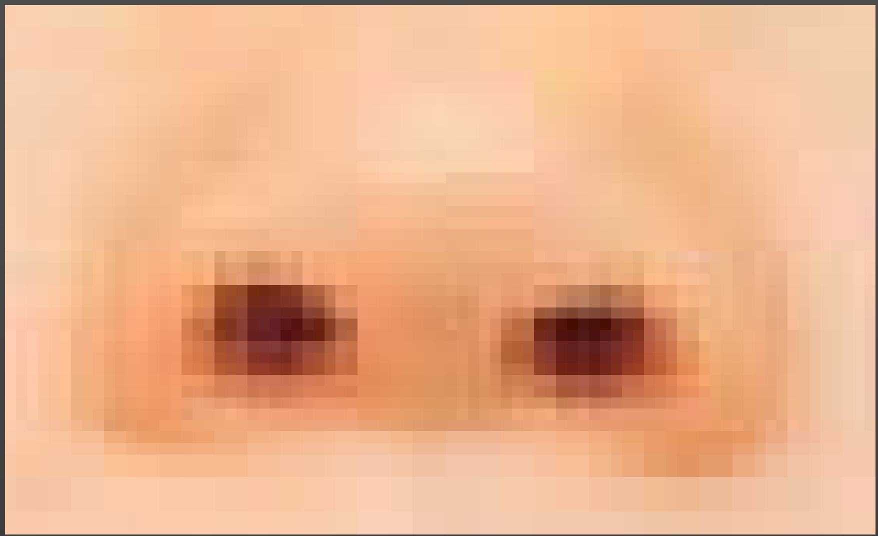


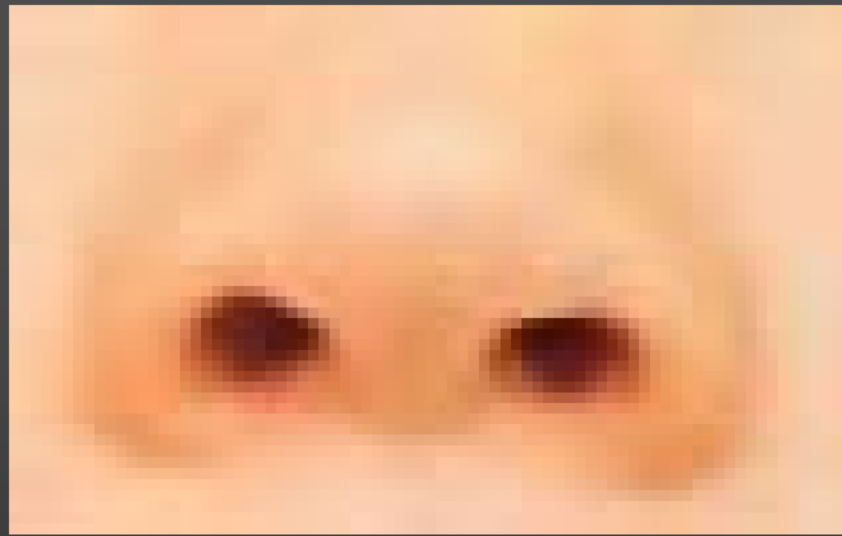
(Fattal, 2007)

(Glasner et al., 2009)

