Adaptive Sparse Representations and their Applications

Saiprasad Ravishankar

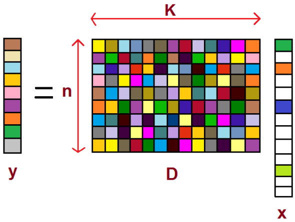
Department of Electrical and Computer Engineering
and Coordinated Science Laborarory
University of Illinois at Urbana-Champaign

January 29, 2015

Introduction to Sparse Signal Models

Synthesis Model (SM) for Sparse Representation

- Given a signal $y \in \mathbb{R}^n$, and dictionary $D \in \mathbb{R}^{n \times K}$, we assume $y = Dx$ with $\|x\|_0 \ll K$.



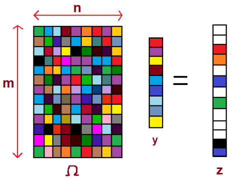
- Real world signals modeled as $y = Dx + e$, e is deviation term.
- Given D, sparsity level s, the *synthesis sparse coding* problem is

$$\hat{x} = \arg\min_{x} \|y - Dx\|_2^2 \;\; s.t. \;\; \|x\|_0 \leq s$$

- This problem is NP-hard.
- Greedy and ℓ_1-relaxation algorithms can be computationally expensive.

Analysis Model (AM) for Sparse Representation

- (Strict) AM : Given a signal $y \in \mathbb{R}^n$, and analysis dictionary $\Omega \in \mathbb{R}^{m \times n}$, $\|\Omega y\|_0 \ll m$.



- Noisy Signal Analysis Model (NSAM) : $y = q + e$, $\Omega q = z$ sparse.
- Given Ω, *co-sparsity level* t, the *analysis sparse coding* problem is

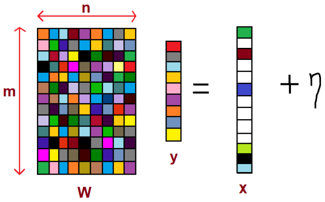
$$\hat{q} = \arg\min_{q} \|y - q\|_2^2 \;\; s.t. \;\; \|\Omega q\|_0 \le m - t$$

- This problem is NP-hard.
- Greedy[1] and ℓ_1-relaxation[2] algorithms are computationally expensive.

[1] [Rubinstein et al. '12] , [2] [Yaghoobi et al. '12].

Transform Model (TM) for Sparse Representation

- Given a signal $y \in \mathbb{R}^n$, and transform $W \in \mathbb{R}^{m \times n}$, we model $Wy = x + \eta$ with $\|x\|_0 \ll m$ and η - error term.



- Natural signals are approximately sparse in Wavelets, DCT.
- Given W, and sparsity s, *transform sparse coding* is

$$\hat{x} = \arg\min_{x} \|Wy - x\|_2^2 \;\; s.t. \;\; \|x\|_0 \leq s$$

- $\hat{x} = H_s(Wy)$ computed by thresholding Wy to the s largest magnitude elements. **Sparse coding is cheap!** Signal recovered as $W^{\dagger}\hat{x}$.
- Sparsifying transforms exploited for compression (JPEG2000), etc.

Learning Synthesis and Analysis Dictionaries

- Learning formulations - typically non-convex and NP-hard.
- Approximate algorithms for Synthesis Learning: MOD[3], K-SVD[4], online dictionary learning[5], etc.
- Heuristics for Analysis Learning:
  - (Strict) Analysis: Sequential Minimal Eigenvalues[6], AOL[7].
  - Noisy Analysis: Analysis K-SVD[8], NAAOL[9], GOAL[10] .
- Algorithms typically computationally expensive.
- Algorithms may not converge.

[3] [Engan et al. '99] , [4] [Aharon et al. '06], [5] [Mairal et al. '09], [6] [Ophir et al. '11], [7] [Yaghoobi et al. '11], [8] [Rubinstein et al. '12], [9] [Yaghoobi et al. '12], [10] [Hawe et al. '13].

Key Topic of Talk: Sparsifying Transform Learning

- **Square Transform Models**
  - Unstructured transform learning [IEEE TSP, 2013 & 2015]
  - Doubly sparse transform learning [IEEE TIP, 2013]
  - Online learning for Big Data [IEEE JSTSP, 2015]
  - Convex formulations for transform learning [ICASSP, 2014]
- **Overcomplete Transform Models**
  - Unstructured overcomplete transform learning [ICASSP, 2013]
  - Learning structured overcomplete transforms with block cosparsity (OCTOBOS) [IJCV, 2014]
- **Applications:** Sparse representation, Image & Video denoising, Classification, Blind compressed sensing (BCS) for imaging.

Unstructured Square Transform Learning

Square Transform Learning Formulation

$$(\text{P1}) \quad \min_{W,X} \overbrace{\|WY - X\|_F^2}^{\text{Sparsification Error}} + \lambda \overbrace{\left(\xi \|W\|_F^2 - \log|\det W|\right)}^{\text{Regularizer } \triangleq v(W)}$$
$$s.t. \quad \|X_i\|_0 \leq s \;\; \forall\, i$$

- $Y = [\,Y_1 \,|\, Y_2 \,|\, ... \,|\, Y_N\,] \in \mathbb{R}^{n \times N}$: matrix of training signals.
- $X = [\,X_1 \,|\, X_2 \,|\, ... \,|\, X_N\,] \in \mathbb{R}^{n \times N}$: matrix of sparse codes of Y_i.
- Sparsification error - measures deviation of data in transform domain from perfect sparsity.
- $\lambda,\, \xi > 0$. The log $|\det W|$ restricts solution to full rank transforms, and avoids repeated rows.
- $\|W\|_F^2$ keeps objective function bounded from below.
- (P1) is non-convex.

Properties of Formulation

$$
\text{(P1)} \quad \min_{W,X} \|WY - X\|_F^2 + \lambda \left(\xi \|W\|_F^2 - \log|\det W| \right)
$$
$$
s.t. \quad \|X_i\|_0 \leq s \;\; \forall \; i
$$

- (P1) attains lower bound of objective if and only if $\exists$ $(\hat{W}, \hat{X})$ with $\hat{X}$ sparse such that $\hat{W}Y = \hat{X}$, and the condition number $\kappa(\hat{W}) = 1$.
- (P1) favors both a low sparsification error and good conditioning.
- Minimizing the $\lambda \left(\xi \|W\|_F^2 - \log|\det W| \right)$ penalty encourages reduction of condition number.
- λ enables complete control over κ. The solution to (P1) is perfectly conditioned ($\kappa = 1$) as $\lambda \to \infty$.
- If w_i is the i^{th} row of W, then $\max_{i \neq j} \left| \frac{\|w_i\| - \|w_j\|}{\|w_i\|} \right| \leq \kappa(W) - 1$.

Algorithm with Iterative Transform Update

- (P1) solved by alternating between updating X and W.

- **Sparse Coding Step** solves for X with fixed W.

$$\min_{X} \|WY - X\|_F^2 \;\; s.t. \;\; \|X_i\|_0 \leq s \;\; \forall \; i \tag{1}$$

  - **Easy** problem: Solution $\hat{X}$ computed exactly by zeroing out all but the s largest magnitude coefficients in each column of WY.

- **Transform Update Step** solves for W with fixed X.

$$\min_{W} \|WY - X\|_F^2 + \lambda \left(\xi \|W\|_F^2 - \log |\det W| \right) \tag{2}$$

  - Solved using Non-linear Conjugate Gradients (NLCG)[11].

