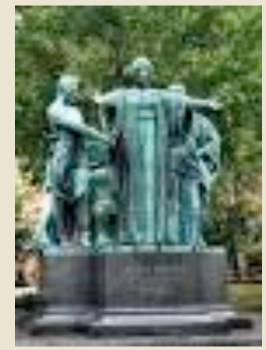
Learning Sparsifying Transforms for Signal, Image, and Video Processing

Yoram Bresler

Coordinated Science Laboratory and the Department of ECE

University of Illinois at Urbana-Champaign



Work with
- Sairprasad Ravishankar
- Bihan Wen
- Luke Pfister

Acknowledgements: NSF Grants CCF-1018660 & CCF-1320953

Andrew T. Yang Fellowship

Santa Clara IEEE Chapter 2016

Overview

Today we will see that

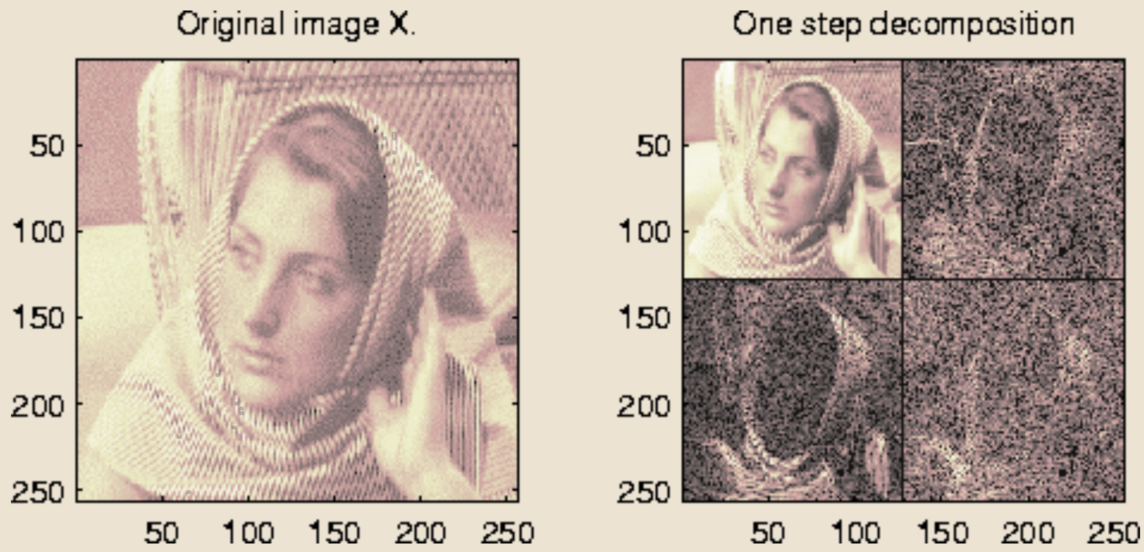
- ⋆ Learned Sparsity Models are valuable and well-founded tools for modeling data
- ⋆ The Transform Learning formulation has computational and performance advantages
- ⋆ When used in imaging and image and video processing, Transform Learning leads to state-of-the-art results

Outline

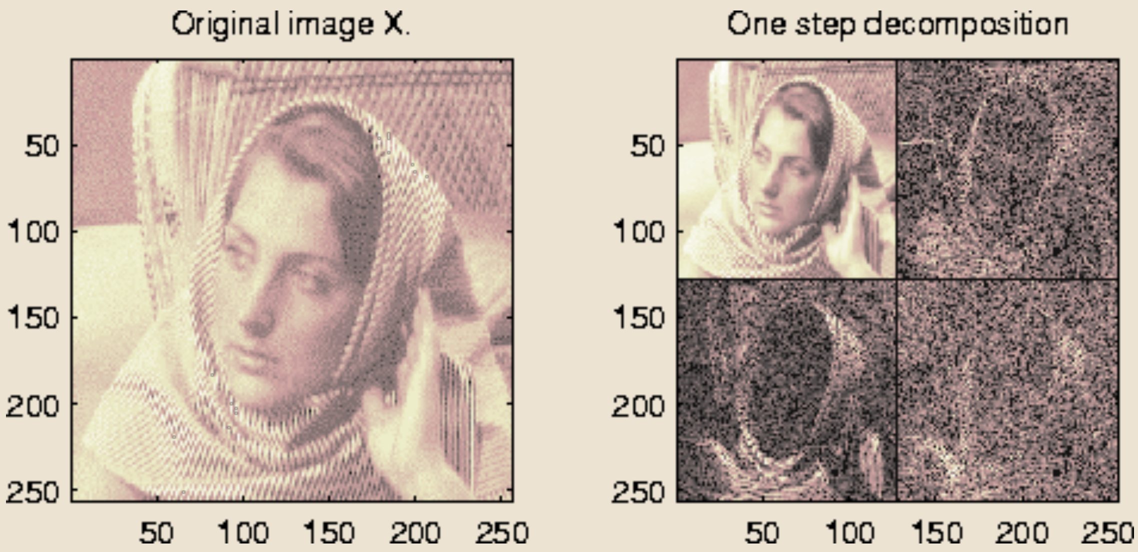
- Sparse signal models – Why and How?
  - Synthesis Dictionaries
  - Sparsifying Transforms
- Basic Transform Learning
- Variations on Transform Learning
  - Union of Transforms for inverse problems
  - Online Transform Learning for big data and video denoising
  - A filter bank formulation of Transform Learning
- Conclusions

Why Sparse Modeling?

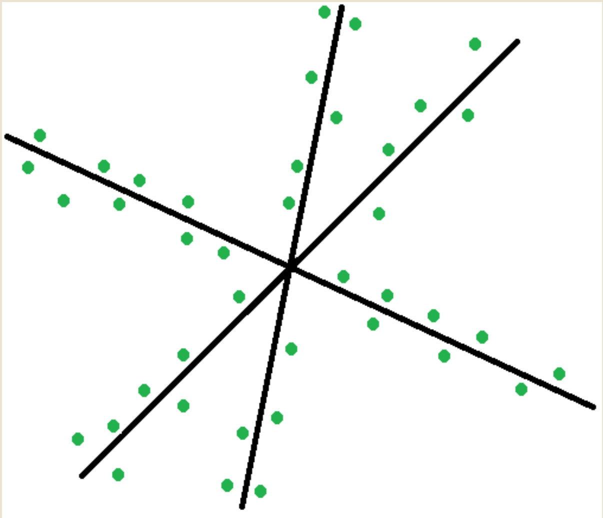
Why Sparse Modeling?