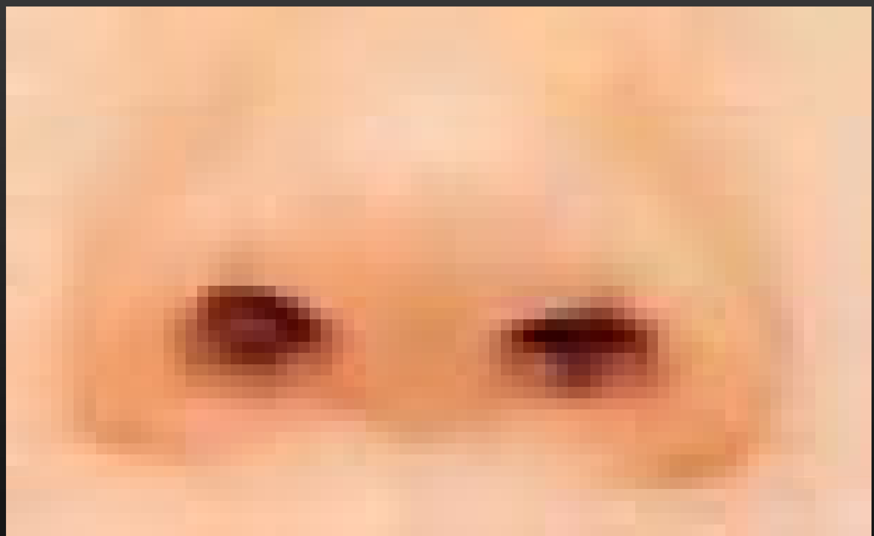


(Couzinie-Devy et al., 2010)

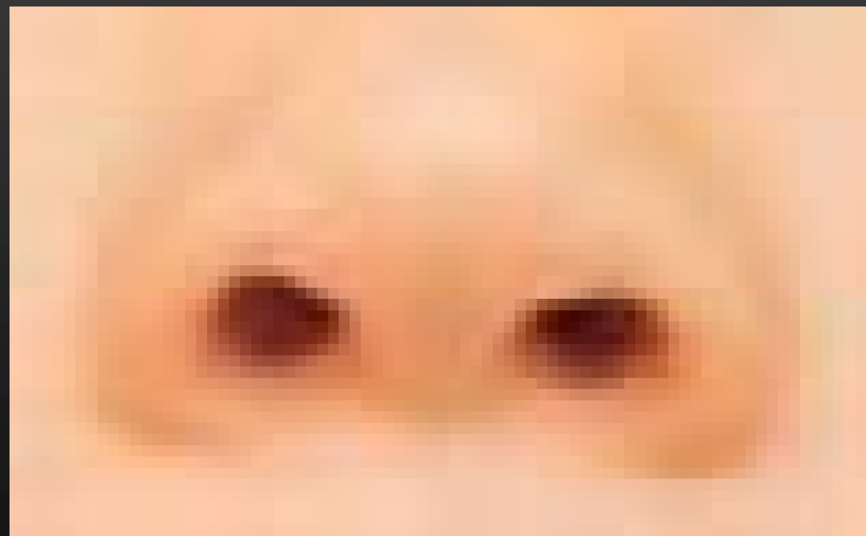




(Fattal, 2007)