[11] [Ravishankar & Bresler, IEEE TSP, 2013].

Exact Transform Update

- **Transform Update Step**:

$$\min_{W} \|WY - X\|_F^2 + \lambda\left(\xi\,\|W\|_F^2 - \log|\det W|\right) \tag{3}$$

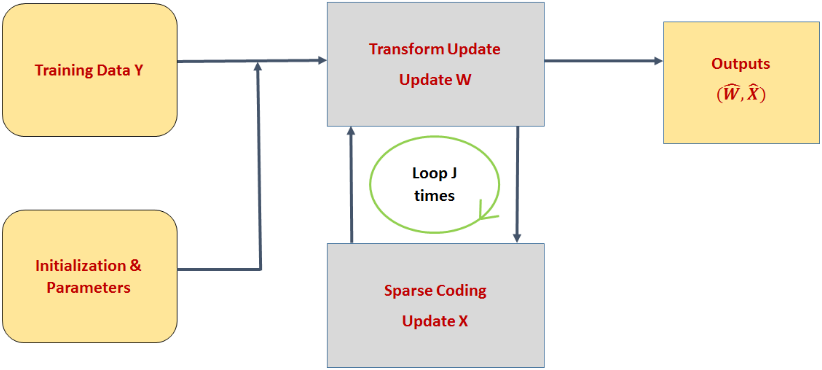
- **Closed-form solution:**

$$\hat{W} = 0.5U\left(\Sigma + \left(\Sigma^2 + 2\lambda I_n\right)^{\frac{1}{2}}\right)Q^T L^{-1} \tag{4}$$

where $YY^T + \lambda\xi I_n = LL^T$, and $L^{-1}YX^T$ has a full singular value decomposition (SVD) of $Q\Sigma U^T$.

- The solution is invariant to the specific choice of square root L.
- It is unique if and only if YX^T is non-singular.

Algorithm A1 for Square Transform Learning



Orthonormal Transform Learning Limit

Proposition 1

For $\xi = 0.5$, as $\lambda \to \infty$, the sparse coding and transform update solutions in (P1) coincide with the solutions obtained by employing alternating minimization on

$$\min_{W,X} \|WY - X\|_F^2 \;\; s.t. \;\; W^T W = I, \; \|X_i\|_0 \leq s \;\; \forall \; i. \tag{5}$$

Specifically, the sparse coding step for Problem (5) involves

$$\min_{X} \|WY - X\|_F^2 \;\; s.t. \;\; \|X_i\|_0 \leq s \;\; \forall \; i \tag{6}$$

and the solution is $\hat{X}_i = H_s(WY_i) \; \forall \, i$. Transform update involves

$$\max_{W} tr\left(WYX^T\right) \;\; s.t. \;\; W^T W = I \tag{7}$$

Let $YX^T = U\Sigma V^T$ be a full SVD. Then, an optimal $\hat{W}$ in (7) is VU^T.

Convergence Guarantees - Notations

- Define the barrier function

$$\psi(X) = \begin{cases} 0, & \|X_i\|_0 \leq s, \ \forall\, i \\ +\infty, & \text{else} \end{cases}$$

- **(P1) is equivalent to the problem of minimizing $g(W, X)$.**

$$g(W, X) \triangleq \|WY - X\|_F^2 + \lambda\xi \|W\|_F^2 - \lambda \log |\det W| + \psi(X) \quad (8)$$

- For $h \in \mathbb{R}^p$, $\phi_j(h)$ is the magnitude of the j^{th} largest element (magnitude-wise) of h.

- For $B \in \mathbb{C}^{p \times q}$, $\|B\|_\infty \triangleq \max_{i,j} |B_{ij}|$.

Convergence Guarantees

Theorem 1

For the sequence $\left\{W^k, X^k\right\}$ generated by Algorithm A1 with initial (W^0, X^0), we have

- $\left\{g(W^k, X^k)\right\}$ *converges to a finite value* $g^* = g^*\left(W^0, X^0\right)$.
- $\left\{W^k, X^k\right\}$ *is bounded, and any specific accumulation point* (W, X) *is a fixed point of Algorithm A1 satisfying*

$$g(W + dW, X + \Delta X) \geq g(W, X) = g^* \tag{9}$$

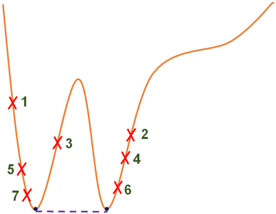
The condition holds for all sufficiently small $dW \in \mathbb{R}^{n \times n}$ *satisfying* $\|dW\|_F \leq \epsilon$ *for some* $\epsilon = \epsilon(W) > 0$, *and all* $\Delta X \in R1 \cup R2$

R1. *The half-space* $tr\left\{(WY - X)\Delta X^T\right\} \leq 0$.

R2. *The local region defined by* $\|\Delta X\|_\infty < \min_i \left\{\phi_s(WY_i) : \|WY_i\|_0 > s\right\}$.

Furthermore, if we have $\|WY_i\|_0 \leq s \,\forall\, i$, *then* ΔX *can be arbitrary.*

Global Convergence Guarantees



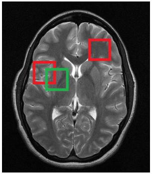
Corollary 1

For each initialization of Algorithm A1, the objective converges to a local minimum, and the iterates converge to an equivalence class of local minimizers.

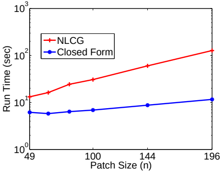
Corollary 2

Algorithm A1 is globally convergent (i.e., from any Initialization) to the set of local minimizers of the non-convex objective $g(W, X)$.

Computational Advantages



Patches of image

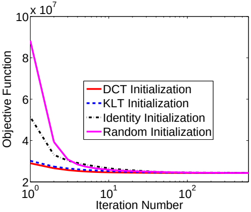


Run times

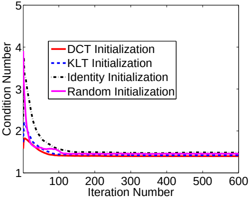
- Cost per iteration of proposed algorithms: $O(Nn^2)$ for N training signals and $W \in \mathbb{R}^{n \times n}$.
- Synthesis/Analysis K-SVD cost per iteration : $O(Nn^3)$. Cost dominated by sparse coding.
- For images, this is a reduction of computations in the order by n, corresponding to $\sqrt{n} \times \sqrt{n}$ patches.
- Closed-form solution for transform update also provides speedup of about J over NLCG, where J is the number of NLCG steps.

Convergence for (P1) with Various Initializations

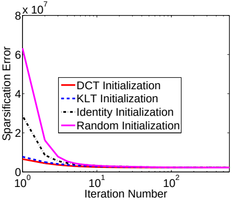
Barbara - 8×8 patches



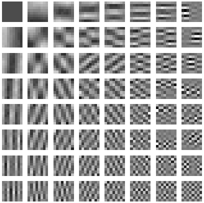
Objective Function



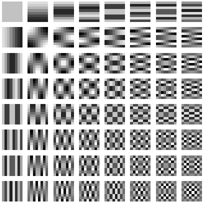
$\kappa(W)$



Sparsification Error $(s = 11)$



Learnt W - DCT Init



2D DCT

Learnt transforms are better than analytical transforms

- Normalized Sparsification Error (NSE) measures the fraction of energy lost in sparse fitting with sparse code X.

$$\mathrm{NSE} = \frac{\|\mathrm{WY}-\mathrm{X}\|_\mathrm{F}^2}{\|\mathrm{WY}\|_\mathrm{F}^2}\,,\ \mathrm{NSE(W)} \approx 4.4\%\,,\ \mathrm{NSE(DCT)} = 6.8\%.$$

- Recovery PSNR (rPSNR) measures the error in recovering image as $\hat{Y} = W^{-1}X$.

