Sparse component analysis: sparse coding, inpainting, denoising and segmentation

Ivica Kopriva

Ruđer Bošković Institute

e-mail: ikopriva@irb.hr ikopriva@gmail.com
Web: http://www.lair.irb.hr/ikopriva/

Talk outline

- Instantaneous blind source separation
- Independent component analysis based approach to sparse coding (dictionary learning)
- Sparse component analysis
- Unsupervised segmentation of multispectral image
- Inpainting and denoising

Instantaneous blind source separation

Problem:

$$\mathbf{X}=\mathbf{AS} \quad \mathbf{X}\in R^{NxT},\ \mathbf{A}\in R^{NxM},\ \mathbf{S}\in R^{MxT}$$

Goal: find **A** and **S** based on **X** only, whereas number of sources *M* is unknown and can be less than, equal to or greater than number of measurements *N*. Herein, *T* stands for number of samples.

Solution $\mathbf{X}=\mathbf{AT}^{-1}\mathbf{TS}$ must be characterized with $\mathbf{T}=\mathbf{P}\Lambda$ where **P** is permutation and Λ is diagonal matrix i.e.: $\mathbf{Y}\cong\mathbf{P}\Lambda\mathbf{S}$

A. Hyvarinen, J. Karhunen, E. Oja, “Independent Component Analysis,” John Wiley, 2001.
A. Cichocki, S. Amari, “Adaptive Blind Signal and Image Processing,” John Wiley, 2002.
P. Comon, C. Jutten, editors, “Handbook of Blind Source Separation,” Elsevier, 2010.

Independent component analysis (ICA)

•source signals $s_i(t)$ must be statistically independent.

$$\mathrm{p}(\mathbf{s}) = \prod_{m=1}^{M} p_m(s_m)$$

•source signals $s_m(t)$, except one, must be non-Gaussian.

$$\left\{C_n(s_m) \neq 0\right\}_{m=1}^{M} \qquad \forall n > 2$$

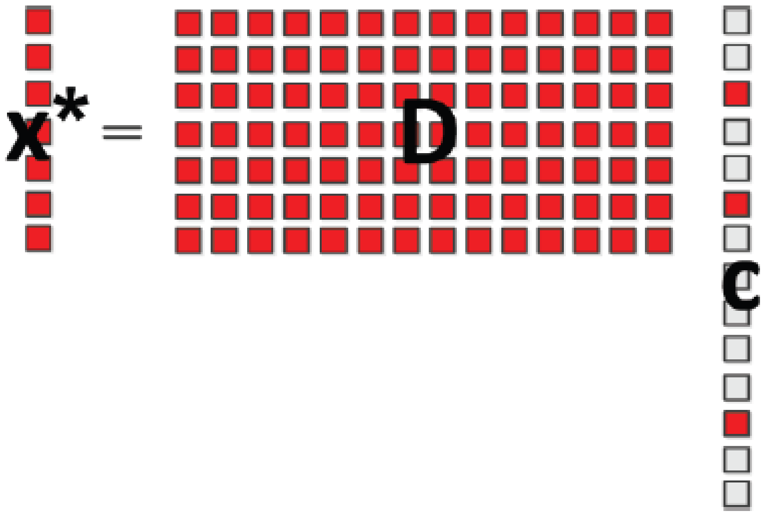
•mixing matrix **A** must be full column rank (number of mixtures *N* must be greater than or equal to number of sources *M*).

P. Common, “Independent Component Analysis – a new concept ?”, Signal Processing,36(3):287-314.
A. Hyvarinen, J. Karhunen, E. Oja, “Independent Component Analysis,” John Wiley, 2001.

ICA-based probabilistic approach to sparse coding

Sparse coding refers to a basis (dictionary), **D**, learning process such that signal **x** can be described in **D** by using few basis vectors (atoms) only, i.e.

$$\mathbf{x}=\mathbf{D}\mathbf{c}$$



where $\mathbf{x}\in R^{N}$, $\mathbf{D}\in R^{N\times M}$, $\mathbf{c}\in R^{M}$, $M\geq N$ and $\|\mathbf{c}\|_0=k$ such that $k<<M$.

When $M>N$ dictionary is overcomplete. That is of practical interest since results in inpainting and denoising are better when dictionary is overcomplete (a frame).

ICA-based probabilistic approach to sparse coding

ICA is applied for solving instantaneous BSS problem:

$$\mathbf{X}=\mathbf{AS},\quad \mathbf{A}\in R^{N\times M}\ \mathbf{S}\in R^{M\times T}$$

"Classical" ICA methods solve complete (determined and over-determined) BSS problems: $M \leq N$. That was one of the main arguments against using ICA for sparse coding/dictionary learning.

Some ICA algorithm such as FastICA[a] can be extended to overcomplete problems[b]. The condition is that components (sources) have sparse distributions. Hence, ICA can be useful in solving sparse coding problem.[c]

[a]A. Hyvärinen, E. Oja, A fast fixed-point algorithm for independent component analysis, *Neural Comput*. 9 (1997), 1483-1492.

[b]A. Hyvärinen, R. Cristescu, E. Oja, A fast algorithm for estimating overcomplete ICA bases for image windows, in: Proc. Int. Joint Conf. on Neural Networks, Washington, D.C., 1999, 894-899.

[c]**M. Filipović, I. Kopriva (2011). A comparison of dictionary based approaches to inpainting and denoising with an emphasis to independent component analysis learned dictionaries, *Inverse Problems and Imaging*, vol. 5, No. 4, (2011), 815-841.**

ICA-based probabilistic approach to sparse coding

When blind source separation problem, **X=AS**, is solved by ICA we are looking for sources that are, possibly, statistically independent and non-Gaussian.

In information-theoretic ICA methods[a-c], statistical properties (distributions) of the sources are not precisely known. The learning equation $\mathbf{W} \cong \mathbf{A}^{-1}$ ($\mathbf{y}=\mathbf{W}\mathbf{x}$) has the form:

$$\mathbf{W}(k+1)=\mathbf{W}(k)+\eta\left[\mathbf{I}-E\left\{\varphi(\mathbf{y})\mathbf{y}^{\mathrm{T}}\right\}\right]\mathbf{W}(k)$$

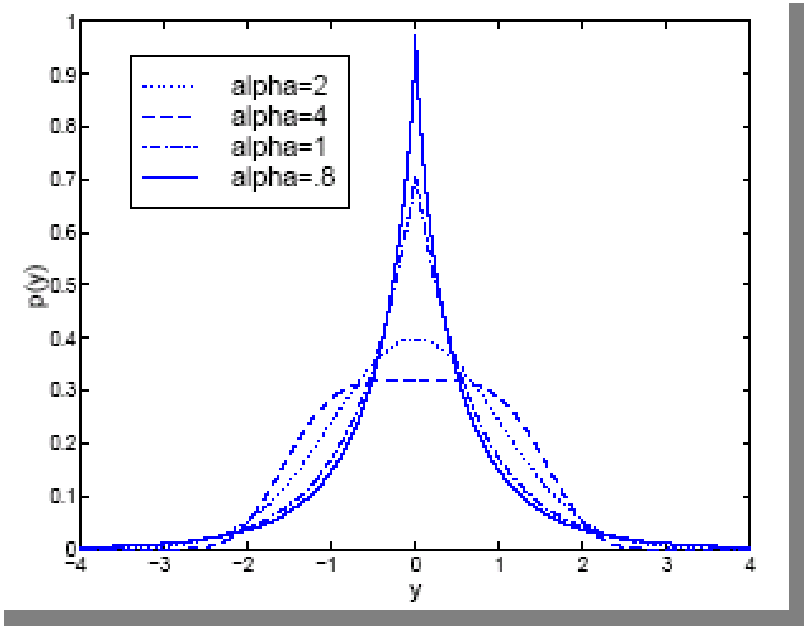
where $\varphi(\mathbf{y})=(-1/p_m)(dp_m/dy_m)$ is the *score* function.

[a]A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Comp.* 7, 1129-1159, 1995.
[b]D. T. Pham, "Blind separation of mixtures of independent sources through a quasimaximum likelihood approach," *IEEE Trans. Signal Processing* 45, pp. 1712-1725, 1997.
[c]D. Erdogmus, K. E. Hild II, Y. N. Rao and J.C. Principe, "Minimax Mutual Information Approach for Independent Component Analysis," *Neural Computation*, vol. 16, No. 6, pp. 1235-1252, June, 2004.

ICA-based probabilistic approach to sparse coding

The unknown density functions p_m can be parameterized, as an example, using generalized Gaussian density[a, b]

$$p_m(y_m) = \frac{\alpha_m}{2\sigma_m \Gamma(1/\alpha_m)} \exp\left(-\frac{1}{\alpha_m} \left| \frac{y_m}{\sigma_m} \right|^{\alpha_m} \right)$$



With the single parameter α_m (called Gaussian exponent) super-Gaussian distributions (α_m <2) and sub-Gaussian distributions (α_m >2) could be modeled.