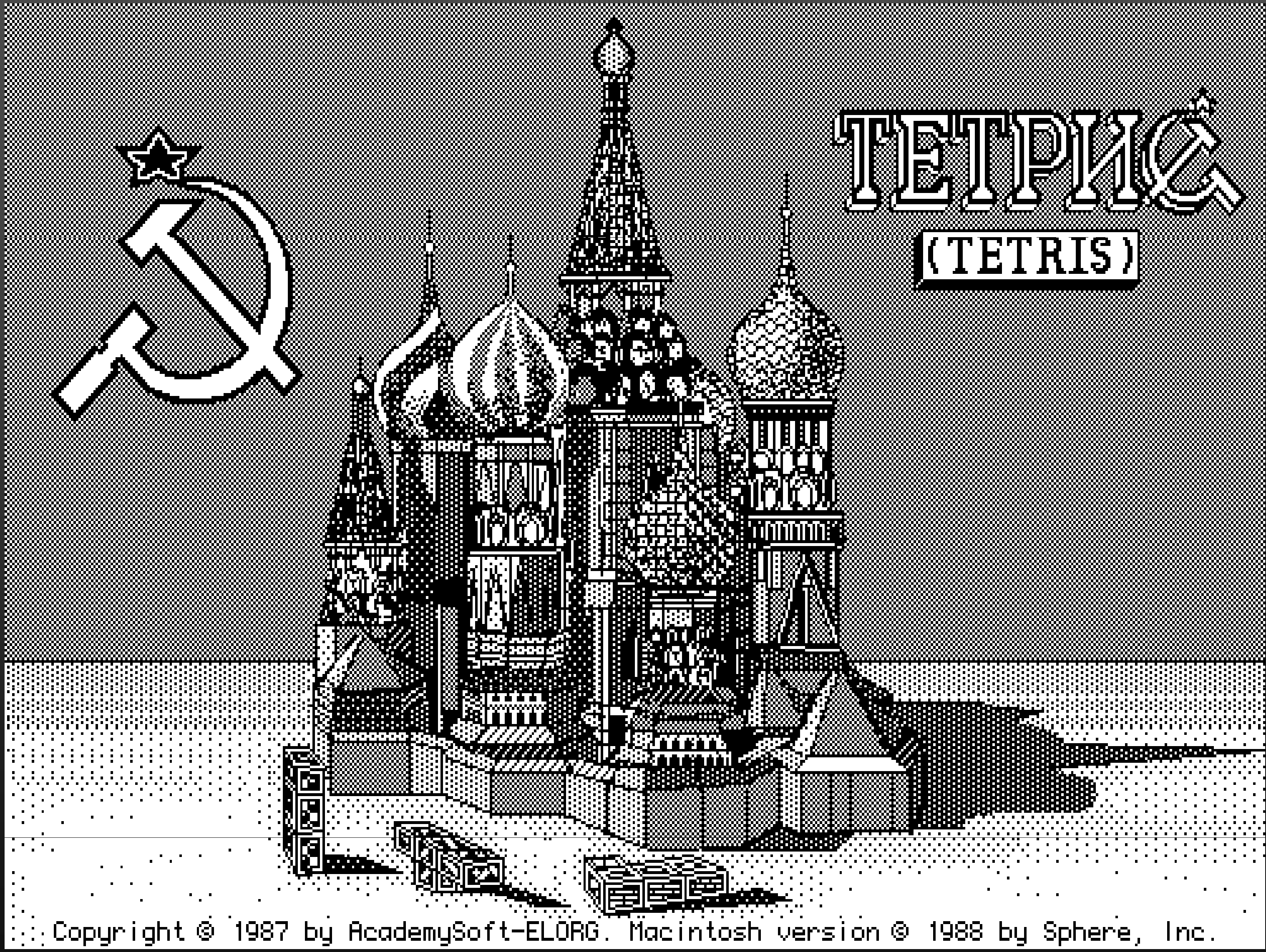


(Glasner et al., 2009)



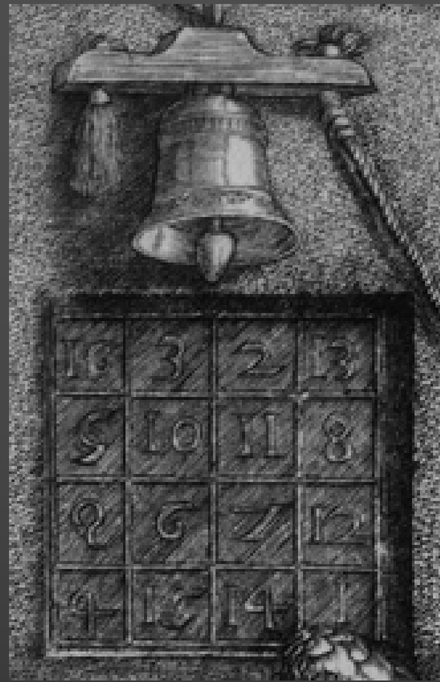
(Couzinie-Devy et al., 2010)