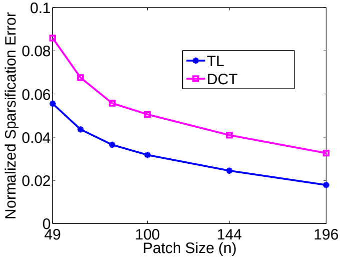
$$\mathrm{rPSNR} = \frac{255\sqrt{P}}{\|Y - W^{-1}X\|_F}$$

  P is $\#$ of image pixels.

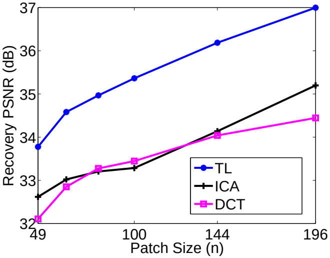
- rPSNRs for the learnt W about 1.7 dB better than for DCT.

- Varying λ allows trade-off between NSE and $\kappa(W)$. rPSNR best at intermediate κ.

Comparison of Algorithms in Image Representation



NSE vs. n



rPSNR vs. n

- Transform learning (TL) provides better sparsification & recovery than DCT.
- Adapted well-conditioned transforms perform better (up to 0.3 dB better recovery) than adapted orthonormal transforms.
- Adapted transforms outperform Independent Component Analysis (ICA).

Application: Image Denoising

$$\min_{\{x_j\}} \overbrace{u(x_1, x_2, .., x_n)}^{\text{Regularizer}} + \tau \overbrace{\sum_{j=1}^{M} \|R_j\, y - x_j\|_2^2}^{\text{Data Fidelity}}$$

- Estimate image $x \in \mathbb{R}^P$ from its noisy measurement $y = x + h$.
- $R_j \in \mathbb{R}^{n \times P}$ extracts patches. $R_j\, y \approx$ noiseless x_j.
- $u(x_1, x_2, .., x_n)$ is a regularizer $\Rightarrow$ **regularized inverse problem.**
- $\tau \propto \frac{1}{\sigma}$ with σ being the noise level.
- Denoised x obtained by averaging x_j's at their 2D locations.

Image Denoising with Transform Learning Regularizer

$$(\mathrm{P2}) \quad \min_{W,\{x_j\},\{\alpha_j\}} \overbrace{\sum_{j=1}^{M} \|Wx_j - \alpha_j\|_2^2}^{\text{Sparsification Error}} + \lambda \overbrace{v(W)}^{\text{Regularizer}} + \tau \overbrace{\sum_{j=1}^{M} \|R_j\, y - x_j\|_2^2}^{\text{Data Fidelity}}$$

$$s.t. \quad \|\alpha_j\|_0 \leq s_j \ \ \forall\, j$$

- $R_j \in \mathbb{R}^{n \times P}$ extracts patches. $R_j\, y \approx$ noiseless x_j, $Wx_j \approx \alpha_j$.
- $\alpha_j \in \mathbb{R}^n$ is transform sparse code of x_j.
- (P2) is solved by an efficient alternating scheme that uses **closed-form updates**, and s_j are found adaptively.
- Denoised x obtained by averaging x_j's at their 2D locations.

Image Denoising Example

Noisy Image
PSNR $= 24.60$ dB

64×64 W ($\kappa = 1.3$)
PSNR $= 31.66$ dB

64×256 Synthesis D
PSNR $= 31.50$ dB

- Closed-form updates-based denoising is better and 17x faster than overcomplete K-SVD denoising.
- Square K-SVD (PSNR $= 31.14$ dB) denoises worse, and is slower.
- Our denoising PSNR increases with patch size n, while still providing speedups over K-SVD of lower n.

Summary

- We proposed formulations for learning square sparsifying transforms.
- Proposed alternating algorithms
  - involve efficient optimal updates
  - converge globally to the set of local minimizers of objective
  - low computational cost
  - encourage well-conditioning
- Adapted transforms provide better representations than analytical ones.
- Adaptive transforms denoise comparably or better than learnt overcomplete synthesis dictionaries, but are faster.

Blind Compressed Sensing for Imaging

Compressed Sensing (CS)

- CS enables accurate recovery of images from fewer measurements than number of unknowns or Nyquist sampling
  - Sparsity in transform domain or dictionary
  - Acquisition incoherent with transform
  - **Reconstruction problem is hard**
- Reconstruction problem (NP-hard) -

$$\min_{x} \|Ax - y\|_2^2 + \lambda \|\Psi x\|_0 \tag{10}$$

- $x \in \mathbb{C}^P$: signal/image as vector, $y \in \mathbb{C}^m$: measurements.
- $A \in \mathbb{C}^{m \times P}$: sensing matrix $(m < P)$, $\Psi \in \mathbb{C}^{T \times P}$: given transform.
- ℓ_1 relaxation of sparsity penalty is used to generate convex problem.

Application: Compressed Sensing MRI (CSMRI)

- Data - samples in k-space of spatial Fourier transform of object, acquired sequentially.
- Acquisition rate limited by MR physics, physiological constraints on RF energy deposition.
- CSMRI accelerates the data acquisition process in MRI.
- CSMRI with non-adaptive transforms or dictionaries limited to 2.5-3 fold undersampling [Ma et al. '08].
- Two directions to improve CSMRI -
  - **better or adaptive sparse modeling**
  - better choice of sampling pattern (F_u) [EMBC, 2011]

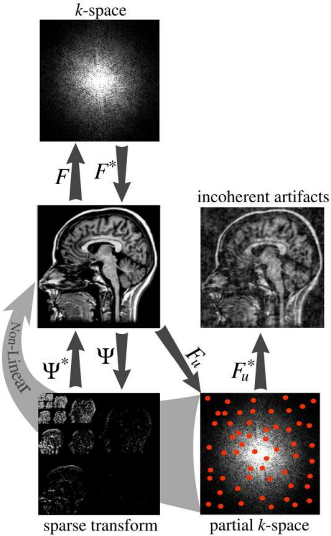


Fig. from Lustig et al. '07

Synthesis-based Blind Compressed Sensing (BCS)

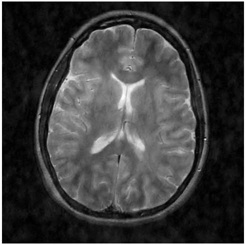
$$\text{(P3)} \quad \min_{x,D,B} \overbrace{\sum_{j=1}^{N} \|R_j x - D b_j\|_2^2}^{\text{Sparse Fitting Regularizer}} + \nu \overbrace{\|Ax - y\|_2^2}^{\text{Data Fidelity}}$$

$$s.t. \quad \|d_k\|_2 = 1 \; \forall \; k, \quad \|b_j\|_0 \leq s \;\; \forall \; j.$$

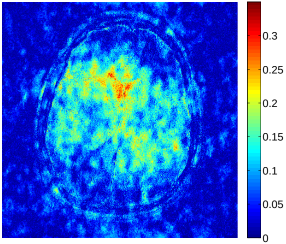
- $B \in \mathbb{C}^{n \times N}$: matrix that has the sparse codes b_j as its columns.
- (P3) learns $D \in \mathbb{C}^{n \times K}$, and reconstructs x, from only undersampled $y \Rightarrow$ **dictionary adaptive to underlying image.**
- DLMRI[12] solves (P3) for MRI and works better than non-adaptive CS methods like Wavelets + TV based LDP [Lustig, Donoho & Pauly '07].