[a]S. Choi, A. Cichocki, S. Amari, Flexible Independent Component Analysis," *J. VLSI Signal Process. Sys*. 26 (2000) 25-38.
[b]L. Zhang, A. Cichocki, S. Amari, Self-adaptive blind source separation based on activation function adaptation, *IEEE Trans. Neural Net.* 15 (2004) 233-244.

ICA-based probabilistic approach to sparse coding

If generalized Gaussian probability density function is inserted in the optimal form for score function the expression for flexible nonlinearity is obtained:

$$\varphi_m(y_m) = sign(y_m)\left|y_m\right|^{\alpha_m - 1}$$

If *a priori* knowledge about statistical distributions of the source signals is available α_m can be fixed in advance. For example if source signals are super-Gaussian α_m can be set to α_m=1.

In sparse coding problem **X=DC** we want code **C** to be sparse (by design). Hence, if **C** is interpreted as a source matrix and **D** as mixing/basis matrix we can *a priori* select the nonlinear score functions to generate sparse code.

Thus, sparse coding can be seen as optimally tuned BSS!!!. The basis D is learned/obtained as a byproduct.

ICA-based probabilistic approach to sparse coding

FastICA algorithm has information-theoretic interpretation and employs fixed-point maximization of the cost function that approximates negentropy of the code c_m:

$$J(c_m) \propto \left[\left\langle \left\{ G(c_m) \right\} \right\rangle - \left\langle \left\{ G(\nu) \right\} \right\rangle \right]^2$$

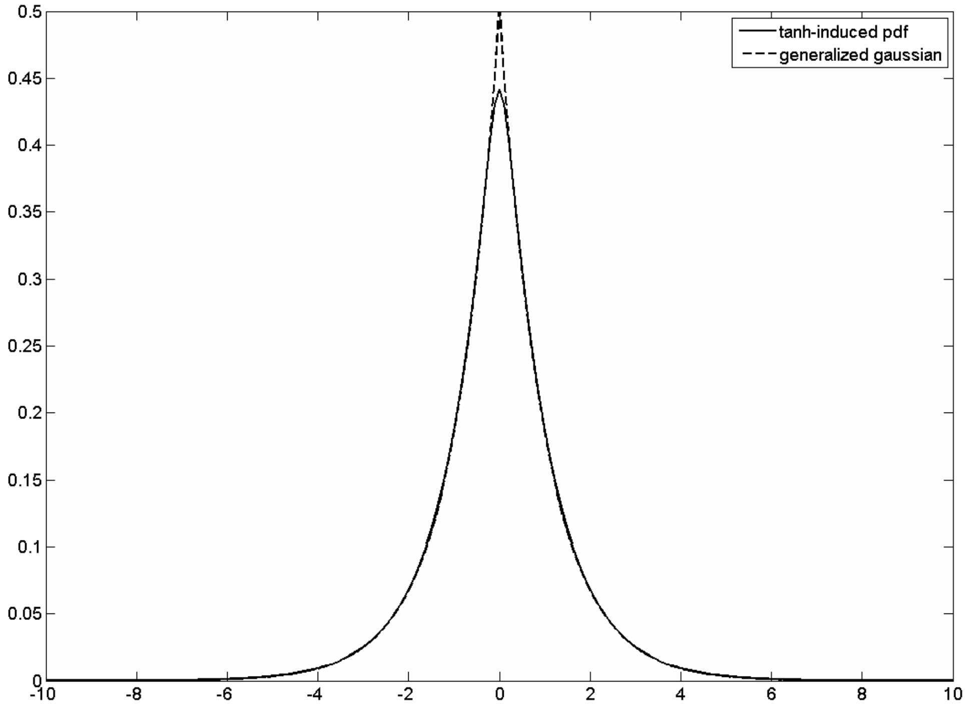
where ν is standardized Gaussian variable. Assuming that $G(c_m)=-\log(p_m(c_m))$ in marginal entropies of mutual information cost function $I(\mathbf{c})$, above approximation can be obtained from $I(\mathbf{c})$. Hence, nonlinear functions in FastICA can also be selected *a priori* to generate sparse (super-Gaussian) code c_m.

`tanh(a`c_m`)` nonlinearity is associated with $G(c_m)=(1/a)\log(\cosh(ac_m))$. In this case $G(c_m)$ approximates density function:

$$p(c_m) \sim \frac{1}{\left(\cosh(ac_m)\right)^{1/a}}$$

that models sparse distributions.

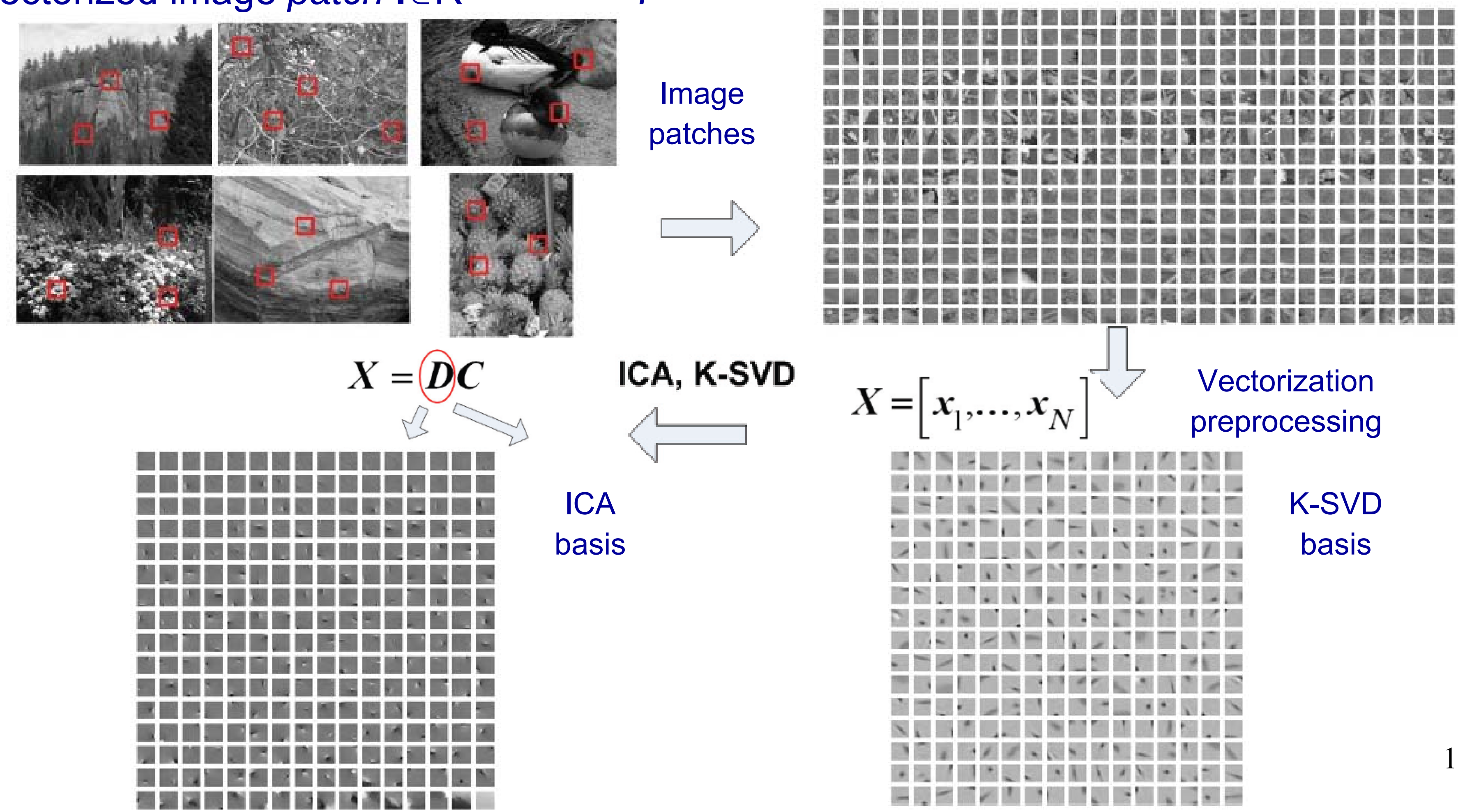
ICA-based probabilistic approach to sparse coding



Probability density functions induced by `tanh` nonlinearity with *a*=5 and generalized Gaussian pdf with α=1, which models Laplacian pdf.

ICA-based probabilistic approach to sparse coding

The dictionary learning problem is organized in the *patch* space. Then, **x** denotes vectorized image *patch* $\mathbf{I} \in R^{sqrt(N) \times sqrt(N)}$.