# Inverse halftoning
(Mairal, Bach, Ponce, 2010)



# Inverse halftoning
(Mairal, Bach, Ponce, 2010)


MELENCOLIA§I

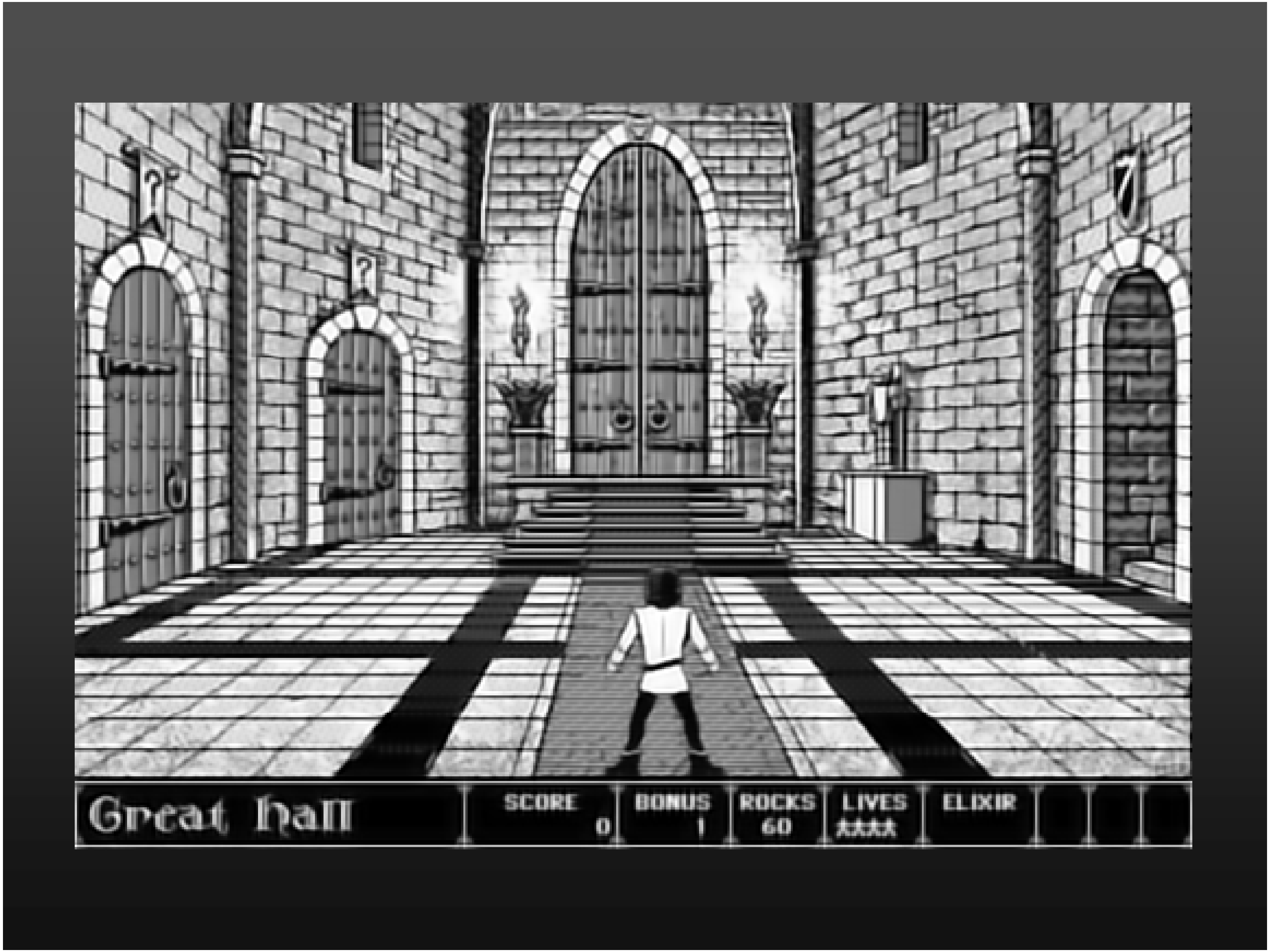

16 3 2 13
5 10 11 8
9 6 7 12
4 15 14 1


MELENCOLIA I

| | Validation set | | | | Test set | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Image | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| FIHT2 | 30.8 | 25.3 | 25.8 | 31.4 | 24.5 | 28.6 | 29.5 | 28.2 | 29.3 | 26.0 | 25.2 | 24.7 |
| WInHD | 31.2 | 26.9 | 26.8 | 31.9 | 25.7 | 29.2 | 29.4 | 28.7 | 29.4 | 28.1 | 25.6 | 26.4 |
| LPA-ICI | 31.4 | 27.7 | 26.5 | 32.5 | 25.6 | 29.7 | 30.0 | 29.2 | 30.1 | 28.3 | 26.0 | 27.2 |
| SA-DCT | 32.4 | 28.6 | 27.8 | 33.0 | 27.0 | 30.1 | 30.2 | 29.8 | 30.3 | 28.5 | 26.2 | 27.6 |
| Ours | 33.0 | 29.6 | 28.1 | 33.0 | 26.6 | 30.2 | 30.5 | 29.9 | 30.4 | 29.0 | 26.2 | 28.0 |