12 [Ravishankar & Bresler '11]

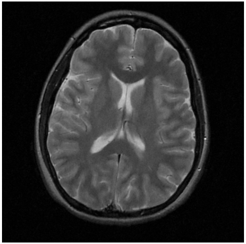
2D Random Sampling - 6 fold undersampling



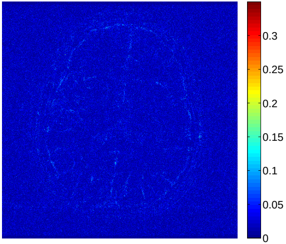
LDP reconstruction (22 dB)



LDP error magnitude



DLMRI reconstruction (32 dB)



DLMRI error magnitude

Drawbacks of Synthesis Dictionary-based BCS

$$(\text{P3}) \quad \min_{x,D,B} \; \overbrace{\sum_{j=1}^{N} \|R_j x - Db_j\|_2^2}^{\text{Sparse Fitting Regularizer}} + \nu \, \overbrace{\|Ax - y\|_2^2}^{\text{Data Fidelity}}$$

$$s.t. \quad \|d_k\|_2 = 1 \;\forall\; k, \quad \|b_j\|_0 \leq s \;\; \forall\; j.$$

- (P3) is NP-hard, non-convex even if ℓ_0-quasinorm relaxed to ℓ_1.
- Synthesis BCS algorithms have no guarantees and are computationally expensive.

Transform-based Blind Compressed Sensing (BCS)

$$(\mathrm{P4}) \quad \min_{x,W,B} \overbrace{\sum_{j=1}^{N} \|WR_j x - b_j\|_2^2}^{\text{Sparsification Error}} + \nu \overbrace{\|Ax - y\|_2^2}^{\text{Data Fidelity}} + \lambda \overbrace{v(W)}^{\text{Regularizer}}$$

$$s.t. \quad \sum_{j=1}^{N} \|b_j\|_0 \leq s, \quad \|x\|_2 \leq C.$$

- (P4) learns $W \in \mathbb{C}^{n\times n}$, and reconstructs x, from only undersampled y $\Rightarrow$ **transform adaptive to underlying image.**
- $v(W) \triangleq -\log|\det W| + 0.5\,\|W\|_F^2$ controls scaling and κ of W.
- We set $\lambda = \lambda_0 N$, with $\lambda_0 > 0$ a constant.
- $\|x\|_2 \leq C$ is an energy/range constraint. $C > 0$.

Transform BCS Properties

Proposition 2

Let $x \in \mathbb{C}^p$, and let $y = Ax$ with $A \in \mathbb{C}^{m \times p}$. Suppose

- $\|x\|_2 \leq C$
- *$W \in \mathbb{C}^{n \times n}$ is a unitary transform*
- $\sum_{j=1}^{N} \|WR_j x\|_0 \leq s$

Further, let B denote the matrix that has $WR_j x$ as its columns. Then, (x, W, B) is a global minimizer of Problem (P4), i.e., it is identifiable by solving (P4).

- Conditions for uniqueness of solution to (P4) an open question.
- Given minimizer (x, W, B) of (P4), $(x, \Theta W, \Theta B)$ is another **equivalent minimizer** $\forall\, \Theta$ s.t. $\Theta^H \Theta = I$, $\sum_j \|\Theta b_j\|_0 \leq s$.

Alternative Transform BCS Formulations

$$
\begin{aligned}
(\mathrm{P5}) \quad & \min_{x,W,B} \sum_{j=1}^{N} \|WR_j x - b_j\|_2^2 + \nu \|Ax - y\|_2^2 \\
& s.t. \;\; W^H W = I, \; \sum_{j=1}^{N} \|b_j\|_0 \leq s, \;\; \|x\|_2 \leq C.
\end{aligned}
$$

- **(P5) is also a unitary synthesis dictionary-based BCS problem, with W^H the synthesis dictionary.**

$$
\begin{aligned}
(\mathrm{P6}) \quad & \min_{x,W,B} \sum_{j=1}^{N} \|WR_j x - b_j\|_2^2 + \nu \|Ax - y\|_2^2 + \lambda\, v(W) + \eta^2 \sum_{j=1}^{N} \|b_j\|_0 \\
& s.t. \;\; \|x\|_2 \leq C.
\end{aligned}
$$

Block Coordinate Descent (BCD) Algorithm for (P4)

- (P4) solved by alternating between updating W, B, and x.
- Alternate a few times between the W and B updates, before performing an image update.

- **Sparse Coding Step** solves (P4) for B with fixed x, W.

$$\min_{B} \sum_{j=1}^{N} \|WR_j x - b_j\|_2^2 \quad s.t. \quad \sum_{j=1}^{N} \|b_j\|_0 \leq s. \tag{11}$$

  - **Cheap Solution:** Let $Z \in \mathbb{C}^{n \times N}$ be the matrix with $WR_j x$ as its columns. Solution $\hat{B} = H_s(Z)$ computed exactly by zeroing out all but the s largest magnitude coefficients in Z.

BCD Algorithm for (P4)

- **Transform Update Step** solves (P4) for W with fixed x, B.

$$\min_{W} \sum_{j=1}^{N} \|WR_j x - b_j\|_2^2 + 0.5\lambda \|W\|_F^2 - \lambda \log |\det W| \tag{12}$$

  - Let $X \in \mathbb{C}^{n \times N}$ be the matrix with $R_j x$ as its columns.
  - **Closed-form solution:**

$$\hat{W} = 0.5R \left(\Sigma + \left(\Sigma^2 + 2\lambda I \right)^{\frac{1}{2}} \right) V^H L^{-1} \tag{13}$$

  where $XX^H + 0.5\lambda I = LL^H$, and $L^{-1}XB^H$ has a full SVD of $V\Sigma R^H$.

  - Solution is unique if and only if XB^H is non-singular.

BCD Algorithm for (P4)

- **Image Update Step** solves (P4) for x with fixed W, B.

$$\min_{x} \sum_{j=1}^{N} \|WR_j x - b_j\|_2^2 + \nu \|Ax - y\|_2^2 \;\; s.t. \;\; \|x\|_2 \leq C. \tag{14}$$

  - Least squares problem with ℓ_2 norm constraint.
  - Solution is unique as long as the set of overlapping patches cover all image pixels.
  - **Solve Least squares Lagrangian formulation:**

$$\min_{x} \sum_{j=1}^{N} \|WR_j x - b_j\|_2^2 + \nu \|Ax - y\|_2^2 + \mu \left(\|x\|_2^2 - C \right) \tag{15}$$

    - The optimal multiplier $\hat{\mu} \in \mathbb{R}^+$ is the smallest real such that $\|\hat{x}\|_2 \leq C$. $\hat{\mu}$ can be found cheaply.