ICA-based probabilistic approach to sparse coding

The sparse coding constitutes mathematical reproduction of experimentally observed behavior that primary visual cortex area of the mammals’ brain processes visual information by receptive fields (neurons) that respond to localized oriented edges in visual scenes.[a]

The first computational model in the literature partially explained experimental evidence has been presented in reference *b*. The underlying principle behind sparse coding models is the principle of efficient coding,[c] which assumes that organisms are adapted to maximize efficiency of information processing (representing visual scene by few basis elements / receptive fields).

It is evident that learned basis vectors also represent texture information contained in the image. Thus, computational methods of sparse coding are also useful for feature extraction.

[a]D. H. Hubel, T. N. Wiesel, “Receptive fields and functional architecture of monkey striate cortex,” *The Journal of Physiology* (1968), pp. 215-243.
[b]B. A. Olshausen, D. J. Field, “Emergence of simple-cell receptive field properties by a learning sparse code of natural images,” *Nature* **381** (1996), 607-609.
[c] M. Rehn, F. T. Sommer, “A network that uses few active neurons to code visual input predicts the diverse shapes of corctical receptive fields,” *J. Comp. Neurosc.* (2007), 135-146.

Sparse component analysis

Underdetermined BSS occurs when number of measurements *N* is less than number of sources *M*. Resulting system of linear equations

$$\mathbf{x}=\mathbf{A}\mathbf{s}$$

is underdetermined. Without constraints on **s** unique solution does not exist <u>even if **A** is known</u>:

$$\mathbf{s}=\mathbf{s}_p + \mathbf{s}_h = \mathbf{A}^{\dagger}\mathbf{x} + \mathbf{V}\mathbf{z} \quad \mathbf{A}\mathbf{V}\mathbf{z}_h=\mathbf{0}$$

where **V** spans *M*-*N* dimensional null-space of **A**.

However, if **s** is <u>*sparse*</u> enough **A** can be identified and unique solution for **s** can be obtained. That is known as <u>*sparse component analysis*</u> (SCA).

Sparse component analysis

Provided that *prior* on **s**(*t*) is Laplacian, maximum likelihood approach to maximization of posterior probability *P*(**s**|**x**,**A**) yields minimum L_1-norm as the solution:

$$\begin{aligned}
\hat{\mathbf{s}}(t) &= \max_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} P\left(\mathbf{s}(t)\middle|\mathbf{x}(t),\hat{\mathbf{A}}\right) \\
&= \max_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} P\left(\mathbf{x}(t)\middle|\mathbf{s}(t),\hat{\mathbf{A}}\right)P\left(\mathbf{s}(t)\right) \\
&\propto \max_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} P\left(\mathbf{s}(t)\right) \\
&= \max_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} \exp-\left(\left|\mathbf{s}_1(t)\right|+\ldots+\left|\mathbf{s}_M(t)\right|\right) \\
&= \min_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} \left|\mathbf{s}_1(t)\right|+\ldots+\left|\mathbf{s}_M(t)\right| \\
&= \min_{\hat{\mathbf{A}}\mathbf{s}(t)=\mathbf{x}(t)} \left\|\mathbf{s}(t)\right\|_1
\end{aligned}$$

Sparse component analysis

SCA-based solution of the uBSS problem is obtained in two stages:

1) estimate basis or mixing matrix **A** using data clustering, ref. a.

2) estimating sources **s** solving underdetermined linear systems of equations **x=As**. Provided that **s** is sparse enough, solution is obtained at the minimum of L_1-norm, ref. *b* and *c*.

[a] F. M. Naini, G.H. Mohimani, M. Babaie-Zadeh, Ch. Jutten, "Estimating the mixing matrix in Sparse Component Analysis (SCA) based on partial *k*-dimensional subspace clustering," *Neurocomputing* **71** (2008), 2330-2343.
[b] Y. Li, A. Cichocki, S. Amari, "Analysis of Sparse Representation and Blind Source Separation," *Neural Computation* **16** (2004), 1193-1234.
[c] D. L. Donoho, M. Elad, "Optimally sparse representation in general (non-orthogonal) dictionaries via l_1 minimization," *Proc. Nat. Acad. Sci.* **100** (2003), 2197-2202.

Sparse component analysis

- Solving underdetermined system of linear equations **x=As** amounts to solving:

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}(t)} \|\mathbf{s}(t)\|_0 \quad \text{s.t. } \hat{\mathbf{A}}\mathbf{s}(t) = \mathbf{x}(t) \quad \forall t = 1,\dots,T$$

or for problems with noise or approximation error:

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}(t)} \frac{1}{2}\left\|\hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t)\right\|_2^2 + \lambda \|\mathbf{s}(t)\|_0 \quad \forall t = 1,\dots,T$$

$$\hat{\mathbf{s}}(t) = \arg\min_{\mathbf{s}(t)} \|\mathbf{s}(t)\|_0 \quad \text{s.t. } \left\|\hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t)\right\|_2^2 \le \varepsilon \quad \forall t = 1,\dots,T$$

Minimization of L_0–norm of **s** is combinatorial problem that is NP-hard. For larger dimension *M* it becomes computationally infeasible. Moreover, minimization of L_0–norm is very sensitive to noise i.e. presence of small coefficients

Sparse component analysis

Replacement of L_0-norm by L_1-norm is done quite often. That is known as *convex relaxation* of the minimum L_0-norm problem. This leads to linear program:

$$\hat{\mathbf{s}}(t) = \underset{\mathbf{s}(t)}{\arg\min} \sum_{m=1}^{\hat{M}} s_m(t) \quad \text{s.t. } \hat{\mathbf{A}}\mathbf{s}(t) = \mathbf{x}(t) \quad \forall t = 1,\ldots,T$$

$$\text{s.t. } \mathbf{s}(t) \geq 0$$

L_1-regularized least square problem, ref. a, b:

$$\hat{\mathbf{s}}(t) = \underset{\mathbf{s}(t)}{\arg\min} \frac{1}{2} \left\| \hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t) \right\|_2^2 + \lambda \left\| \mathbf{s}(t) \right\|_1 \quad \forall t = 1,\ldots,T$$

and L_2-regularized linear problem, ref. b, c:

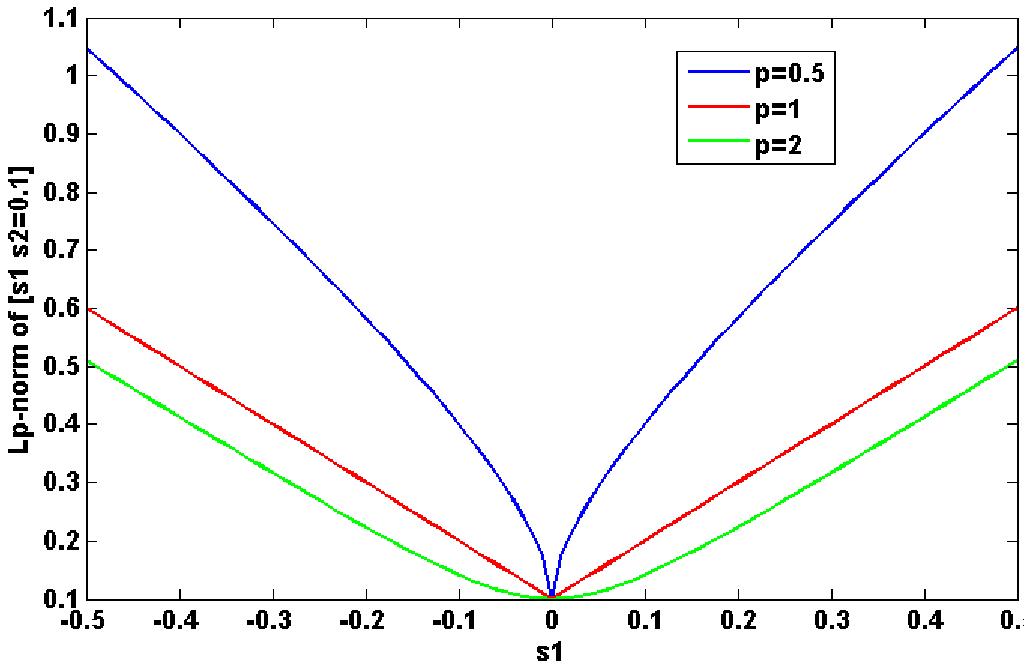
$$\hat{\mathbf{s}}(t) = \underset{\mathbf{s}(t)}{\arg\min} \left\| \mathbf{s}(t) \right\|_1 \quad \text{s.t. } \left\| \hat{\mathbf{A}}\mathbf{s}(t) - \mathbf{x}(t) \right\|_2^2 \leq \varepsilon \quad \forall t = 1,\ldots,T$$

[a] S..J. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, "An Interior-Point Method for Large-Scale -Regularized Least Squares,“ *IEEE Journal of Selected Topics in Signal Processing* **1**, 606-617 (2007), **http://www.stanford.edu/~boyd/l1_ls/.**
[b] E. van den Berg, M.P. Friedlander, "Probing the Pareto Frontier for Basis Pursuit Solutions," *SIAM J. Sci. Comput*. 31, 890-912 (2008).
[c] M.A.T. Figuiredo, R.D. Nowak, S.J. Wright, "Gradient Projection for Sparse Reconstruction: Application to Compressed Sensing and Other Inverse Problems," *IEEE Journal on Selected Topics in Signal Processing* **1**, 586-597 (2007).