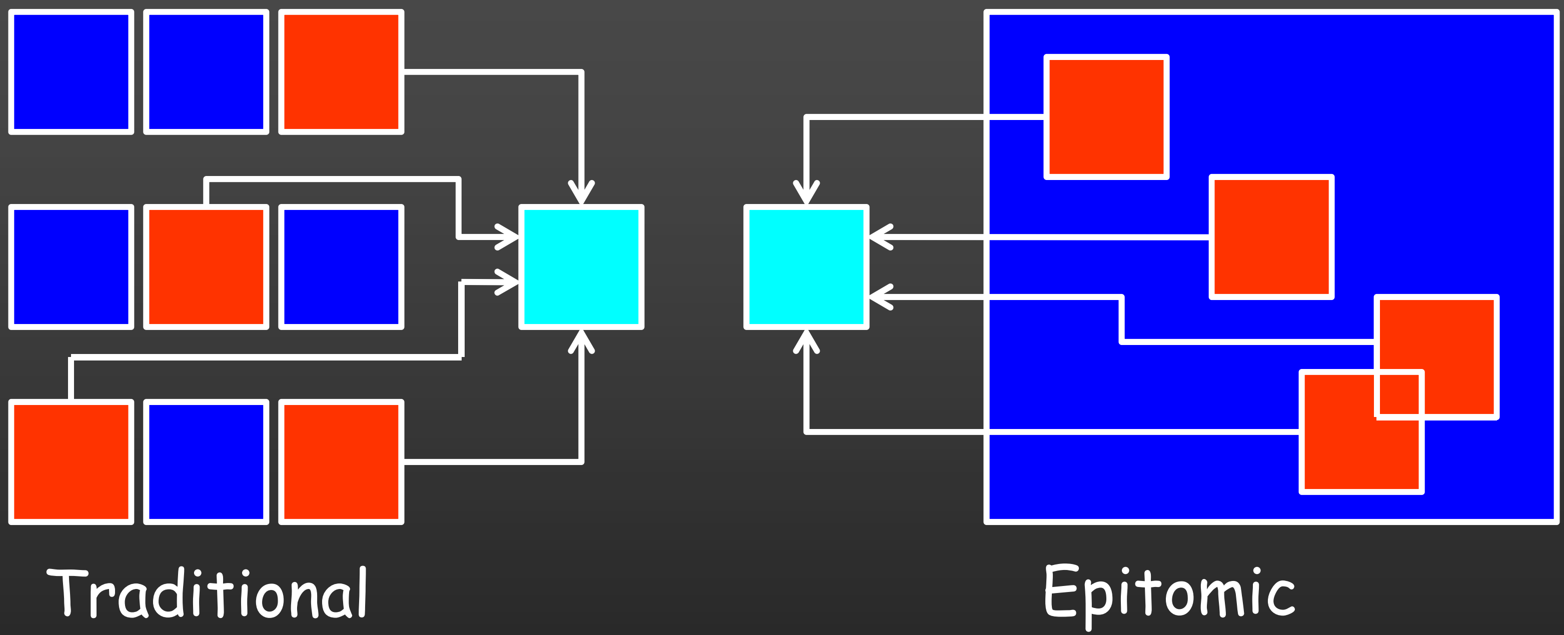
PSNR comparison between our method and Kite et al.'00 [FIHT2]; Neelamini et al.'09 [WInHD]; Foi et al.'04 [LPA-ICI]; and Dabov et al.'06 [SA-DCT].