BCS Convergence Guarantees - Notations

- Define the barrier function $\psi(B)$ as

$$\psi(B) = \begin{cases} 0, & \sum_{j=1}^{N} \|b_j\|_0 \leq s \\ +\infty, & \text{else} \end{cases}$$

- $\chi(x)$ is the barrier function corresponding to $\|x\|_2 \leq C$.
- (P4) is equivalent to the problem of minimizing $g(W, B, x) = \sum_{j=1}^{N} \|WR_j x - b_j\|_2^2 + \nu \|Ax - y\|_2^2 + \lambda\, v(W) + \psi(B) + \chi(x)$.
- For $H \in \mathbb{C}^{p \times q}$, $\rho_j(H)$ is the magnitude of the j^{th} largest element (magnitude-wise) of H.
- $X \in \mathbb{C}^{n \times N}$ denotes a matrix with $R_j x$, $1 \leq j \leq N$, as its columns.

Transform BCS Convergence Guarantees

Theorem 2

For the sequence $\{W^t, B^t, x^t\}$ generated by the BCD Algorithm with initial (W^0, B^0, x^0), we have

- $\{g\,(W^t, B^t, x^t)\}$ *converges to a finite* $g^* = g^*(W^0, B^0, x^0)$.
- $\{W^t, B^t, x^t\}$ *is bounded, and all its accumulation points are equivalent, i.e., they achieve the same value* g^* *of the objective.*
- *The sequence* $\{a^t\}$ *with* $a^t \triangleq \left\|x^t - x^{t-1}\right\|_2$, *converges to zero.*
- *Every accumulation point* (W, B, x) *is a critical point of* g *satisfying the following partial global optimality conditions*

$$x \in \underset{\tilde{x}}{\arg\min}\ g\,(W, B, \tilde{x}) \tag{16}$$

$$W \in \underset{\tilde{W}}{\arg\min}\ g\left(\tilde{W}, B, x\right),\ B \in \underset{\tilde{B}}{\arg\min}\ g\left(W, \tilde{B}, x\right) \tag{17}$$

Transform BCS Convergence Guarantees

Theorem 3

Each accumulation point (W, B, x) of $\{W^t, B^t, x^t\}$ also satisfies the following partial local optimality conditions

$$g(W + dW, B + \Delta B, x) \geq g(W, B, x) = g^* \tag{18}$$

$$g(W, B + \Delta B, x + \tilde{\Delta} x) \geq g(W, B, x) = g^* \tag{19}$$

The conditions each hold for all $\tilde{\Delta} x \in \mathbb{C}^p$, and all $dW \in \mathbb{C}^{n \times n}$ satisfying $\|dW\|_F \leq \epsilon$ for some $\epsilon = \epsilon(W) > 0$, and all $\Delta B \in \mathbb{C}^{n \times N}$ in R1 $\cup$ R2

R1. *The half-space $Re\left(tr\left\{(WX - B)\Delta B^H\right\}\right) \leq 0$.*

R2. *The local region defined by $\|\Delta B\|_\infty < \rho_s(WX)$.*

Furthermore, if $\|WX\|_0 \leq s$, then ΔB can be arbitrary.

Global Convergence Guarantees

Corollary 3

For each initialization, the iterate sequence in the BCD algorithm converges to an equivalence class of critical points, that are also partial global/local minimizers.

Corollary 4

The BCD algorithm is globally convergent (i.e., from any Initialization) to a subset of the set of critical points of the non-convex BCS objective $g(W, B, x)$, *that includes all* (W, B, x) *that are at least partial global minimizers of* g *with respect to each of* W, B, *and* x, *and partial local minimizers of* g *with respect to* (W, B), *and* (B, x).

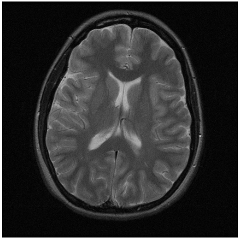
Computational Advantages of Transform BCS

- Cost per iteration of transform BCS: $O(n^2 NL)$
  - N overlapping patches of size $\sqrt{n} \times \sqrt{n}$, $W \in \mathbb{C}^{n \times n}$.
  - L : $\#$ inner alternations between transform update & sparse coding.
- Cost per iteration of Synthesis BCS method DLMRI[13]: $O(n^3 NJ)$.
  - $D \in \mathbb{C}^{n \times K}$, $K \propto n$, sparsity $s \propto n$.
  - J : $\#$ of inner iterations of dictionary learning using K-SVD.
- Transform BCS much cheaper as n increases $\Rightarrow$ 3D or 4D imaging.

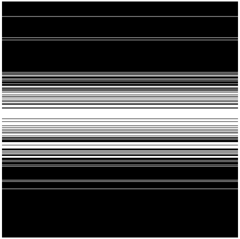
[13] [Ravishankar & Bresler '11]

Application: Transform Learning Based CSMRI (TLMRI)

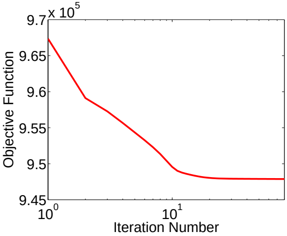
TLMRI Convergence - 4x Undersampling ($s = 3.4\%$)



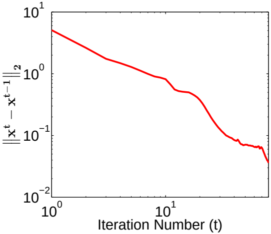
Reference [14]



Sampling mask



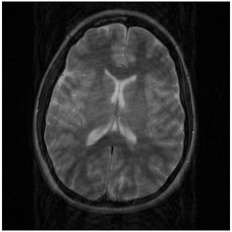
Objective



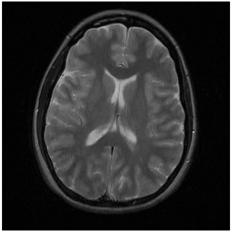
$\left\|x^t - x^{t-1}\right\|_2$ vs. t

[14] Data from Miki Lustig.

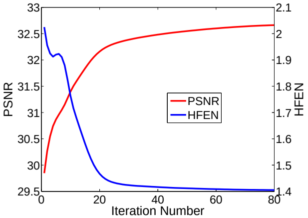
Convergence & Learning - 4x Undersampling ($s = 3.4\%$)



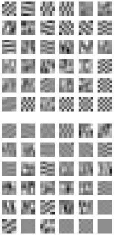
Zero-filling (28.94 dB)



TLMRI (32.66 dB)



PSNR and HFEN [15]



real (top), imaginary (bottom) parts of learnt 36×36 W

[15] [Ravishankar & Bresler '11]

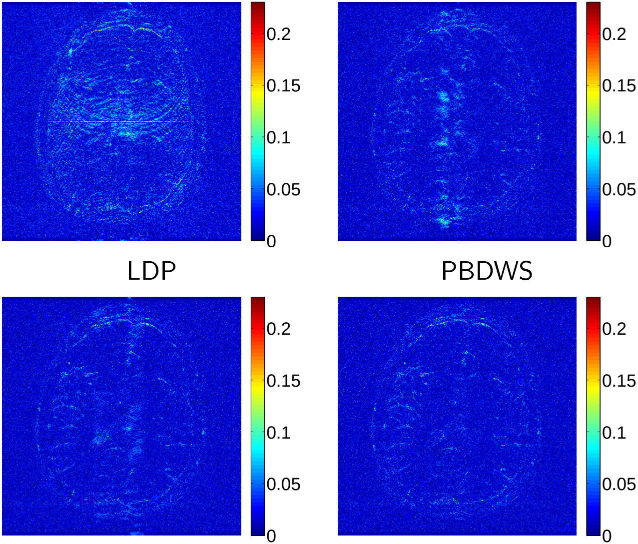
Comparison (PSNR & Runtime) to Recent Methods