SCA : L_p norm minimization: $0 < p \leq 1$

Minimizing L_p-norm, $0<p<1$, of **s** yields better performance when solving underdetermined system **x=As** than when using L_1-norm minimization.

This occurs despite the fact that minimization of L_p-norm, $0<p<1$ is *non-convex* problem. Yet, in practical setting (when noise or approximation errors are present) its local minimum can be smaller than global minimum of L_1 i.e. min L_p-norm solution is sparser than min L_1-norm solution.



L_p-norm of $[s_1\ 0.1]$: $\|\mathbf{s}\|_p = \left(\sum_{m=1}^{M} |s_m|^p \right)^{1/p}$

SCA – L_p norm minimization: $0< p \leq 1$

The idea of ref. *a* was to replace L_0-norm by continuous parametric approximation:

$$\|\mathbf{s}\|_0 \approx M - F_\sigma(\mathbf{s})$$

where:

$$F_\sigma(\mathbf{s}) = \sum_m f_\sigma(s_m)$$

and:

$$f_\sigma(s_m) = \exp\left(-\frac{s_m^2}{2\sigma^2}\right)$$

approximates indicator function of a set {0}.

[a]H. Mohimani, M. Babaie-Zadeh, C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed L_0 norm," *IEEE Trans. Signal Process.* 57 (2009) 289-301.

SCA – L_p norm minimization: $0< p \leq 1$

Smaller parameter σ brings us closer to $L_0(\mathbf{s})$, while larger σ yields smoother approximation that is easier to optimize.

Minimizing approximation of $L_0(\mathbf{s})$ is equivalent to maximize $F_\sigma(\mathbf{s})$. The idea is to maximize $F_\sigma(\mathbf{s})$ for large σ and than use obtained solution as initial value for next maximization of $F_\sigma(\mathbf{s})$ for smaller σ.

Matlab code for smooth L_0 algorithm can be downloaded from:

http://ee.sharif.ir/~SLzero/

SCA – L_p norm minimization: $0< p \leq 1$

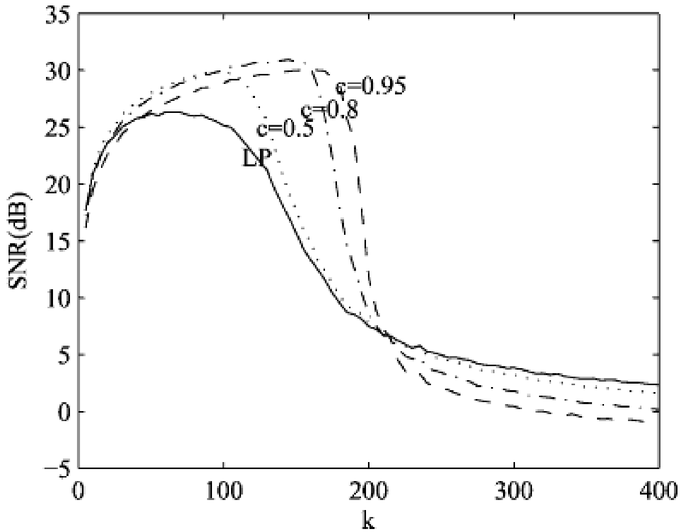


Fig. 6. Averaged SNRs (over 100 runs of the algorithm) versus k, the average number of active sources, for SL0 algorithm with several values of c, and for LP. The parameters are $m = 1000$, $n = 400$, $\sigma_1 = 1$, $\sigma_J = 0.01$, $\sigma_n = 0.01$.

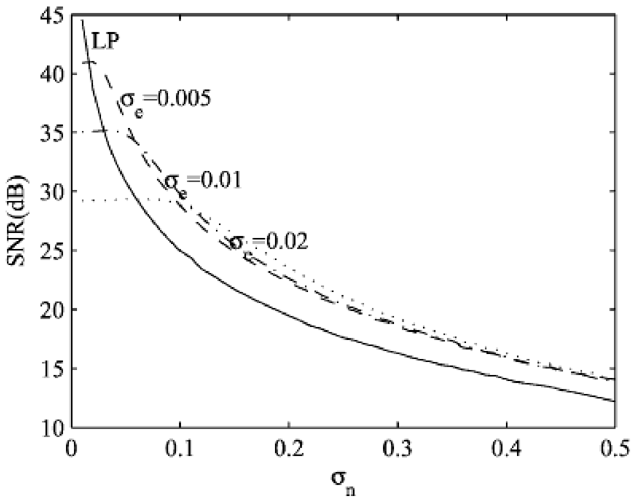
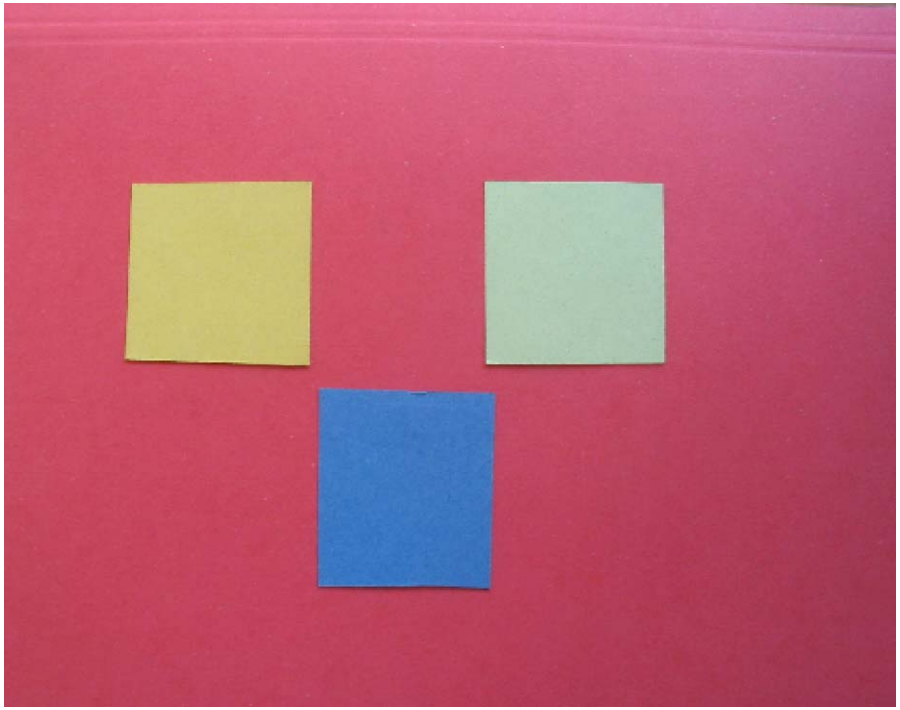


Fig. 7. Averaged SNRs (over 100 runs of the algorithm) versus the noise power σ_n for different values of σ_J, and for LP. The parameters are $m = 100$, $n = 400$, $k = 100$, $\sigma_1 = 1$, and $c = 0.8$.

$$SNR[dB] = 20\log\left(\frac{\mathbf{s}}{\mathbf{s}-\hat{\mathbf{s}}}\right)$$

Unsupervised segmentation of multispectral images

Consider blind decomposition of the RGB image (N=3) composed of four materials (M=4):



I. Kopriva and A . Cichocki, "Sparse component analysis-based non-probabilistic blind decomposition of low-dimensional multi-spectral images," *Journal of Chemometrics*, vol. 23, Issue 11, pp. 590-597 (2009).

Unsupervised segmentation of multispectral images

For image consisting of *N* spectral bands and *M* objects linear data model is assumed:

$$\mathbf{x} = \mathbf{A}\mathbf{s} = \sum_{m=1}^{M} \mathbf{a}_m s_m \qquad \begin{bmatrix} \mathbf{a}_1 & \mathbf{a}_2 & \ldots & \mathbf{a}_M \end{bmatrix} \equiv \mathbf{A}$$

x - measured data intensity vector, $\mathbf{x} \in R^{N\times 1}$

$\mathbf{A} \in R^{N\times M}$: unknown spectral reflectance matrix: column vectors are also called endmembers (they represent spectral profiles of the objects/materials present in the image). Non-singularity condition implies $\mathbf{a}_i \neq \mathbf{a}_j$.