Why Sparse Modeling?



✓ Image data usually lives in low dimensional subspaces



Why Sparse Modeling?

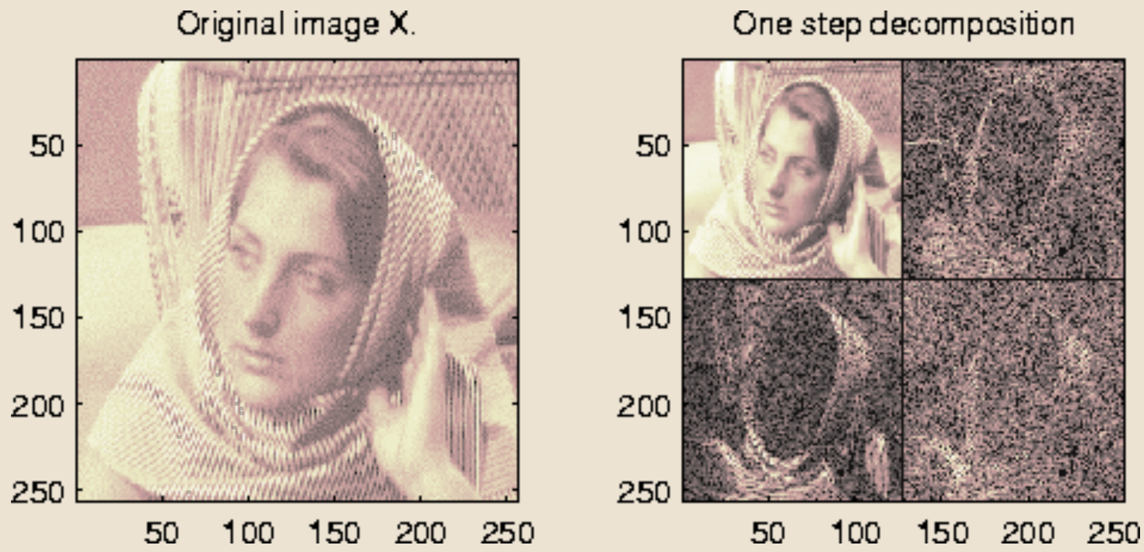
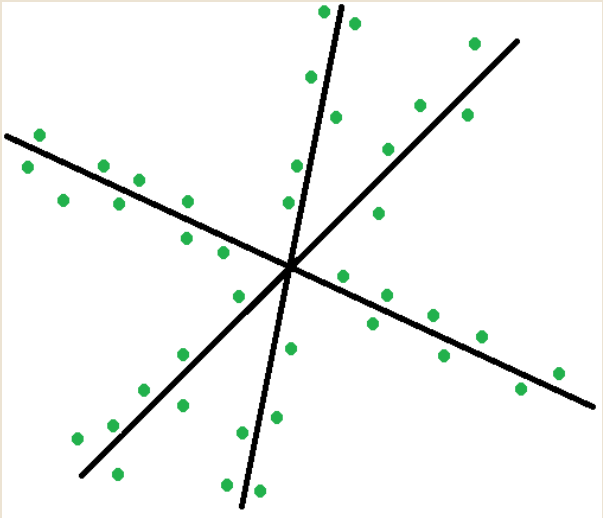


Image data usually lives in low dimensional spaces

Applications:

- Compact representations (compression)
- Regularization in inverse problems
  - Denoising
  - recovery from degraded data
  - Compressed Sensing
- Classification



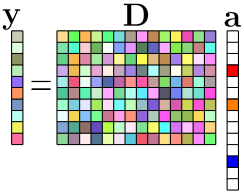
Introduction to Sparse Signal Models

* The Synthesis Dictionary Model
* Learning Synthesis Dictionaries
* The Transform Model

Sparse Representations: Model

- We model $\mathbf{y} \in \mathbb{R}^N$ as

$$\mathbf{y} = \mathbf{D}\mathbf{a}, \quad \|\mathbf{a}\|_0 \leq s$$

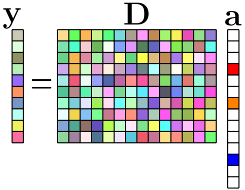


- $\mathbf{a}$ is a sparse coefficient vector
- $\mathbf{D} \in \mathbb{R}^{n \times K}$ is a dictionary. Can be square ($n = K$) or rectangular ($n < K$)
- Columns of $\mathbf{D}$ are called atoms
- $\mathbf{y}$ belongs to a union of subspaces spanned by s atoms of $\mathbf{D}$

Sparse Representations: Model

- We model $\mathbf{y} \in \mathbb{R}^N$ as

$$\mathbf{y} = \mathbf{D}\mathbf{a}, \quad \|\mathbf{a}\|_0 \leq s$$

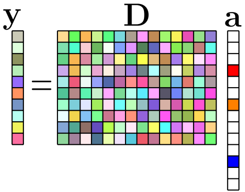


- $\mathbf{a}$ is a sparse coefficient vector
- $\mathbf{D} \in \mathbb{R}^{n \times K}$ is a dictionary. Can be square ($n = K$) or rectangular ($n < K$)
- Columns of $\mathbf{D}$ are called atoms
- $\mathbf{y}$ belongs to a union of subspaces spanned by s atoms of $\mathbf{D}$

Sparse Representations: Model

- We model $\mathbf{y} \in \mathbb{R}^N$ as

$$\mathbf{y} = \mathbf{D}\mathbf{a}, \quad \|\mathbf{a}\|_0 \leq s$$

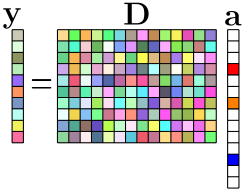


- $\mathbf{a}$ is a sparse coefficient vector
- $\mathbf{D} \in \mathbb{R}^{n \times K}$ is a dictionary. Can be square ($n = K$) or rectangular ($n < K$)
- Columns of $\mathbf{D}$ are called atoms
- $\mathbf{y}$ belongs to a union of subspaces spanned by s atoms of $\mathbf{D}$

Sparse Representations: Model

- We model $\mathbf{y} \in \mathbb{R}^N$ as

$$\mathbf{y} = \mathbf{D}\mathbf{a}, \quad \|\mathbf{a}\|_0 \leq s$$

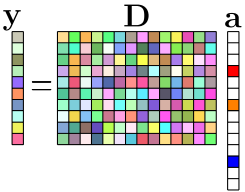


- $\mathbf{a}$ is a sparse coefficient vector
- $\mathbf{D} \in \mathbb{R}^{n \times K}$ is a dictionary. Can be square ($n = K$) or rectangular ($n < K$)
- Columns of $\mathbf{D}$ are called atoms
- $\mathbf{y}$ belongs to a union of subspaces spanned by s atoms of $\mathbf{D}$