Sampling Scheme	Undersampling	Zero-filling	LDP	PBDWS	DLMRI	TLMRI
2D Random	4x	25.3	30.32	32.64	32.91	**33.04**
	7x	25.3	27.34	31.31	31.46	**31.81**
Cartesian	4x	28.9	30.20	32.02	32.46	**32.64**
	7x	27.9	25.53	30.09	30.72	**31.04**
Avg. Runtime (s)			251	794	2051	**211**

- TLMRI is up to 5.5 dB better than LDP[16], that uses Wavelets + TV.
- TLMRI provides up to 1 dB improvement in PSNR over the PBDWS[17] method, that uses redundant Wavelets and trained patch-based geometric directions.
- It is up to 0.35 dB better than DLMRI[18], that learns 4x overcomplete dictionary.
- **TLMRI is 10x faster than DLMRI, and 4x faster than the PBDWS method.**

16 [Lustig et al. '07] 17 [Ning et al. '13] 18 [Ravishankar & Bresler '11]

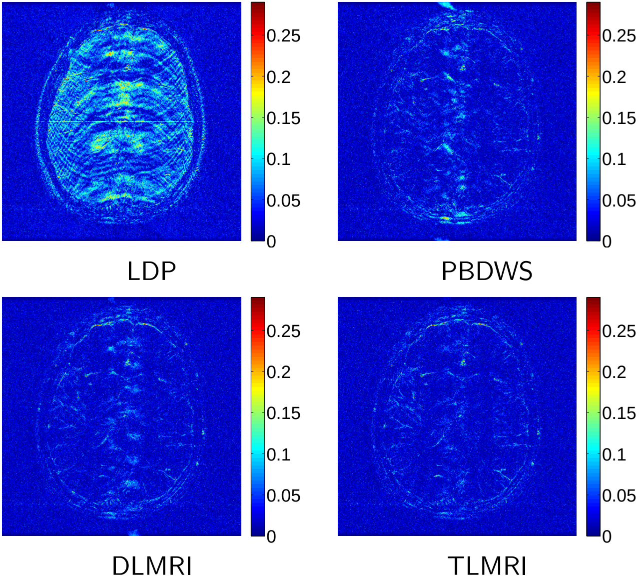
Reconstruction Errors - Cartesian 4x Undersampling



LDP PBDWS

DLMRI TLMRI

Reconstruction Errors - Cartesian 7x Undersampling



Summary

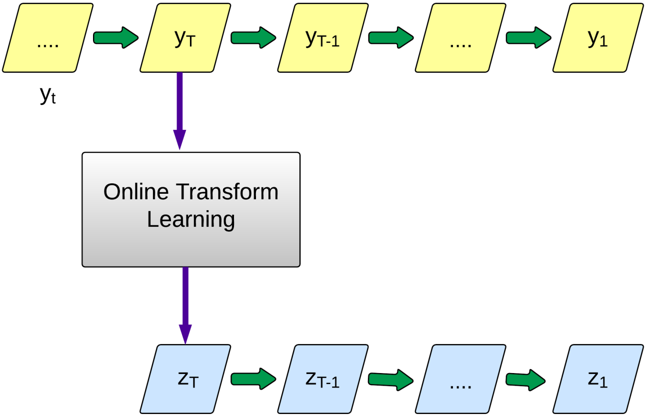
- We introduced a transform-based BCS framework
- Proposed BCS algorithms have a low computational cost.
- We provided novel convergence guarantees for the algorithms.
- For CSMRI, the proposed approach is better than leading image reconstruction methods, while being much faster.
- Future work: uniqueness of solution in BCS; Convergence to global minimizer.

Online Learning for Big Data

Why Online Transform Learning?

- **Prior work:** batch transform learning, where learning is done using all the training data simultaneously.
- Big data $\Rightarrow$ training set is very large $\Rightarrow$ batch learning computationally expensive in time and memory.
- In real-time applications, data arrives sequentially, and must be processed sequentially to limit latency.
- Online learning enables sequential adaptation of the transform (and sparse codes or signal estimates)
  - amenable to big data, and real-time applications.
  - involves cheap computations and modest memory requirements.

Online Transform Learning



z_t : Learnt Transform/Sparse Codes/Signal Estimates

Online Transform Learning Formulations

- For $t = 1, 2, 3, \ldots,$ solve

$$\text{(P7)} \quad \left\{\hat{W}_t, \hat{x}_t\right\} = \underset{W, x_t}{\arg\min} \frac{1}{t} \sum_{j=1}^{t} \left\{ \|Wy_j - x_j\|_2^2 + \lambda_j v(W) \right\}$$
$$s.t. \quad \|x_t\|_0 \leq s, \; x_j = \hat{x}_j, \; 1 \leq j \leq t-1.$$

- $\lambda_j = \lambda_0 \|y_j\|_2^2 \; \forall j$, with $\lambda_0 > 0$. $v(W) \triangleq \|W\|_F^2 - \log|\det W|$.
- λ_0 controls the condition number and scaling of learnt $\hat{W}_t$.
- At time t, estimate of $\{y_j\}$ obtained as $\left\{\hat{W}_t^{-1}\hat{x}_j\right\}$ (decompression).
- For non-stationary data, use forgetting factor $\rho \in [0, 1]$, to diminish the influence of old data.

$$\frac{1}{t} \sum_{j=1}^{t} \rho^{t-j} \left\{ \|Wy_j - x_j\|_2^2 + \lambda_j v(W) \right\} \tag{20}$$

Mini-Batch Transform Learning

- For $J = 1, 2, 3, ...,$ solve

$$\left\{\hat{W}_J, \hat{X}_J\right\} = \underset{W, X_J}{\arg\min} \frac{1}{JM} \sum_{j=1}^{J} \left\{ \|WY_j - X_j\|_F^2 + \Lambda_j v(W) \right\}$$

$$s.t. \quad \|x_{JM-M+i}\|_0 \leq s \;\; \forall\, i \in \{1, .., M\} \quad \text{(P8)}$$

- $Y_J = [\, y_{JM-M+1} \,|\, y_{JM-M+2} \,|\, ... \,|\, y_{JM} \,]$, with M : mini-batch size.
- $X_J = [\, x_{JM-M+1} \,|\, x_{JM-M+2} \,|\, ... \,|\, x_{JM} \,]$. $\Lambda_j = \lambda_0 \, \|Y_j\|_F^2$.
- Mini-batch learning
  - **can provide reductions in operation count over online learning.**
  - **increased latency and memory requirements.**

Online Adaptive Transform Denoising

$$\text{(P9)} \quad \min_{W,\, x_t} \frac{1}{t} \sum_{j=1}^{t} \left\{ \|Wy_j - x_j\|_2^2 + \lambda_j v(W) + \tau_j^2 \|x_j\|_0 \right\}$$

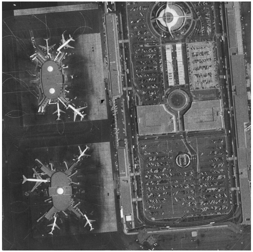
- Goal: Given $\{y_t\}$, with $y_t = z_t + h_t$, and $h_t \in \mathbb{R}^n$ the noise, find $z_t \; \forall \; t$.
- $\tau_j \propto \sigma$, with σ - noise level.
- Denoised signal is $\hat{z}_t = \hat{W}_t^{-1} \hat{x}_t$ – computed efficiently in our algorithm.
- (P9) can be used for denoising images, or image sequences:
    - overlapping patches of the noisy image(s) denoised sequentially.
    - image estimated by averaging denoised patches at 2D locations.