$\mathbf{s} \in R^{M\times 1}$: s_m are called abundances. If constraint $\sum_{m=1}^{M} s_m = 1$ is applied sources

represent percentage of objects presence in the pixel footprint. Sources s_m could be recovered by unsupervised and properly constrained factorization of **X**.

Unsupervised segmentation of multispectral images

Evidently degree of overlap between materials in spatial domain is very small i.e. $s_m(t)*s_n(t)\approx\delta_{nm}$. Hence RGB image decomposition problem can be solved with some SCA algorithm. Here, clustering and sparseness constrained minimization of L_p-norm have been used.

For the L_p-norm minimization approach estimate of the mixing **A** and number of materials M is necessary.

Because materials in principle do not overlap in spatial domain it applies $\|\mathbf{s}(t)\|_0\approx 1$.

Unsupervised segmentation of multispectral images

Assuming unit L_2-norm of $\boldsymbol{a}_{\mathbf{m}}$ we can parameterize column vectors in 3D space by means of azimuth and elevation angles

$$\mathbf{a}_m = [\cos(\varphi_m)\sin(\theta_m) \ \ \sin(\varphi_m)\sin(\theta_m) \ \ \cos(\theta_m)]^{\mathrm{T}}$$

Due to nonnegativity constraints both angles are confined in [0,π/2]. Now estimation of **A** and *M* is obtained by means of data clustering algorithm:

We remove all data points close to the origin for which applies: $\left\{\left\|\mathbf{x}(t)\right\|_2 \leq \varepsilon\right\}_{t=1}^{T}$ where ε represents some predefined threshold.

Normalize to unit L_2-norm remaining data points **x**(*t*), i.e., $\left\{\mathbf{x}(t) \rightarrow \mathbf{x}(t) / \left\|\mathbf{x}(t)\right\|_2\right\}_{t=1}^{\bar{T}}$

Unsupervised segmentation of multispectral images

Calculate function *f*(**a**):

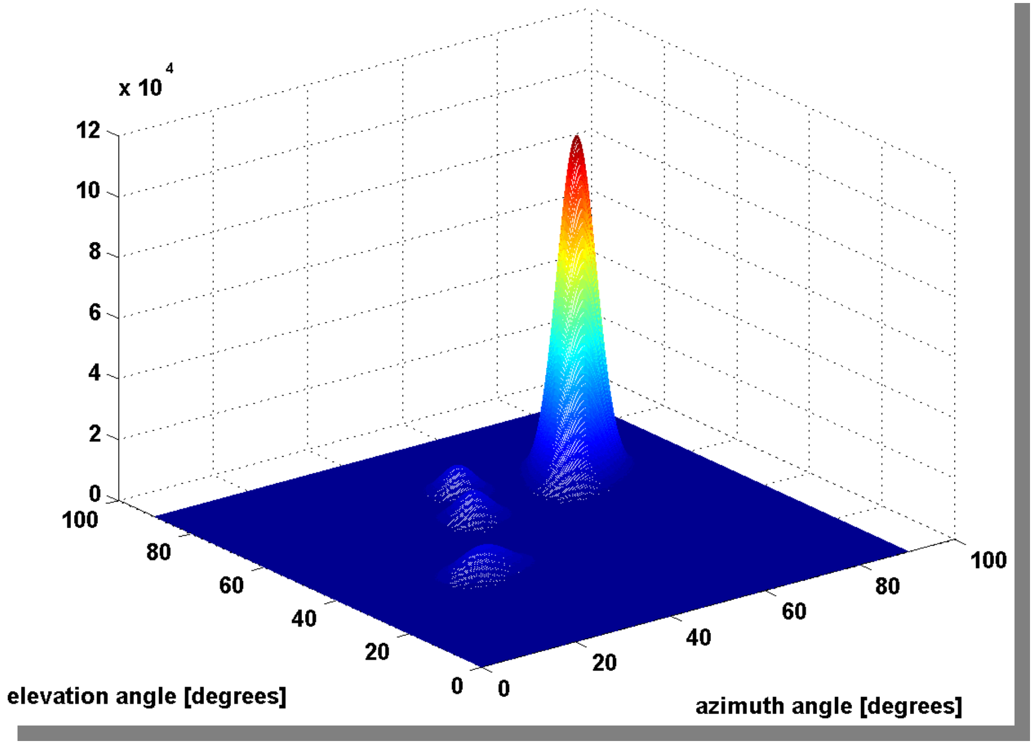
$$f(\mathbf{a}) = \sum_{t=1}^{\bar{T}} \exp\left(-\frac{d^2(\mathbf{x}(t),\mathbf{a})}{2\sigma^2}\right)$$

where $d(\mathbf{x}(t),\mathbf{a}) = \sqrt{1-(\mathbf{x}(t)\cdot\mathbf{a})^2}$ and $(\mathbf{x}(t)\cdot\mathbf{a})$ denotes inner product. Parameter σ is called dispersion. If set to sufficiently small value the value of the function *f*(**a**) will approximately equal the number of data points close to **a**. Thus by varying mixing angles $0\leq\varphi,\theta\leq\pi/2$ we effectively cluster data.

Number of peaks of the function *f*(**a**) corresponds with the estimated number of materials *M*. Locations of the peaks correspond with the estimates of the mixing angles $\left\{\left(\hat{\varphi}_m,\hat{\theta}_m\right)\right\}_{m=1}^{\hat{M}}$, i.e., mixing vectors $\left\{\hat{\mathbf{a}}_m\left(\hat{\varphi}_m,\hat{\theta}_m\right)\right\}_{m=1}^{\hat{M}}$.

Unsupervised segmentation of multispectral images

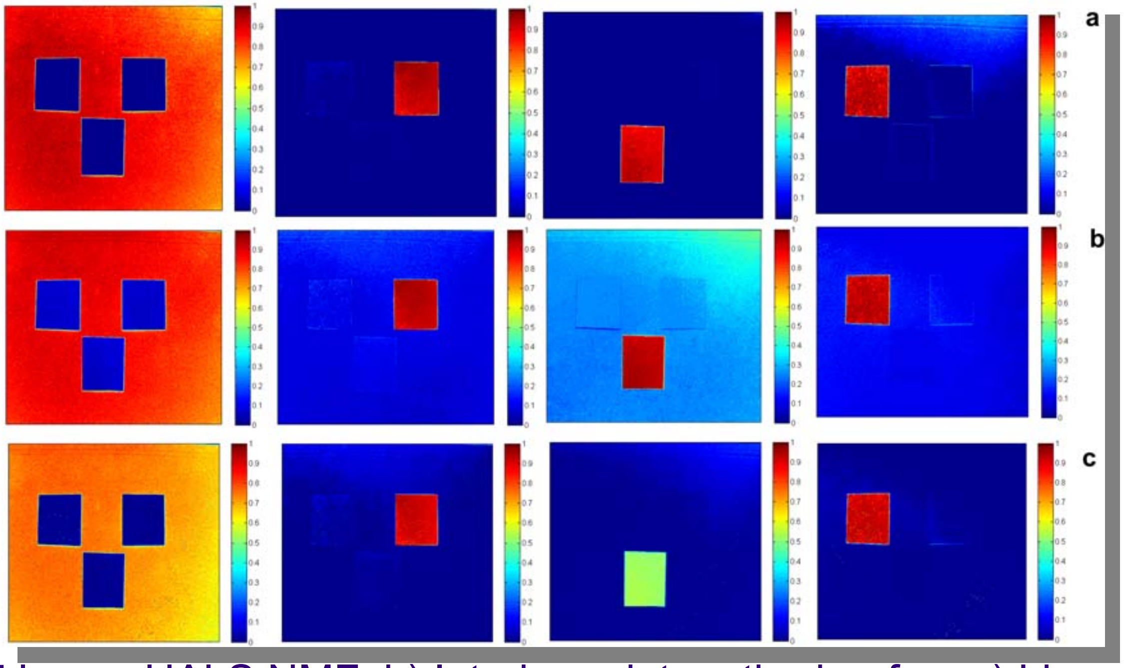
For shown experimental RGB image clustering function is obtained as:



Four peaks suggest existence of four materials in the RGB image **i.e. *M*=4**.

Unsupervised segmentation of multispectral images

Spatial maps of the materials extracted by HALS NMF with 25 layers,[a] linear programming and interior point method[b] are obtained as:



a) 25 layers HALS NMF; b) Interior point method, ref. a; c) Linear programming.