Sparse Representations: the Synthesis Model

- Model $\mathbf{y} \in \mathbb{R}^N$ as

$$\mathbf{y} = \mathbf{D}\mathbf{a}, \quad \|\mathbf{a}\|_0 \leq s$$



- $\mathbf{a}$ is a sparse coefficient vector
- $\mathbf{D} \in \mathbb{R}^{n \times K}$ is a dictionary. Can be square ($n = K$) or rectangular ($n < K$)
- Columns of $\mathbf{D}$ are called atoms
- $\mathbf{y}$ belongs to a union of subspaces spanned by s atoms of $\mathbf{D}$

Sparse Representations: Sparse Coding

- Given an overcomplete $\mathbf{D}$ and vector $\mathbf{y}$, how can we find the sparsest $\mathbf{a}$ such that $\mathbf{y} = \mathbf{Da}$?
- Solve

$$\begin{aligned} \min_{a} \quad & \|\mathbf{a}\|_0 \\ \text{subject to} \quad & \mathbf{y} = \mathbf{Da} \end{aligned}$$

- NP-Hard![1] Look for approximate solutions
  - Convex Relaxation
    - Basis pursuit
  - Greedy Algorithms
    - Orthogonal Matching Pursuit (OMP)

[1] Natarajan, 1995

Sparse Representations: Sparse Coding

- Given an overcomplete $\mathbf{D}$ and vector $\mathbf{y}$, how can we find the sparsest $\mathbf{a}$ such that $\mathbf{y} = \mathbf{Da}$?
- Solve

$$\begin{aligned} \min_{a} \quad & \|\mathbf{a}\|_0 \\ \text{subject to} \quad & \mathbf{y} = \mathbf{Da} \end{aligned}$$

- NP-Hard![1] Look for approximate solutions
  - Convex Relaxation
    - Basis pursuit
  - Greedy Algorithms
    - Orthogonal Matching Pursuit (OMP)

[1] Natarajan, 1995

Sparse Representations: Sparse Coding

- Given an overcomplete $\mathbf{D}$ and vector $\mathbf{y}$, how can we find the sparsest $\mathbf{a}$ such that $\mathbf{y} = \mathbf{Da}$?
- Solve

$$\begin{aligned} \min_{a} \quad & \|\mathbf{a}\|_0 \\ \text{subject to} \quad & \mathbf{y} = \mathbf{Da} \end{aligned}$$

- NP-Hard![1] Look for approximate solutions
  - Convex Relaxation
    - Basis pursuit
  - Greedy Algorithms
    - Orthogonal Matching Pursuit (OMP)

[1] Natarajan, 1995

Sparse Representations: Sparse Coding

- Given an overcomplete $\mathbf{D}$ and vector $\mathbf{y}$, how can we find the sparsest $\mathbf{a}$ such that $\mathbf{y} = \mathbf{D}\mathbf{a}$?
- Solve

$$\begin{aligned} \min_{a} \quad & \|\mathbf{a}\|_0 \\ \text{subject to} \quad & \mathbf{y} = \mathbf{D}\mathbf{a} \end{aligned}$$

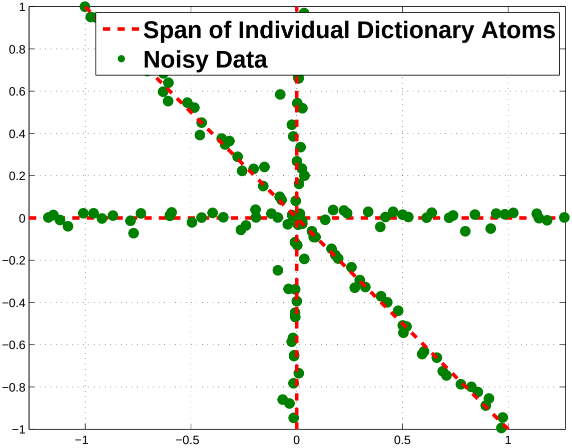
- NP-Hard![1] Look for approximate solutions
  - Convex Relaxation
    - Basis pursuit
  - Greedy Algorithms
    - Orthogonal Matching Pursuit (OMP)

[1] Natarajan, 1995

Sparse Representations: Denoising

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

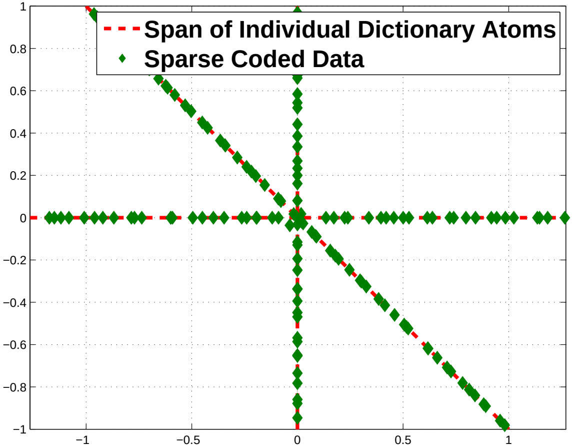
$$s = 1$$



Sparse Representations: Denoising

$$\mathbf{D} = \begin{bmatrix} 1 & 0 & \frac{1}{\sqrt{2}} \\ 0 & 1 & -\frac{1}{\sqrt{2}} \end{bmatrix}$$

$s = 1$



What are good dictionaries for sparse representation of signals and images?

The more sparse the representation, the better!

What are good dictionaries for sparse representation of signals and images?

The more sparse the representation, the better!