Image Denoising – PSNR (dB) and runtime (sec)

Images	σ	Noisy PSNR		Batch K-SVD	Batch TL	Mini-batch TL ($M = 64$)
Couple (512 × 512)	5	34.16	PSNR	37.28	37.33	**37.33**
			time	1250	92	**20**
	10	28.11	PSNR	33.51	33.62	**33.62**
			time	671	68	**19**
	20	22.11	PSNR	30.02	30.02	**30.03**
			time	190	61	**20**
Man (768 × 768)	5	34.15	PSNR	36.47	36.66	**36.75**
			time	1279	205	**45**
	10	28.13	PSNR	32.71	32.96	**33.00**
			time	701	130	**44**
	20	22.11	PSNR	29.40	**29.57**	29.52
			time	189	80	**41**

- Overlapping 8×8 patches are denoised sequentially with a forgetting factor. We observed better denoising with a forgetting factor.
- Mini-batch denoising is better and provides average speedup of $26.0\times$ and $3.4\times$ over the batch K-SVD and batch transform denoising schemes.

Big Image Denoising - Data


Airport (1024×1024)


Man (1024×1024)


Railway (2048×2048)


Campus ($3264 \times 3264 \times 3$)

Big Image Denoising – PSNR (dB) and runtime (sec)

Images	Methods	$\sigma = 20$ (22.11)	$\sigma = 50$ (14.15)	$\sigma = 100$ (8.13)	Run Times
Airport	DCT	28.79	24.65	21.00	23
	TL	**28.83**	**25.07**	**22.53**	28
Man	DCT	30.44	25.80	21.87	23
	TL	**30.64**	**26.62**	**23.88**	27
Railway	DCT	31.90	26.44	22.04	90
	TL	**32.42**	**27.58**	**24.35**	111
Campus	DCT	30.89	25.88	21.99	323
	TL	**33.10**	**27.47**	**23.24**	451

- Adaptive mini-batch denoising (TL) performs better than the DCT, without much loss in runtime.
- Results demonstrate the potential of our schemes for real-time denoising of large-scale data.

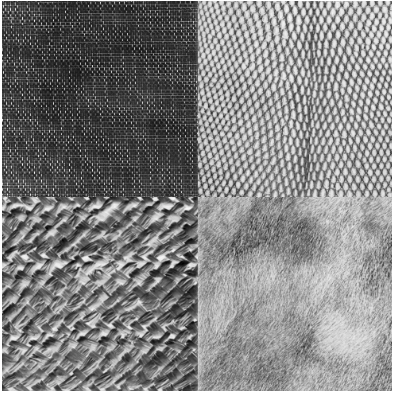
Summary

- We introduced an online sparsifying transform learning framework.
- Proposed methods are particularly useful for big data & real-time applications.
- Iterates converge to the set of stationary points of the expected transform learning cost [IEEE JSTSP, 2015].
- The online schemes perform well and are highly efficient for sparse representation & denoising.
- Ongoing work: video denoising, online blind compressed sensing.
  - Video denoising by online 3D transform learning provides 0.7 dB better denoising PSNR compared to the popular VBM4D.

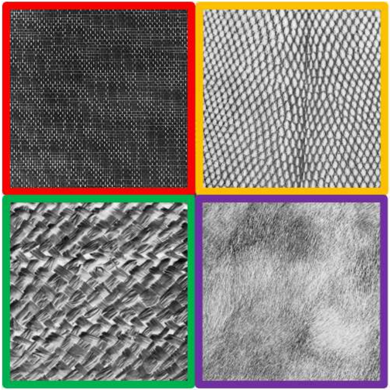
Union of Transforms or OCTOBOS

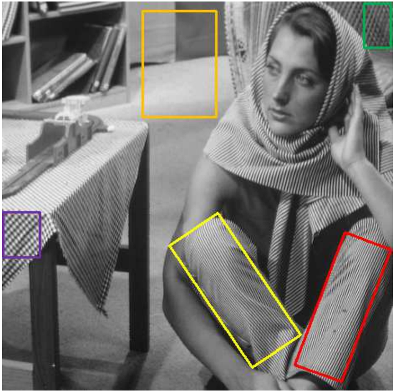
Why Union of Transforms (UOT)?

- Natural images typically have diverse textures.



- **Union of transforms: one for each class of textures or features.**





OCTOBOS Learning Formulation

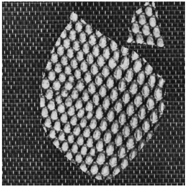
$$\text{(P12)} \quad \min_{\{W_k, X_i, C_k\}} \overbrace{\sum_{k=1}^{k} \sum_{i \in C_k} \|W_k Y_i - X_i\|_2^2}^{\text{Sparsification Error}} + \overbrace{\sum_{k=1}^{k} \lambda_k \left(\|W_k\|_F^2 - \log|\det W_k| \right)}^{\text{Regularizer} \triangleq \sum \lambda_k v(W_k)}$$

$$s.t. \quad \|X_i\|_0 \leq s \ \forall\, i, \quad \{C_k\}_{k=1}^{K} \in G$$

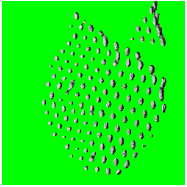
- G is the set of all partitions of $[1:N]$ into K disjoint subsets $\{C_k\}_{k=1}^{K}$.
- (P12) jointly learns the union-of-transforms $\{W_k\}$ and clusters the data Y.
- $\lambda_k = \lambda_0 \|Y_{C_k}\|_F^2$, with Y_{C_k} the matrix of all $Y_i \in C_k$, achieves **scale invariance** of the solution in (P12).
    - As $\lambda_0 \to \infty$, $\kappa(W_k) \to 1$, $\left\|W_k\right\|_2 \to \frac{1}{\sqrt{2}}\ \forall\, k$ for the solution in (P12).
- We have proposed a cheap globally convergent alternating algorithm for (P12) [IJCV, 2014].

Unsupervised Classification by OCTOBOS

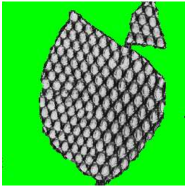
- Overlapping image patches are clustered by learnt OCTOBOS.
- Each pixel is then classified by a vote among the patches that cover it.