[a] A. Cichocki, R. Zdunek, S.I. Amari, Hierarchical ALS Algorithms for Nonnegative Matrix Factorization and 3D Tensor Factorization, LNCS **4666** (2007) 169-176

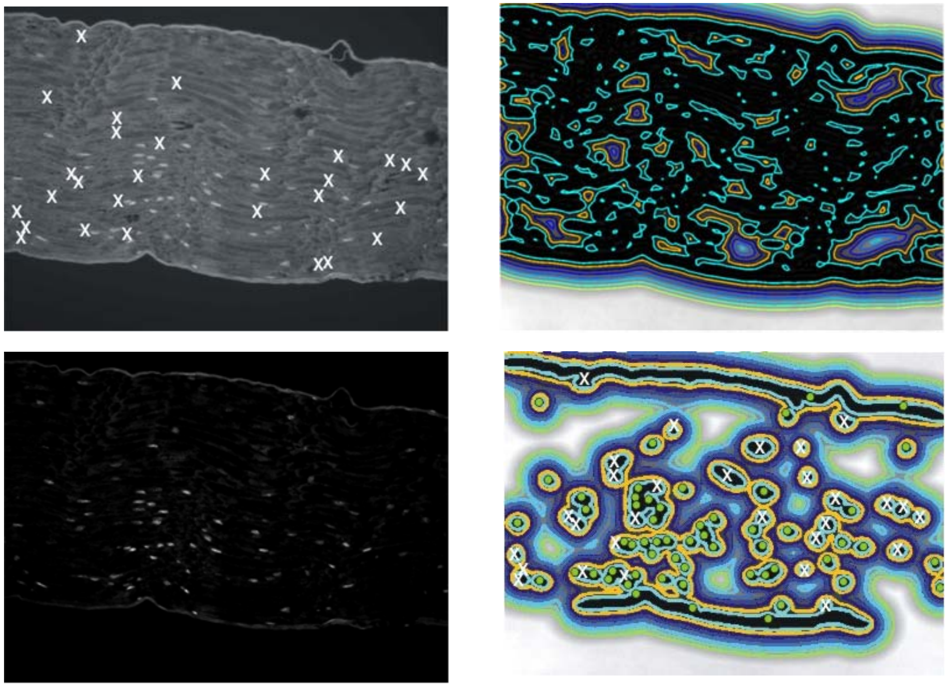
[b] S. J. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, "An Interior-Point Method for Large-Scale L_1 -Regularized Least Squares,"IEEE Journal of Selected Topics in Signal Processing **1**, 606-617 (2007). http://www.stanford.edu/~boyd/l1_ls/.

Unsupervised segmentation of multispectral images

Sparseness constrained NMF in combination with nonlinear dimensionality expansion mapping yields contrast-enhanced segmentation of the RGB image of *unstained* specimen in histopathology.[a]

First row: gray scale (green component) image of the *nervus ischiadicus* obtained without contrast agent (i.e. staining) and corresponding iso-controur map.

Second row: corresponding component after decomposition and corresponding iso-contour map.



[a]I. Kopriva, M. Hadžija, M. Popović-Hadžija, M. Korolija, A. Cichocki (2011). Rational Variety Mapping for Contrast-Enhanced Nonlinear Unsupervised Segmentation of Multispectral Images of Unstained Specimen, *American Journal of Pathology*, vol. 179, No. 2, pp. 547-553.

Inpainting

Inpainting problem is described as:

$$\mathbf{y}=\mathbf{Mx}=\mathbf{MDc}=\Phi\mathbf{c}$$

where column-wise vectorized image $\mathbf{x}\in R^N$ is to be reconstructed from the vector of known pixels $\mathbf{y}\in R^L$, $L < N$. $\mathbf{M}\in R^{L\times N}$ contains 0s and 1s and is representing layout of the missing values. $\mathbf{D}\in R^{N\times M}$, $\Phi\in R^{L\times M}$ $\mathbf{c}\in R^M$ $M\geq N>L$.

Hence, above system of equations is underdetemined. Smoothed L_0 algorithm[a] has been used in experiments performed in ref. *b* and reported herein.

[a]H. Mohimani, M. Babaie-Zadeh, C. Jutten, "A fast approach for overcomplete sparse decomposition based on smoothed L_0 norm," IEEE Trans. Signal Process. 57 (2009) 289-301.
[b]**M. Filipović, I. Kopriva (2011). A comparison of dictionary based approaches to inpainting and denoising with an emphasis to independent component analysis learned dictionaries, *Inverse Problems and Imaging*, vol. 5, No. 4, 815-841.**

Denoising (high density) salt and pepper noise

Denoising problem: $\mathbf{y}=\mathbf{x}+\mathbf{n}$ → Estimate $\mathbf{x}$ based on $\mathbf{y}$. For $\mathbf{n}\sim N(0,\sigma^2)$ maximizing *a posteriori* probability yields:

$$\hat{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2\sigma^2}\left\|\mathbf{y}-\mathbf{x}\right\|_2^2 - \phi\left(\mathbf{x}\right)$$

where $\phi(\mathbf{x})$ is prior on $\mathbf{x}$. However, for impulsive noise variance is infinite. Likelihood $p(\mathbf{y}|\mathbf{x})$ is heavy tailed. Very often used model is Cauchy density but analytic form may even not exist.

Nonlinear filters such as *median* and *myriad* filters are optimal if noise distribution is respectively Lalpace or Cauchy.[a] Image is processed locally by windows of *L*x*L.* The filtered pixel value is weighted *median* or weighted *myriad* in *L*x*L* neighborhood of the pixel. <u>For high density noise the window size *L* has to be increased and that leads to blurring</u>.

[a]G. R. Arce, *Nonlinear Signal Processing - A Statistical Approach*, J. Wiley, 2005.

Denoising (high density) salt and pepper noise

Salt and pepper noise corrupts image in a way that pixels are very bright (salt) or very dark (pepper). Sometimes denoising of such pixels is called desaturation. [a,b]

Such pixels are easy to detect and can be declared as missing. Thus, denoising of (high density) salt and pepper noise becomes effectively "noiseless" inpainting problem.

While method in ref. *a* has been using fixed basis we provide solution in learned dictionary. While method in ref. *b* relies on correlation structure of the image we do not use any specific assumption in this regard.

[a] H. Mansour, R. Saab, P. Nasiopoulos, R. Ward, "Color image desaturation using sparse reconstruction," in *Proc. 2010 ICASP*, Dallas, TX, USA, (2010), 778-781.
[b] X. Zhang, D. H. Brainard, "Estimation of saturated pixel values in digital color imaging," *J. Opt. Soc. Amer. A*, **21** (2004), 2301-2310.

Experiments: basis learning

All the experiments have been run in MATLAB 7.7 environment on a 3 GHz dual core processor with 2GB of RAM.

We have used six training images[a] to extract randomly 18000 patches (3000 per image) of the size 16x16 pixels. All the patches were vectorized and mean has been removed from each patch. The data matrix **X** was of the size 256x18000.

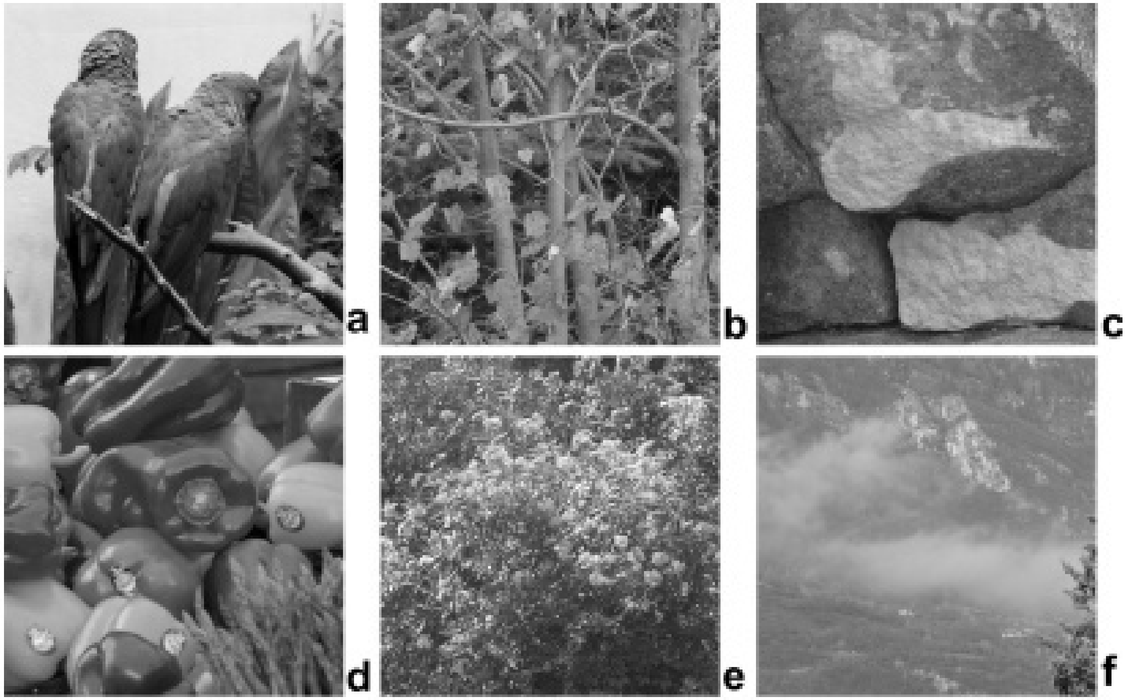
The patch size in the experiments varies between 8x8 to 16x16 pixels.