Analytic Dictionaries

- Design dictionary around a predefined set of functions
  - Fourier
  - Wavelet
  - Curvelet
  - Contourlet

    $\vdots$

- Fast implementations

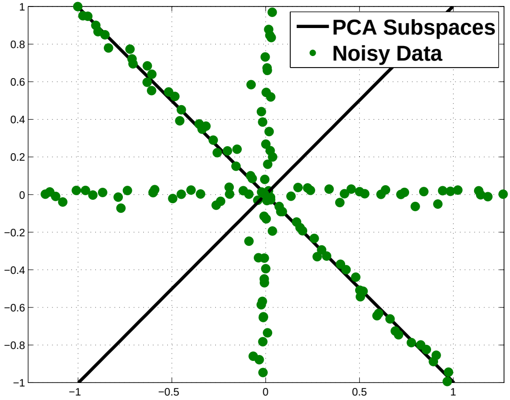
- But, hard to design effective dictionaries in high dimensions

Adaptive Dictionaries

- Adaptively learn dictionary from data itself
- Karhunen-Loève/PCA: fit low-dim subspace to minimize ℓ_2 error

Adaptive Dictionaries

- Adaptively learn dictionary from data itself
- Karhunen-Loève/PCA: fit low-dim subspace to minimize ℓ_2 error



Dictionary Learning

- Training data: $\{\mathbf{y}_j\}_{j=1}^M \in \mathbb{R}^N$
- Want:

$$\begin{aligned}
\mathbf{y}_1 &= \mathbf{D}\mathbf{a}_1, & \|\mathbf{a}_1\|_0 \le s \\
\mathbf{y}_2 &= \mathbf{D}\mathbf{a}_2, & \|\mathbf{a}_2\|_0 \le s \\
&\vdots \\
\mathbf{y}_M &= \mathbf{D}\mathbf{a}_M, & \|\mathbf{a}_M\|_0 \le s
\end{aligned}$$

- **Dictionary Learning:** Given a set of training signals $\{\mathbf{y}_j\}_{j=1}^M$ formed into a matrix $\mathbf{Y} \in \mathbb{R}^{N\times M}$, we seek to find $\mathbf{D} \in \mathbb{R}^{N\times K}$, $\mathbf{A} \in \mathbb{R}^{K\times M}$ such that $\mathbf{Y} \approx \mathbf{D}\mathbf{A}$ with $\|\mathbf{a}_j\|_0 \le s$

Summary: Learning Synthesis Dictionary Models

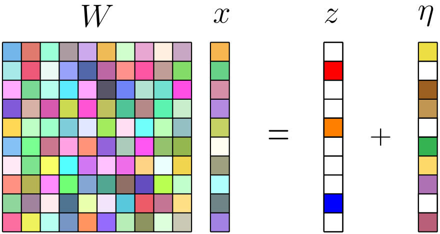
The Good

- Sparsity in an appropriate dictionary is a powerful model; learned sparsity models even better!
- Many successful applications: denoising, in-painting, image super resolution, compressed sensing(MRI, CT), classification, etc.

The Bad

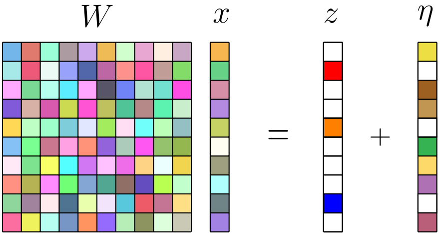
- Synthesis sparse coding solved repeatedly for learning is NP-hard
- Approximate synthesis sparse coding algorithms can be computationally expensive
- The synthsis dictionary learning problem is highly non-convex, and algorithms can get stuck in bad local minima

A Classical Alternative: Transform Sparsity



- W: Sparsifying transform
- z: Sparse Code
- $Wx \approx$ sparse
- Approximation error in the transform domain: $\|\eta\|_2 \ll \|z\|_2$

A Classical Alternative: Transform Sparsity



- W: Sparsifying transform
- z: Sparse Code
- $Wx \approx$ sparse
- Approximation error in the transform domain: $\|\eta\|_2 \ll \|z\|_2$

Transform Sparse Coding

$$z^* = \arg\min_z \frac{1}{2}\|Wx - z\|_2^2$$

subject to $\|z\|_0 \leq s$ sparsity constraint

Easy Exact Solution:

$z^* \triangleq H_s(Wx)$ Thresholding to s largest elements

Transform Sparse Coding

$$z^* = \arg\min_z \frac{1}{2}\|Wx - z\|_2^2$$

subject to $\|z\|_0 \leq s$ sparsity constraint

Easy Exact Solution:

$z^* \triangleq H_s(Wx)$ Thresholding to s largest elements

Transform Sparse Coding

- Penalized Form

$$z^* = \min_z \frac{1}{2}\|Wx - z\|_2^2 + \nu\|z\|_0$$

Easy Exact Solution:

$$z_i^* = \begin{cases} [Wx]_i & |[Wx]_i| \geq \sqrt{\nu} \\ 0 & \text{else} \end{cases}$$

$$\triangleq \mathcal{T}_\nu(Wx) \quad \text{Hard Thresholding}$$

Transform Sparse Coding

- Penalized Form

$$z^* = \min_z \frac{1}{2}\|Wx - z\|_2^2 + \nu\|z\|_0$$

Easy Exact Solution:

$$z_i^* = \begin{cases} [Wx]_i & |[Wx]_i| \geq \sqrt{\nu} \\ 0 & \text{else} \end{cases}$$

$$\triangleq \mathcal{T}_\nu\left(Wx\right)$$ Hard Thresholding

Summary of the Models

- SM : finding x with given D is NP-hard.

$$y = Dx + e\,,\ \ \|x\|_0 \le s \tag{1}$$

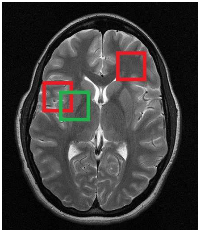
- NSAM : finding q with given Ω is NP-hard.

$$y = q + e\,,\ \ \|\Omega q\|_0 \le m - t \tag{2}$$

- TM : finding x with given W is easy $\Rightarrow$ efficiency in applications.

$$Wy = x + \eta\,,\ \ \|x\|_0 \le s \tag{3}$$

Learning on Image Patches



Patches of image

- $Y_j = R_j y$, $j = 1,..N$: jth image patch, vectorized.
- $Y = [\, Y_1 \,|\, Y_2 \,|\, ... \,|\, Y_N \,] \in \mathbb{R}^{n \times N}$: matrix of vectorized patches - training signals

Basic Transform Learning Formulation[6]

$$
\text{(P0)} \quad \min_{W,X} \overbrace{\|WY - X\|_F^2}^{\text{Sparsification Error}}
$$
$$
\text{s.t.} \quad \|X_i\|_0 \leq s \;\; \forall \; i
$$

- $Y = [\, Y_1 \,|\, Y_2 \,|\, \ldots.. \,|\, Y_N \,] \in \mathbb{R}^{n \times N}$: matrix of training signals
- $X = [\, X_1 \,|\, X_2 \,|\, \ldots.. \,|\, X_N \,] \in \mathbb{R}^{n \times N}$: matrix of sparse codes of Y_i
- $W \in \mathbb{R}^{n \times n}$: square transform
- Sparsification error - measures deviation of data in transform domain from perfect sparsity

[6] [Ravishankar & Bresler ICIP 2012, TSP 2013, TSP 2015]