Great hall
SCORE
0
BONUS
1
ROCKS
60
LIVES
ELIXIR
MSF


Great hall
SCORE
0
BONUS
1
ROCKS
60
LIVES
ELIXIR

# Epitomic dictionaries

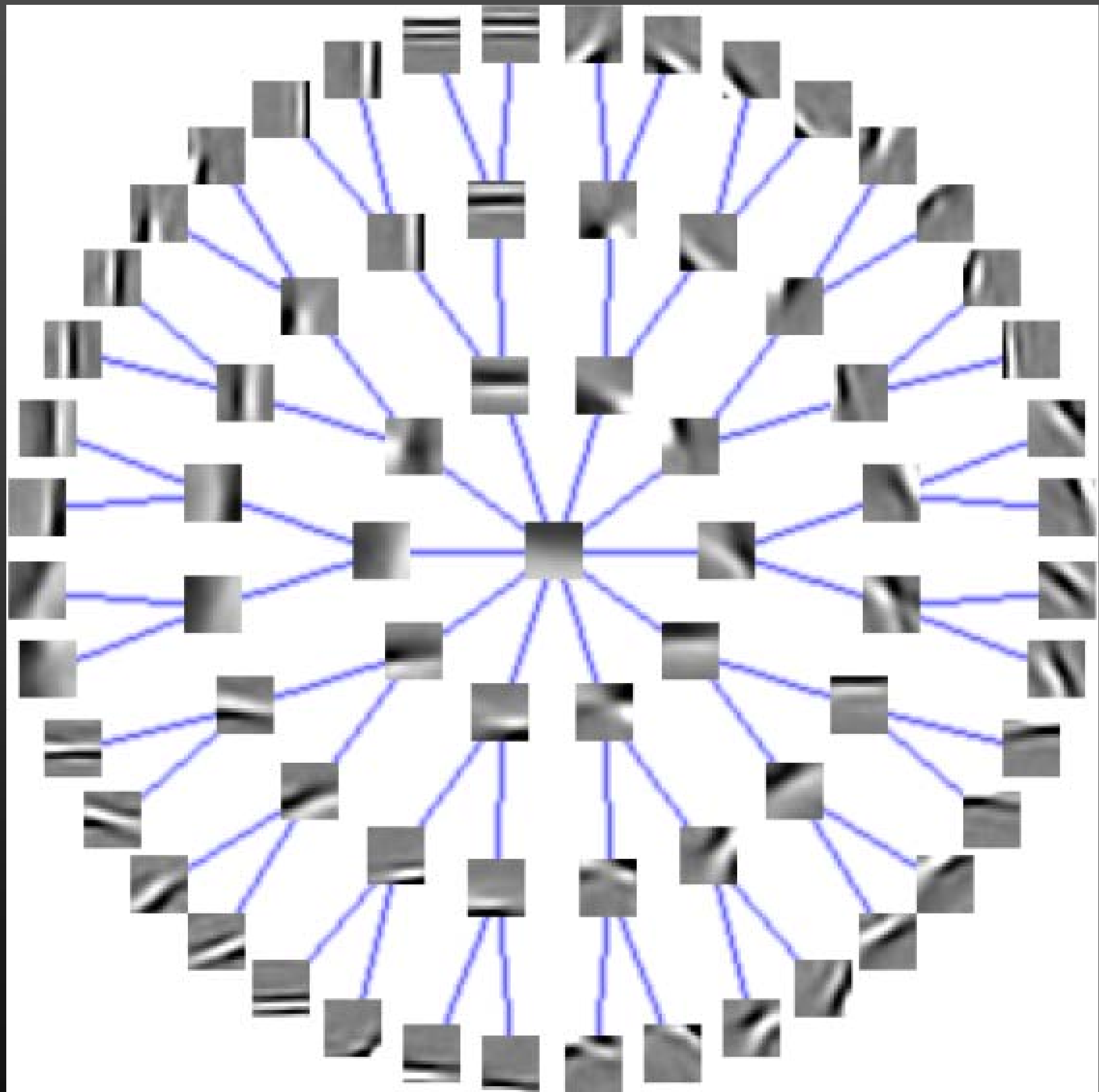
(Benoit, Mairal, Bach, Ponce, 2010)