256x256 basis has been learned by K-SVD (with 40 nonzero coefficients of the code) in 5 hours and by FastICA in 3 hours.

[a] A. Olmos, F. A. A. Kingdom, McGill calibrated colour image database, 2004., http://pirsquared.org/research/mcgilldb/

Experiments: validation set

Six images were selected randomly from the same web site to perform comparative performance analysis of the methods on various types of inpainting problems.



Experiments: performance measures

As performance measures we have used PSNR in [dB] and structural similarity index (SSIM)[a,b] .

The SSIM better corresponds/reflects subjective quality of visual perception. PSNR (it is based on mean square error) can give high values that not always correspond well with the subjective quality of visual perception.

SSIM has values between -1 and 1, achieving maximal value 1 if and only if the images being compared are equal. MATLAB code for computing the SSIM is available at: http://www.ece.uwaterloo.ca/~z70wang/research/ssim/

[a] Z. Wang, A. Bovik, Mean squared error: Love it or leave it? A new look at signal fidelity measures, IEEE Signal Process. Mag. 26(1) (2009) 98-117.
[b] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13(4) (2004) 600-612.

Experiments: image reconstruction details

In all reported image reconstruction (inpainting) experiments the smoothed L_0 (SL0) algorithm[a,b] has been used.

Numerical experiments have been performed to compare SL0 algorithm against OMP algorithm and interior point method[c,d] used to solve L_1-regularized least square problem.

On average, reconstruction using L_1-ls took 10 to 15 minutes per image, while the SL0 took around 30 seconds per image only. The OMP yielded significantly worse quality of reconstructed images.

[a] H. Mohimani, M. Babaie-Zadeh, C. Jutten, A fast approach for overcomplete sparse decomposition based on smoothed norm, IEEE Trans. Signal Process. 57 (2009) 289-301.
[b] http://ee.sharif.ir/~SLzero/
[c] S. J. Kim, K. Koh, M. Lustig, S. Boyd, D. Gorinevsky, An Interior-Point Method for Large-Scale -Regularized Least Squares, IEEE J. Sel. Top. Signal Process. 1 (2007), 606-617.
[d] http://www.stanford.edu/~boyd/l1_ls/

Experiments: image reconstruction details

For each patch, before reconstruction, the mean value has been removed and added back after reconstruction. Thus, DC component was artificially added in reconstruction yielding better results than if it has been a part of the basis.

To prevent border effects, reconstruction has been done such that the adjacent patches overlapped in two rows and two columns. After reconstruction overlapping regions were averaged.

In comparative performance analysis we have also used the FoE and MCA methods. MCA used curvelet dictionary for the cartoon part and 2D cosine packets for the texture part of the image. Parameters used with MCA were as specified in the MCALab package (32x32 patches for cosine packets, coarsest scale for curvelets 2, iterative hard thresholding with 300 iterations for reconstruction). For FoE method default parameters were used, as suggested in the MATLAB package, as well.

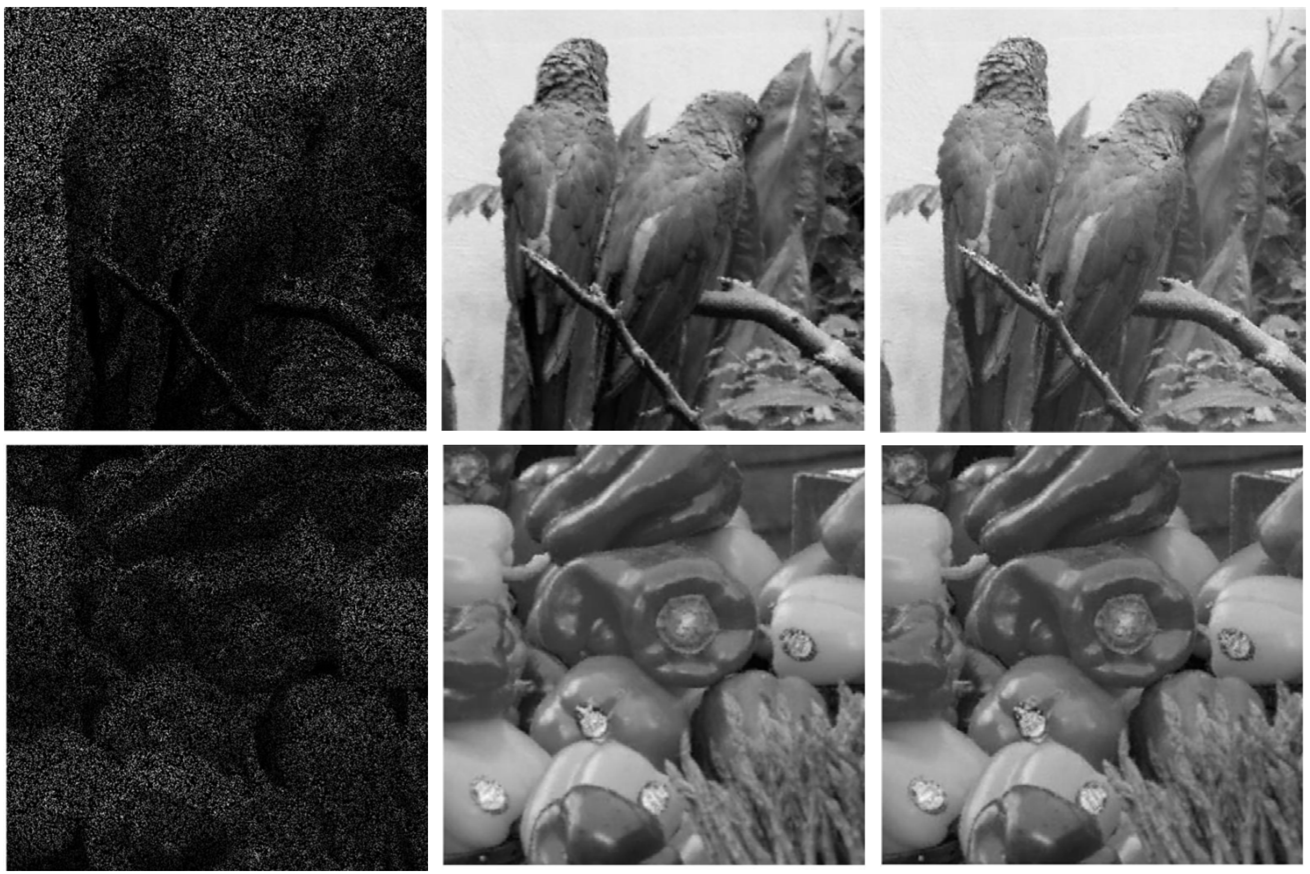
Experiments: image reconstruction details

For a random pattern of missing pixels each inpainting experiment has been repeated 10 times and the final performance measure has been obtained as an average.

Since MCA and FoE were slow (the MCA took around 50 minutes per image, while the FoE took around 5 hours per image) the inpainting experiments have been not repeated 10 times for these methods.

Experiments: denoising 80% dense pepper noise

Salt and pepper noise generates random pattern of missing values and that is the easiest inpainting problem to solve. **Complete basis: 256**x**256**.



Corrupted images ICA basis K-SVD basis

Experiments: denoising 80% dense pepper noise

Complete basis: 256x256.

TABLE 1. Inpainting results for the complete bases in terms of the SSIM metric for the complete bases learned on patches of size 16×16 pixels.

	ICA	**K-SVD**	**DCT**	**Symmlet 4 wavelet**	**MCA**	**FoE**
Fig. 6a	0.907 ± 0.0008	0.905 ± 0.001	0.75 ± 0.0008	0.736 ± 0.0022	0.789	0.92
Fig. 6b	0.76 ± 0.0016	0.749 ± 0.0012	0.55 ± 0.0015	0.503 ± 0.0015	0.682	0.77
Fig. 6c	0.773 ± 0.0007	0.766 ± 0.0011	0.617 ± 0.0011	0.562±0.003	0.644	0.78
Fig. 6d	0.944 ± 0.0005	0.94 ± 0.0004	0.81 ± 0.0007	0.81±0.0017	0.854	0.95
Fig. 6e	0.6 ± 0.002	0.577 ± 0.0015	0.434 ± 0.0015	0.35±0.0022	0.491	0.6
Fig. 6f	0.919 ± 0.0006	0.917 ± 0.0005	0.84 ± 0.0003	0.812 ± 0.0009	0.852	0.92
Mean	**0.817 ± 0.001**	**0.809 ± 0.0009**	**0.666 ± 0.0008**	**0.63±0.002**	**0.719**	**0.824**

TABLE 2. Inpainting results in terms of the PSNR metric for the complete bases learned on patches of size 16×16 pixels. The values are in dB.