Basic Transform Learning Formulation[6]

$$(\mathrm{P0}) \quad \min_{W,X} \overbrace{\|WY - X\|_F^2}^{\text{Sparsification Error}} + \overbrace{\lambda\left(\|W\|_F^2 - \log|\det W|\right)}^{\text{Regularizer} \triangleq \lambda v(W)}$$
$$\text{s.t.} \quad \|X_i\|_0 \leq s \;\; \forall\, i$$

- $Y = [\, Y_1 \,|\, Y_2 \,|\, ... \,|\, Y_N \,] \in \mathbb{R}^{n \times N}$: matrix of training signals
- $X = [\, X_1 \,|\, X_2 \,|\, ... \,|\, X_N \,] \in \mathbb{R}^{n \times N}$: matrix of sparse codes of Y_i
- $W \in \mathbb{R}^{n \times n}$: square transform
- Sparsification error - measures deviation of data in transform domain from perfect sparsity
- $\lambda > 0$. Regularizer cost $v(W)$ prevents trivial solutions and fully controls condition number of W

[6] [Ravishankar & Bresler ICIP 2012, TSP 2013, TSP 2015]

Alternating Algorithm for Transform Learning

- (P0) solved by alternating between updating X and W.

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

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

  - **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.

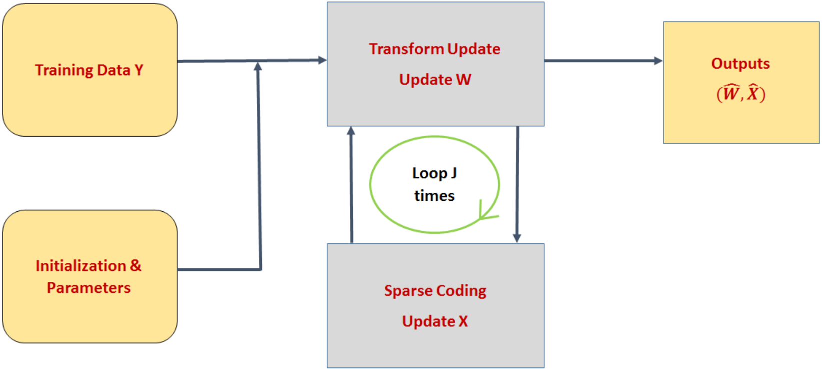
$$\min_{W} \|WY - X\|_F^2 + \lambda\left(\|W\|_F^2 - \log|\det W|\right) \tag{2}$$

  - **Closed-form solution:**

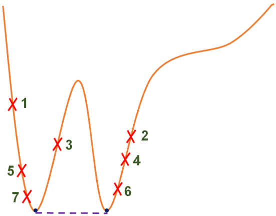
$$\hat{W} = 0.5U\left(\Sigma + \left(\Sigma^2 + 2\lambda I\right)^{\frac{1}{2}}\right)Q^T L^{-1} \tag{3}$$

    - $YY^T + \lambda I = LL^T$, and $L^{-1}YX^T$ has a full SVD of $Q\Sigma U^T$.

Algorithm A1 for Square Transform Learning



Convergence Guarantees[7]



Theorem 1

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

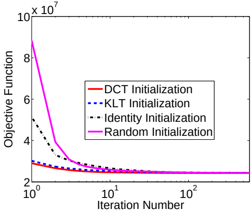
Corollary 1

Algorithm A1 is globally convergent (i.e., from any Initialization) to the set of local minimizers in the problem.

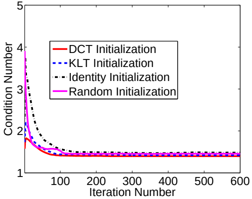
[7] [Ravishankar & Bresler, TSP 2015]

Convergence with Various Initializations

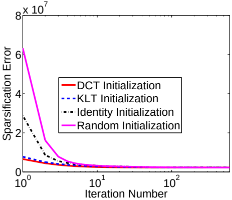
Barbara - 8×8 patches



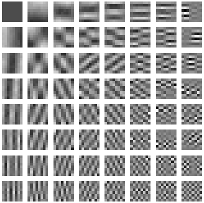
Objective Function



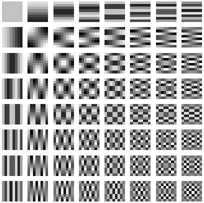
$\kappa(W)$



Sparsification Error ($s = 11$)



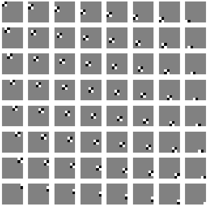
Learned W - DCT Init



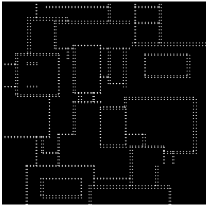
2D DCT

Piecewise-Constant Images

Image



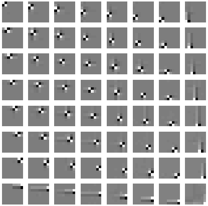
Finite difference (FD)
$\kappa(W) = 113.5$



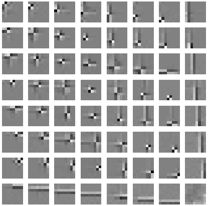
Sparse Result
$s = 5$

- 2D FD obtained as kronecker product of two square 1D-FD matrices - exact sparsifier for patches of image for $s \geq 5$.
- However, the 2D FD transform is poorly conditioned.

Well-Conditioned Adaptive Transforms Perform Well!



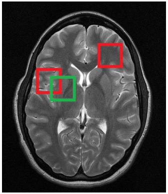
Learnt (FD Init)
$\kappa(W) = 15.35$



Learnt (FD Init)
$\kappa(W) = 5.77$

- The learnt transforms provide almost zero NSE ($\sim 10^{-4}/10^{-5}$).
- Such well-conditoned transforms perform better than poorly conditioned ones in applications such as denoising.
- For $s < 5$, the learnt well-conditioned transforms provide significantly lower NSE at the same s, than FD.

Computational Advantages



Patches of image

- Synthesis/Analysis K-SVD[8,9] for N training samples and $D \in \mathbb{R}^{n \times K}$: cost per iteration (dominated by sparse coding):

  $O(Nn^3) \propto (\text{Image Size}) \times (\text{pixels in patch})^3$
- Transform Learning Algorithm A1 for N training samples and $W \in \mathbb{R}^{n \times n}$: $O(Nn^2) \propto (\text{Image Size}) \times (\text{pixels in patch})^2$
- In 2D with $p \times p$ patches $\Rightarrow$reduction of computations in the order by p^2
- In 3D with $p \times p \times p$ patches $\Rightarrow$ reduction of computations in the order by p^3 (=1000X for $p = 10$)

8 [Aharon, Elad & Bruckstein '06] 9 [Rubinstein, Peleg & Elad '13]

Does Transform Learning work? : Denoising Example

Noisy Image
PSNR = 24.60 dB

64×256 Synthesis D
PSNR = 31.50 dB

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

- Transform learning-based denoising is better and highly efficient (17X faster) compared to overcomplete K-SVD denoising.