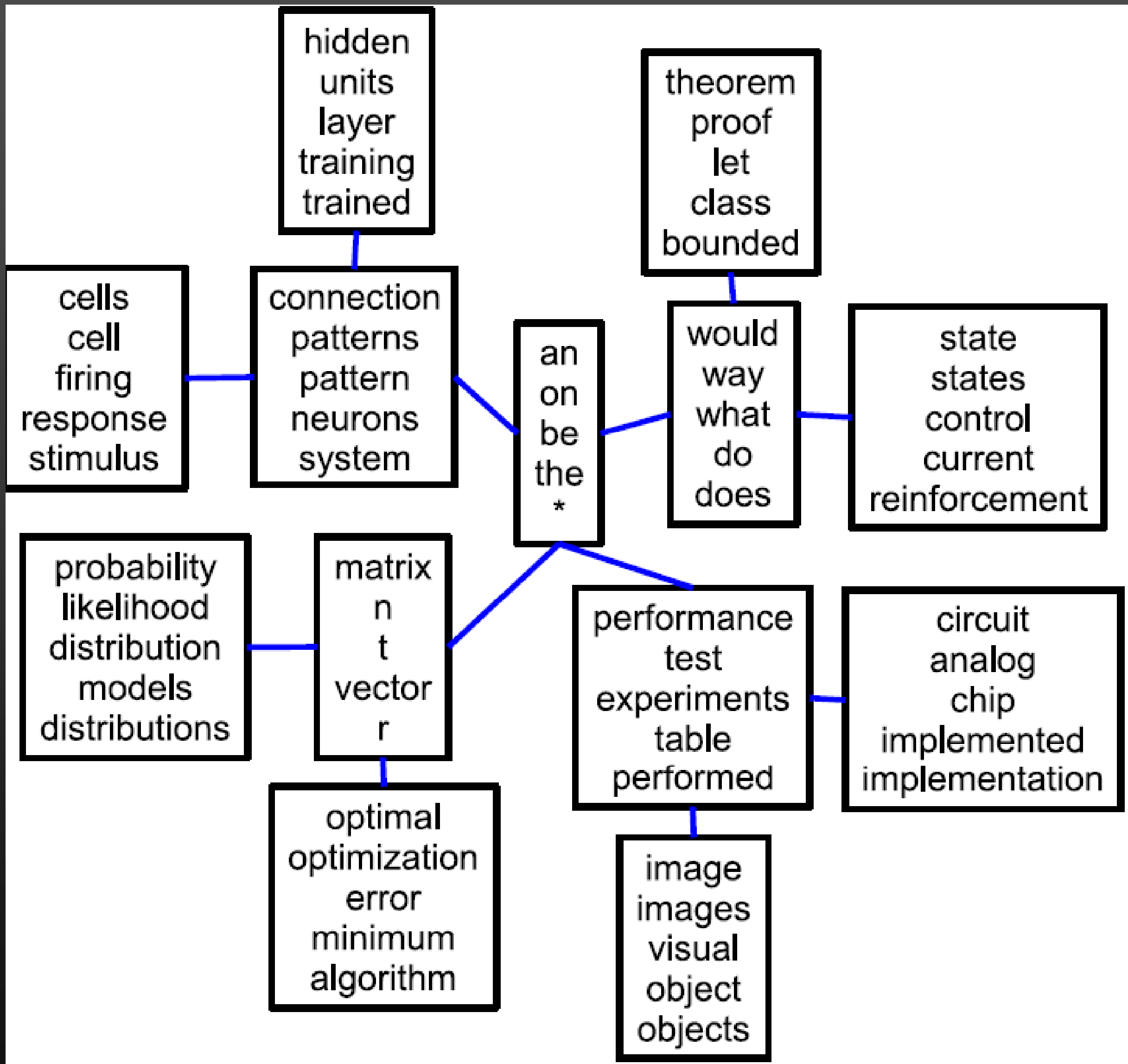

Epitomes: (Jojic, Frey, Kannan, 2003)

Related ideas: (Aharon & Elad, 2007; Hyvarinen & Hoyer, 2001; Kavukcuoglu et al., 2009; Zeiler et al., 2010)