	ICA	**K-SVD**	**DCT**	**Symmlet 4 wavelet**	**MCA**	**FoE**
Fig. 6a	30.2 ± 0.07	30.2 ± 0.08	25.4 ± 0.02	23.4 ± 0.1	27.1	30.9
Fig. 6b	24.4 ± 0.04	24.1 ± 0.04	21.6 ± 0.02	19.8 ± 0.03	23.6	24.7
Fig. 6c	29.7 ± 0.01	29.3 ± 0.02	27.1 ± 0.02	25.6 ± 0.05	27.6	29.8
Fig. 6d	34.3 ± 0.08	34.4 ± 0.06	28.4 ± 0.05	27 ± 0.1	30.3	35.5
Fig. 6e	19.5 ± 0.03	18.8 ± 0.03	18.2 ± 0.01	16 ± 0.01	18.4	19.5
Fig. 6f	33.4 ± 0.09	32.9 ± 0.07	30.5 ± 0.02	28.6 ± 0.09	30.7	33.1
Mean	**28.6 ± 0.05**	**28.3 ± 0.05**	**25.2 ± 0.03**	**23.4 ± 0.07**	**26.3**	**28.9**

Experiments: inpainting missing pattern with a block structure

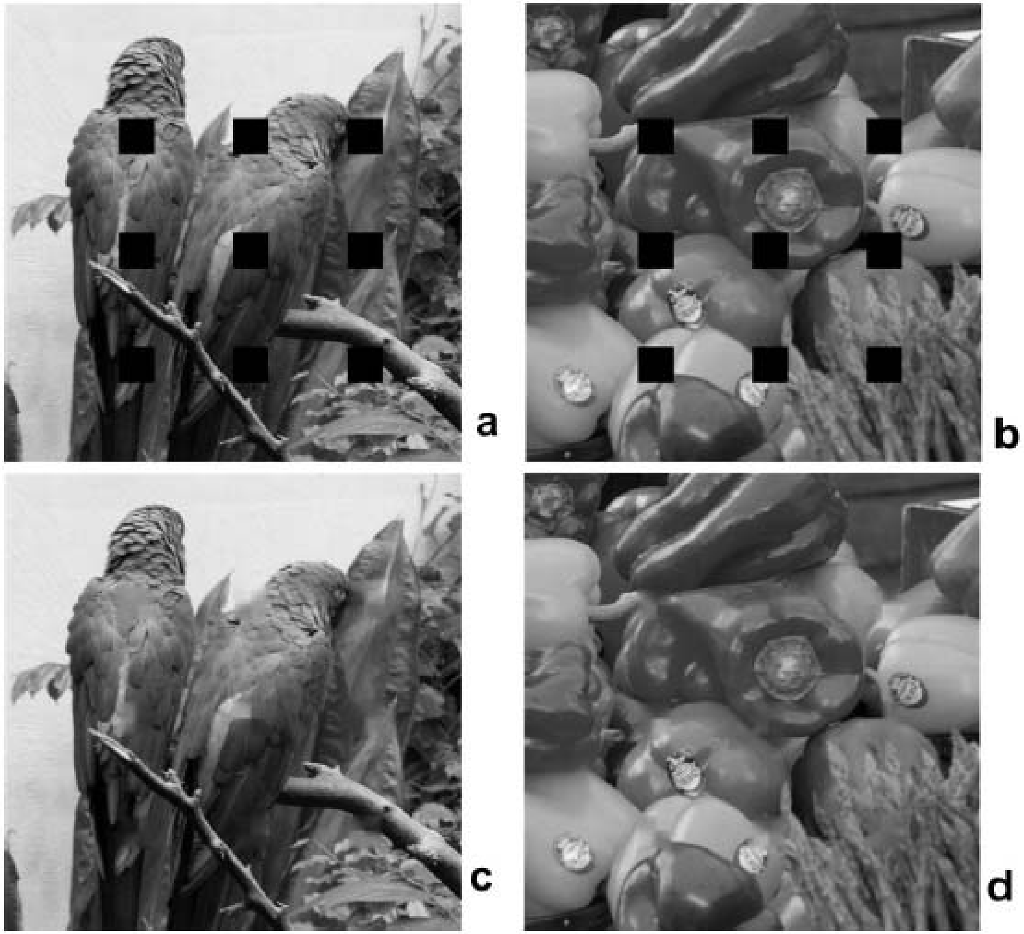


FIGURE 8. Examples of two images from the validation set with the block pattern of missing pixels: a) and b). Inpainting of two degraded images using ICA learned basis: c) and d).

Recovered missing regions are blurry but that is a known effect when larger missing regions are being inpainted.

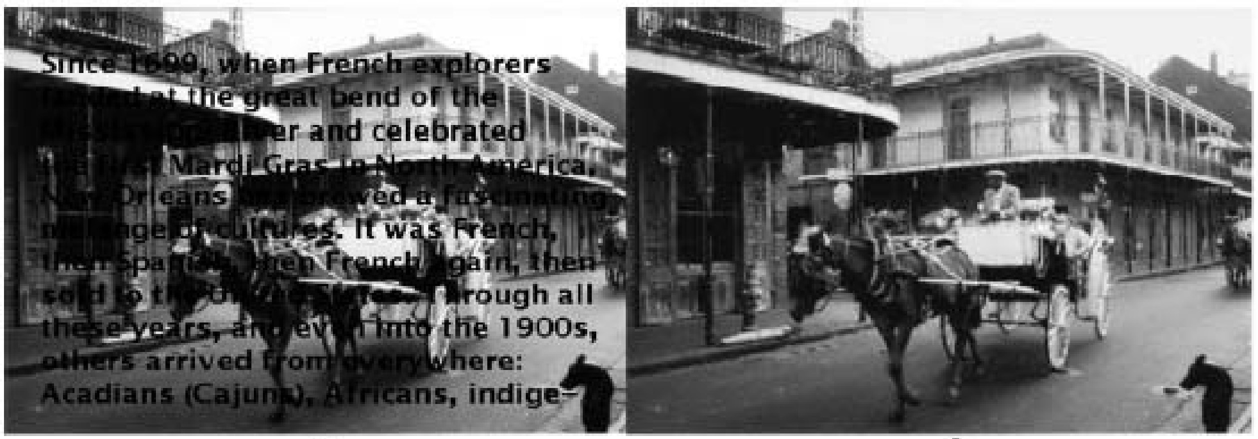
Experiments: inpainting missing pattern with a line structure

FIGURE 11. a) The Girls image. b) Inpainting using ICA learned basis.

Inpainting in ICA learned dictionary on this example is not as good as in dictionary learned by FoE: S. Roth, M. J. Black, Fields of experts, *Int. J. Computer Vision* 82 (2009) 205-229.

However, results are decent considering striking difference in computational complexity: **less than a minute for ICA/SL0 combination vs. 5 hours for FoE.**

Experiments: inpainting missing pattern with a “text” structure



a b

FIGURE 12. Inpainting for text removal. a) Image with text. b) Inpainting using ICA learned basis.

For this example the images and corresponding mask of missing pixels were taken from: http://www.dtic.upf.edu/~mbertalmio/restoration0.html.

Experiments: denoising 5% and 20% dense salt noise

5% of corrupted pixels

20% of corrupted pixels

2D 5x5 myriad filter

Inpainting in ICA basis

Experiments: denoising 5% and 20% dense salt noise

TABLE 9. Denoising results in terms of the SSIM metric when 5 percent of pixels were corrupted by impulsive noise.

	ICA	Myriad filtering
Fig. 6a	0.998	0.873
Fig. 6b	0.983	0.93
Fig. 6c	0.986	0.865
Fig. 6d	0.999	0.909
Fig. 6e	0.871	0.854
Fig. 6f	0.998	0.771
Mean	**0.973**	**0.867**

TABLE 10. Denoising results in terms of SSIM metric when 20 percent of pixels were corrupted by impulsive noise.

	ICA	Myriad filtering
Fig. 6a	0.991	0.643
Fig. 6b	0.977	0.691
Fig. 6c	0.974	0.569
Fig. 6d	0.997	0.674
Fig. 6e	0.916	0.743
Fig. 6f	0.993	0.287
Mean	**0.975**	**0.601**

THANK YOU !!!!!!!!