Compressed Sensing with a Learned Transform

- CS enables accurate recovery of images from far fewer measurements than the number of unknowns
  - Sparsity of image in transform domain or dictionary
  - Measurement procedure incoherent with transform
  - **Reconstruction non-linear**
- Conventional CS Reconstruction problem -

$$\min_{x} \overbrace{\|Ax - y\|_2^2}^{\text{Data Fidelity}} + \lambda \overbrace{\|\Psi x\|_0}^{\text{Regularizer}} \tag{4}$$

- $x \in \mathbb{C}^P$: vectorized image, $y \in \mathbb{C}^m$: measurements $(m < P)$.
- A : fat sensing matrix, Ψ : transform. ℓ_0 "norm" counts non-zeros.
- CS with non-adaptive regularizer limited to low undersampling in imaging.

Compressed Sensing MRI

- Data - samples in k-space of spatial Fourier transform of object, acquired sequentially in time.
- Acquisition rate limited by MR physics, physiological constraints on RF energy deposition.
- CSMRI enables accurate recovery of images from far fewer measurements than # unknowns or Nyquist sampling.
- Two directions to improve CSMRI -
  - **better sparse modeling - TLMRI**
  - better choice of sampling pattern (F_u)

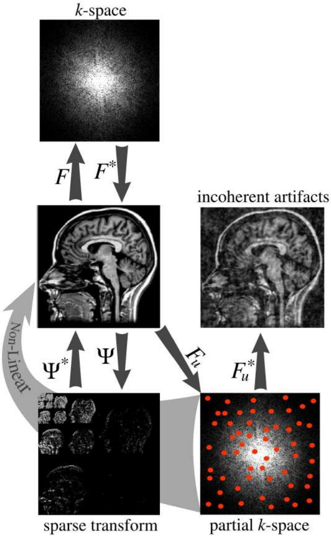


Fig. from Lustig et al. '07

Transform Blind Compressed Sensing Idea

- ~~*Could* use an image database to train the sparsifying transform~~
- Learn transform W to sparsify the *unknown image* x using only the undersampled data $y \approx Ax$
- $\Rightarrow$ **model adaptive to underlying image.**
- Use the learned transform W to perform compressed sensing reconstruction of the image x from undersampled data y

Transform Blind Compressed Sensing Idea

- *Could* use a ~~database to train~~ the sparsifying transform
- Learn transform W to sparsify the *unknown image* x using only the undersampled data $y \approx Ax$
- $\Rightarrow$ **model adaptive to underlying image.**
- Use the learned transform W to perform compressed sensing reconstruction of the image x from undersampled data y

Transform-based Blind Compressed Sensing (BCS)

$$(\mathrm{P1}) \quad \min_{x,W,B} \overbrace{\sum_{j=1}^{N} \|WR_j x - b_j\|_2^2}^{\text{Sparsification Error}} + \nu \overbrace{\|Ax - y\|_2^2}^{\text{Data Fidelity}} + \lambda \overbrace{v(W)}^{\text{Regularizer}}$$

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

- (P1) 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.
- $\|x\|_2 \leq C$ is an energy/range constraint. $C > 0$.

Block Coordinate Descent (BCD) Algorithm for (P1)

- Alternate the updating of W, B, and x.

- **Sparse Coding Step:** solve (P1) for B with fixed x, W.

$$\min_{B} \sum_{j=1}^{N} \|WR_j x - b_j\|_2^2 \quad s.t. \quad \sum_{j=1}^{N} \|b_j\|_0 \leq s. \tag{5}$$

  - **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.

Block Coordinate Descent Algorithm for (P1)

- **Transform Update Step:** solve (P1) for W with fixed x, B.

$$\min_{W} \sum_{j=1}^{N} \|WR_jx - b_j\|_2^2 + 0.5\lambda \|W\|_F^2 - \lambda \log |\det W| \qquad (6)$$

  - **Exact Closed-form solution involving SVD of a small matrix**

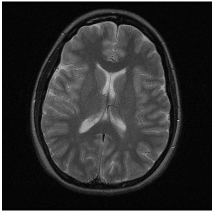
Block Coordinate Descent Algorithm for (P1)

- **Image Update Step:** solve (P1) for x with fixed W, B.

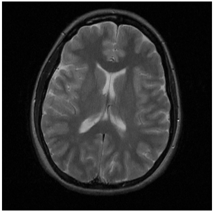
$$\min_{x} \sum_{j=1}^{N} \|WR_j x - b_j\|_2^2 + \nu \|Ax - y\|_2^2 \;\; s.t. \;\; \|x\|_2 \leq C. \qquad (7)$$

  - Standard least squares problem with ℓ_2 norm constraint. For MRI can be solved iteratively efficiently using CG+ FFT.

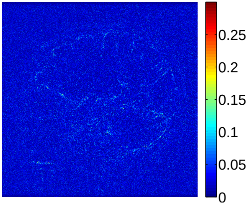
Example - 2D Cartesian 7x Undersampling



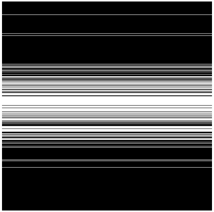
Reference



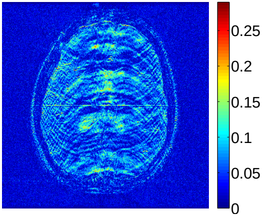
TLMRI (31 dB)



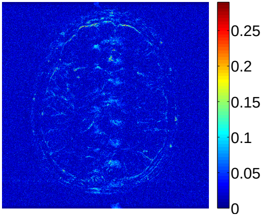
TLMRI Error



Sampling Mask

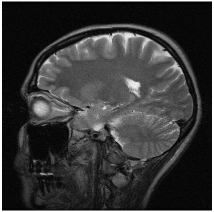


Sparse MRI Error (25.5dB)
Lustig et al, 2007



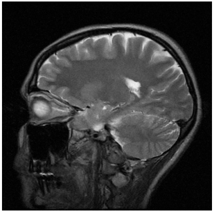
DLMRI Error (30.7 db)
Saiprasad & Bresler, 2011