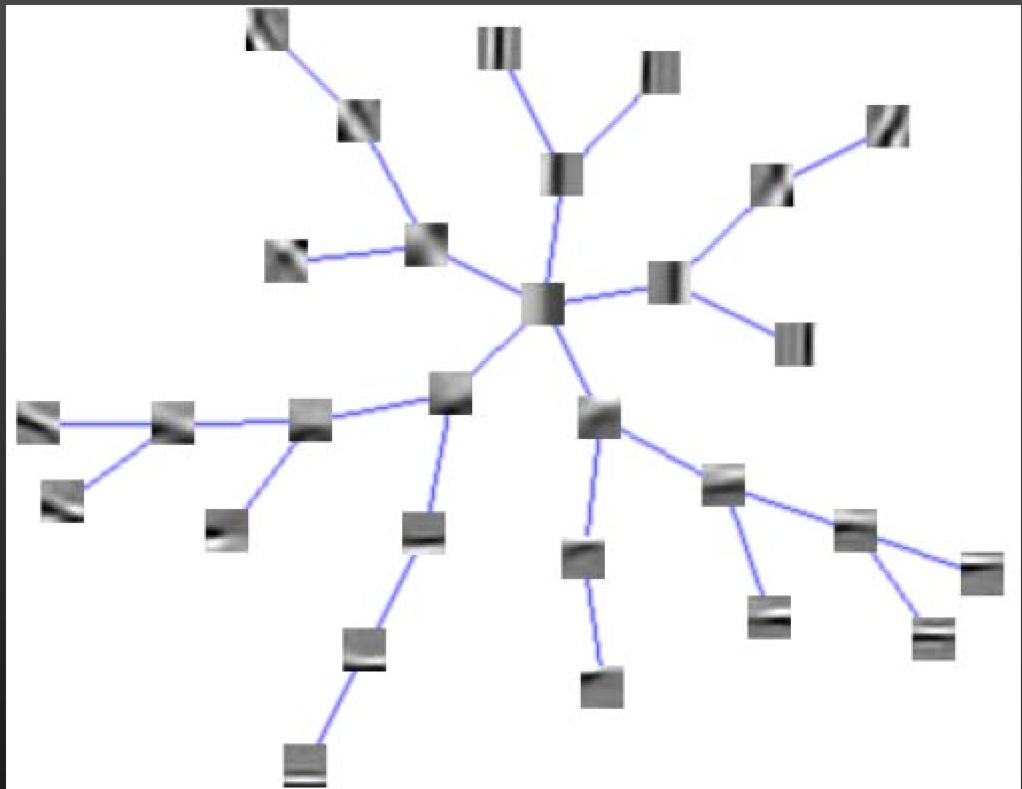
# Proximal methods for sparse hierarchical dictionary learning
(Jenatton, Mairal, Obozinski, Bach, ICML'10)

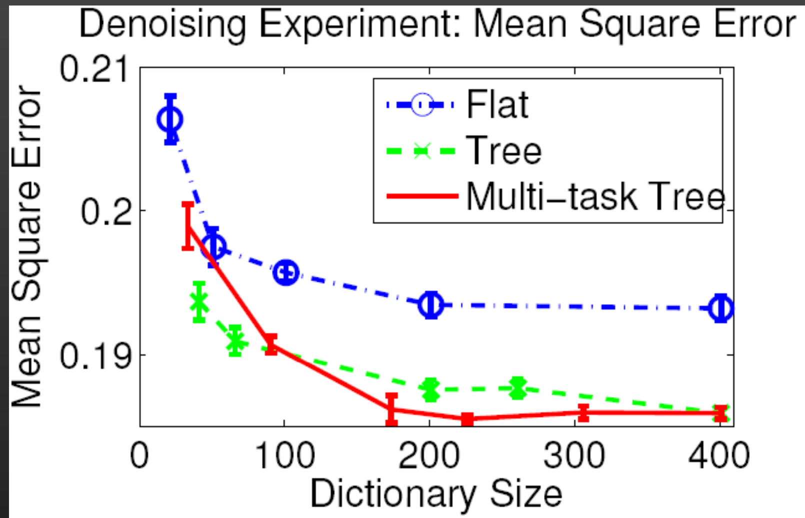


# Proximal methods for sparse hierarchical dictionary learning
(Jenatton, Mairal, Obozinski, Bach, ICML'10)



# Network flow algorithms for structured sparsity
(Mairal, Jenatton, Obozinski, Bach, NIPS'11)





SPArse Modeling software (SPAMS)

http://www.di.ens.fr/willow/SPAMS/

Tutorials on sparse coding and dictionary learning for image analysis

ICCV'09: www.di.ens.fr/~mairal/tutorial_iccv09/
NIPS'09: www.di.ens.fr/~fbach/nips2009tutorial/
CVPR'10: www.di.ens.fr/~mairal/tutorial_cvpr2010/