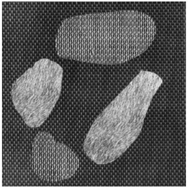
Original Image



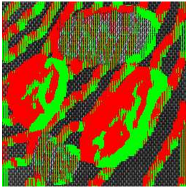
K-means



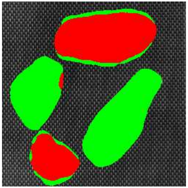
OCTOBOS



Original Image



K-means



OCTOBOS

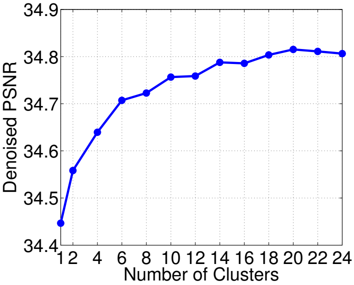
Image Denoising

Image	σ	Noisy PSNR	BM3D	K-SVD	GMM	OCTOBOS
Cameraman	5	34.12	38.21	37.81	38.06	38.19
	10	28.14	34.15	33.72	34.00	34.15
	15	24.61	31.91	31.50	31.85	31.94
	20	22.10	30.37	29.82	30.21	30.24
	100	8.14	23.15	21.76	22.89	22.24
Barbara	5	34.15	38.30	38.08	37.59	38.31
	10	28.14	34.97	34.41	33.61	34.64
	15	24.59	33.05	32.33	31.28	32.53
	20	22.13	31.74	30.83	29.74	31.05
	100	8.11	23.61	21.87	22.13	22.41

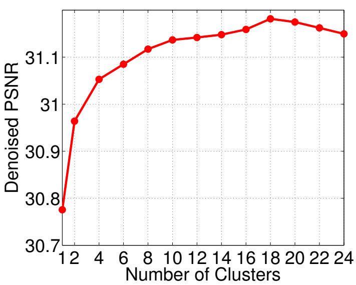
- OCTOBOS denoises 0.36 dB better than K-SVD[19] on avg., and is faster.
- OCTOBOS also denoises 0.43 dB better than GMM[20] on average here.
- Its performance is comparable to BM3D[21] in some cases.

19 [Elad & Aharon '06] 20 [Zoran & Weiss '11] 21 [Dabov et al. '07]

Image Denoising - Effect of Overcompleteness (K)



PSNR for Barbara at $\sigma = 10$



PSNR for Barbara at $\sigma = 20$

- **OCTOBOS denoises up to 0.4 dB better than the square transform here.**
- Best choice of K (number of clusters) lower at higher σ.

Summary

- We proposed learning Union-of-Transforms or OCTOBOS.
- Proposed algorithms have global convergence guarantees.
- Algorithms are cheap and perform well in applications.
- Future Work
  - Combination of OCTOBOS and non-local methods in denoising.
  - Online OCTOBOS learning.

Overall Conclusions and Future Directions

- We proposed several methods for learning square or overcomplete sparsifying transforms.
- Proposed algorithms typically
  - have low computational cost
  - have convergence guarantees
- Adaptive transforms are useful for various applications
  - sparse representation, denoising
  - compressed sensing, classification, big data processing.
- Future Work: Analyze blind denoising or compressed sensing further.

Acknowledgments

- Advisor: Yoram Bresler
- Collaborators: Bihan Wen
- Current & Former Colleagues: Kiryung & Soomin Lee, Luke Pfister, Yanjun Li, Elad Yarkony
- UIUC Staff: Peggy Wells
- Funding: National Science Foundation (NSF)

References - Journals

- S. Ravishankar and Y. Bresler, "MR image reconstruction from highly undersampled k-space data by dictionary learning," IEEE TMI, vol. 30, no. 5, pp. 1028–1041, 2011.
- S. Ravishankar and Y. Bresler, "Learning sparsifying transforms," IEEE TSP, vol. 61, no. 5, pp. 1072–1086, 2013.
- S. Ravishankar and Y. Bresler, "Learning doubly sparse transforms for images," IEEE TIP, vol. 22, no. 12, pp. 4598-4612, 2013.
- B. Wen*, S. Ravishankar*, and Y. Bresler, "Structured overcomplete sparsifying transform learning with convergence guarantees and applications," IJCV, 2014, pp. 1-31. (*Equal contributors).
- S. Ravishankar*, B. Wen*, and Y. Bresler, "Online Sparsifying Transform Learning – Part I: Algorithms," IEEE JSTSP, 2014, accepted for publication. (*Equal contributors).
- S. Ravishankar and Y. Bresler, "Online Sparsifying Transform Learning – Part II: Convergence Analysis," IEEE JSTSP, 2014, accepted for publication.

References - Journals

- S. Ravishankar and Y. Bresler, "ℓ_0 sparsifying transform learning with efficient optimal updates and convergence guarantees," IEEE TSP, 2015, to appear.
- S. Ravishankar and Y. Bresler, "Efficient Blind Compressed Sensing using Sparsifying transforms with convergence guarantees and Application to MRI," SIIMS, 2014, submitted.
- S. Ravishankar and Y. Bresler, "Fast doubly sparse transform learning with convergence guarantees," 2014, manuscript to be submitted.

References - Conferences

- S. Ravishankar and Y. Bresler, "Highly undersampled MRI using adaptive sparse representations," in IEEE ISBI, pp. 1585–1588, 2011.
- S. Ravishankar and Y. Bresler, "Multiscale dictionary learning for MRI," in Proc. ISMRM, page 2830, 2011.
- S. Ravishankar and Y. Bresler, "Adaptive sampling design for compressed sensing MRI," in Conf. Proc. IEEE EMBS, pp. 3751–3755, 2011.
- S. Ravishankar and Y.Bresler, "Learning Sparsifying Transforms for Signal and Image Processing," in SIAM Conference on Imaging Science, May 2012, p. 51.
- S. Ravishankar and Y. Bresler, "Learning sparsifying transforms for image processing," in IEEE ICIP, 2012, pp. 681–684.
- S. Ravishankar and Y. Bresler, "Learning doubly sparse transforms for image representation," in IEEE ICIP, 2012, pp. 685-688.

References - Conferences

- S. Ravishankar and Y. Bresler, “Sparsifying transform learning for compressed sensing MRI,” in Proc. IEEE ISBI, 2013, pp. 17-20.
- S. Ravishankar and Y. Bresler, ”Closed-form solutions within sparsifying transform learning,” in Proc. IEEE ICASSP, 2013, pp. 5378-5382.
- S. Ravishankar and Y. Bresler, ”Learning overcomplete sparsifying transforms for signal processing,” in Proc. IEEE ICASSP, 2013, pp. 3088-3092.
- S. Ravishankar and Y. Bresler, ”Doubly sparse transform learning with convergence guarantees,” in Proc. IEEE ICASSP, 2014, pp. 5262-5266.
- S. Ravishankar and Y. Bresler, “Closed-Form Optimal Updates In Transform Learning,” in SPARS workshop, July 2013.
- S. Ravishankar and Y. Bresler, “Learning Overcomplete Signal Sparsifying Transforms,” in SPARS workshop, July 2013.

References - Conferences

- S. Ravishankar and Y. Bresler, "Efficient Sparsifying Transform Learning and its Applications," in GRC Image Science, June 2014, Presented as Poster.
- B. Wen, S. Ravishankar, and Y. Bresler, "Online Sparsifying Transform Learning and Applications," in GRC Image Science, June 2014, Presented as Poster.
- B. Wen*, S. Ravishankar*, and Y. Bresler, "Learning overcomplete sparsifying transforms with block cosparsity," in IEEE ICIP, October, 2014, to appear. (*Equal contributors).
- S. Ravishankar*, B. Wen*, and Y. Bresler, "Online Sparsifying Transform Learning for Signal Processing," in IEEE GlobalSIP, 2014, to appear. (*Equal contributors).
- B. Wen, S. Ravishankar, and Y. Bresler, "Video Denoising by Online 3D Sparsifying Transform Learning," in IEEE ICIP, 2015, submitted.
- S. Ravishankar and Y. Bresler, "Blind Compressed Sensing using Sparsifying Transforms," in SAMPTA, 2015, submitted.

Thank you! Questions??