Example - 2D random 5x Undersampling

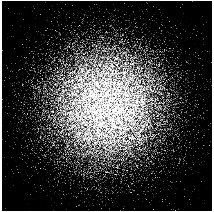


Reference

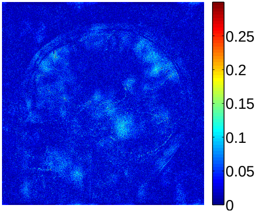
DLMRI (28.54 dB)



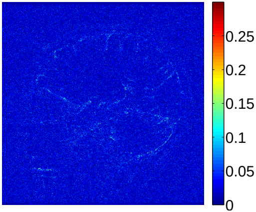
TLMRI (30.47 dB)



Sampling Mask



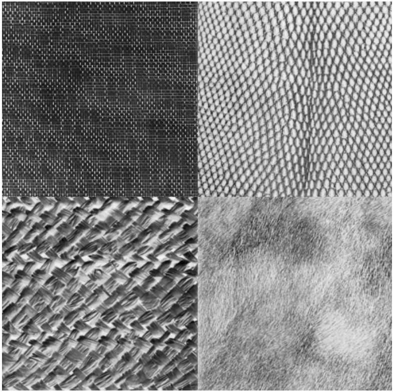
DLMRI Error



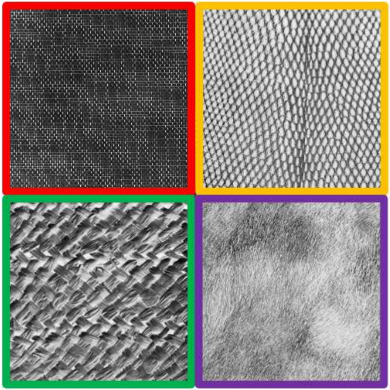
TLMRI Error

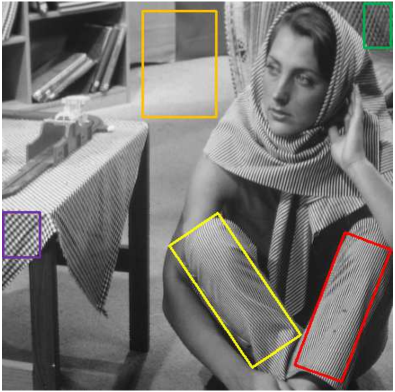
The More the Merrier?

- A single square transform is learned in the Basic TL Algorithm for all the data.
- But, natural images typically have diverse features or textures.



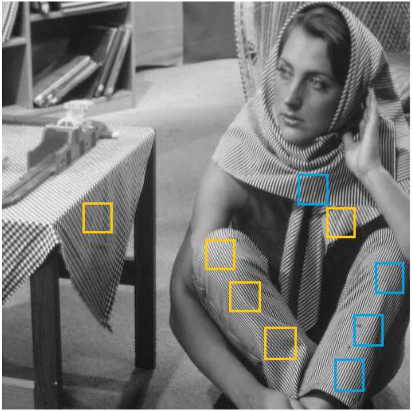
- **Union of transforms: one for each class of textures or features.**

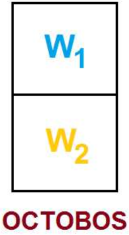




OCTOBOS Learning Idea

- Group patches based on their match to a common transform.
- **Learn the transforms + cluster the data jointly**





OCTOBOS Learning Formulation

- **Goal:** jointly learn a union-of-transforms $\{W_k\}$ and cluster the data Y.

$$(\text{P2}) \quad \min_{\{W_k, X_i, C_k\}} \overbrace{\sum_{k=1}^{K} \sum_{i \in C_k} \|W_k Y_i - X_i\|_2^2}^{\text{Sparsification Error}} + \overbrace{\sum_{k=1}^{K} \lambda_k \left(\|W_k\|_F^2 - \log |\det W_k| \right)}^{\text{Regularizer} = \sum_{k=1}^{K} \lambda_k v(W_k)}$$

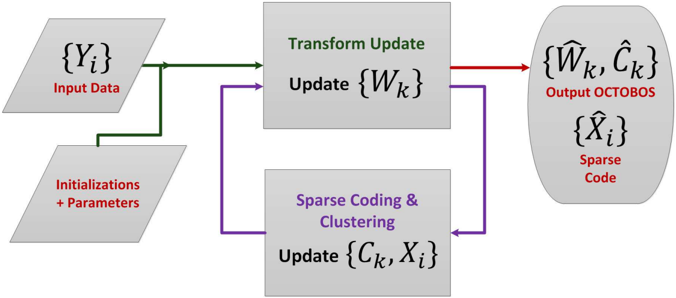
$$\text{s.t.} \quad \|X_i\|_0 \leq s \ \forall i, \quad \{C_k\}_{k=1}^{K} \in G$$

- C_k is the set of indices of signals in class k.
- G is the set of all possible partitions of $[1:N]$ into K disjoint subsets.
- The regularizer controls the scaling and conditioning of the transforms

Alternating Minimization Algorithm for (P2)

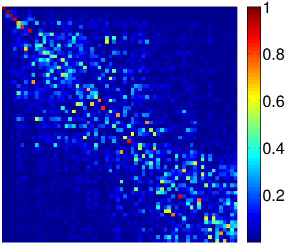
$$(\mathrm{P2}) \quad \min_{\{W_k, X_i, C_k\}} \overbrace{\sum_{k=1}^{K} \sum_{i \in C_k} \|W_k Y_i - X_i\|_2^2}^{\text{Sparsification Error}} + \overbrace{\sum_{k=1}^{K} \lambda_0 \|Y_{C_k}\|_F^2 v(W_k)}^{\text{Regularizer}}$$

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

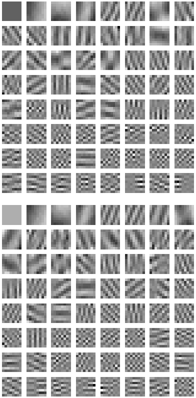


Visualization of Learned OCTOBOS

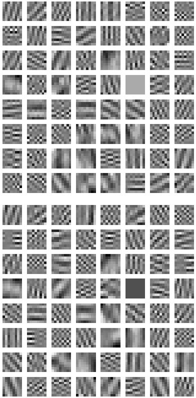
- The square blocks of a learnt OCTOBOS are **NOT** similar $\Rightarrow$ cluster-specific W_k.
- OCTOBOS **W** learned with different initializations appear different.
- **The W learned with different initializations sparsify equally well.**



Cross-gram matrix between W_1 and W_2 for KLT Init.



Random matrix Init.



KLT Init.

Example: Unsupervised Classification

- The overlapping image patches are first clustered by OCTOBOS learning
- Each image pixel is then classified by a majority vote among the patches that cover that pixel

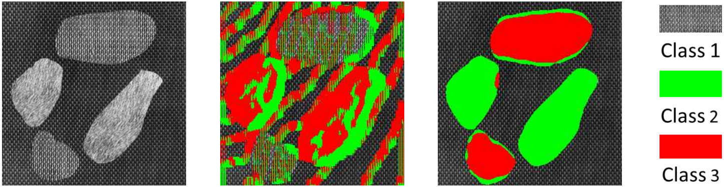


Image　　k-Means　　OCTOBOS

Imaging:

Transform Blind Compressed Sensing with a Union of Transforms

UNITE-BCS: Union of Transforms Blind CS

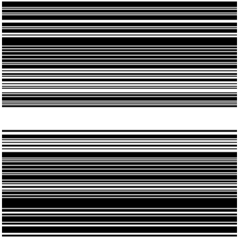
- **Goal:** learn union of transforms, reconstruct x, and cluster the patches of x, using only the undersampled y.
  - $\Rightarrow$ **model adaptive to underlying image.**

$$\text{(P2)} \quad \min_{x, B, \{W_k, C_k\}} \nu \overbrace{\|Ax - y\|_2^2}^{\text{Data Fidelity}} + \overbrace{\sum_{k=1}^{K} \sum_{j \in C_k} \|W_k R_j x - b_j\|_2^2}^{\text{Sparsification Error}} + \eta^2 \overbrace{\sum_{j=1}^{N} \|b_j\|_0}^{\text{Sparsity Penalty}}$$

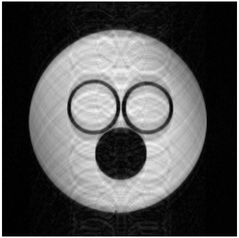
$$\text{s.t. } W_k^H W_k = I \ \forall\, k, \ \ \|x\|_2 \leq C.$$

- $R_j \in \mathbb{R}^{n \times P}$ extracts patches. $W_k \in \mathbb{C}^{n \times n}$ is a unitary cluster transform.
- $\|x\|_2 \leq C$ is an energy or range constraint. $B \triangleq [b_1 \mid b_2 \mid \ldots \mid b_N]$.
- Efficient alternating algorithm for (P2) with convergence guarantee

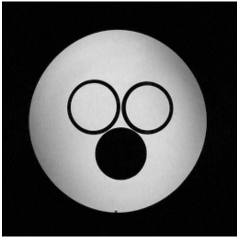
CS MRI Example - 2.5x Undersampling ($K = 3$)



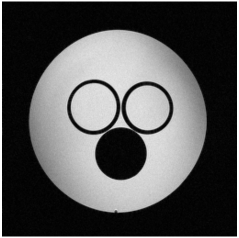
Sampling mask



Initial recon (24.9 dB)

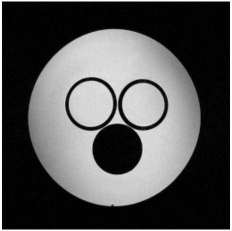


UNITE-MRI recon (37.3 dB)

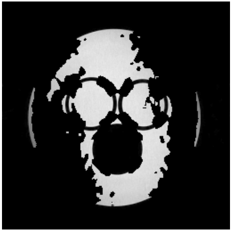


Reference

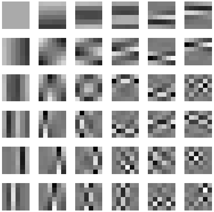
UNITE-MRI Clustering with $K = 3$ ($\eta = 0.07$, $\nu = 15.3$)

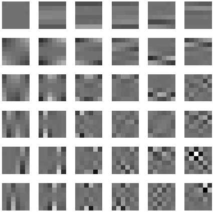

UNITE-MRI recon


Cluster 1

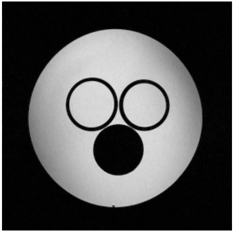

Cluster 2


Cluster 3

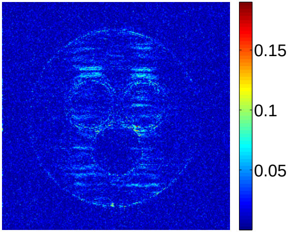

Real part of learned W for cluster 2


Imaginary part of learned W for cluster 2

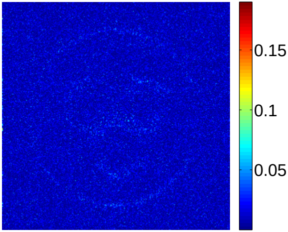
Reconstructions - Cartesian 2.5x Undersampling ($K = 16$)



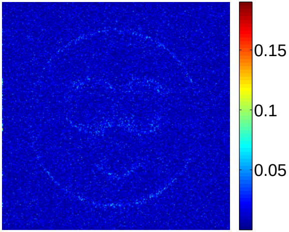
UNITE-MRI recon (37.4 dB)



PANO[11] error (34.8 dB)



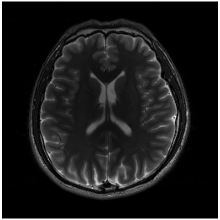
UNITE-MRI error



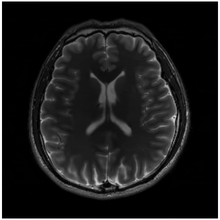
UTMRI ($K = 1$) error (37.2 dB)

[11] [Qu et al. '14]

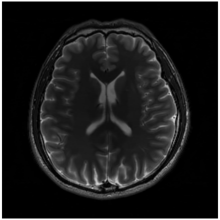
Example - Cartesian 2.5x Undersampling ($K = 16$)



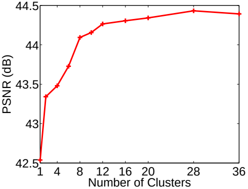
Reference



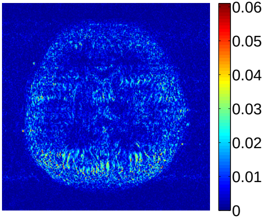
UTMRI (42.5 dB)



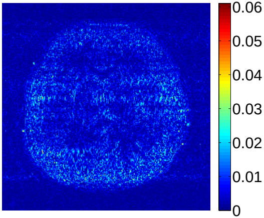
UNITE-MRI (44.3 dB)



PSNR vs. K



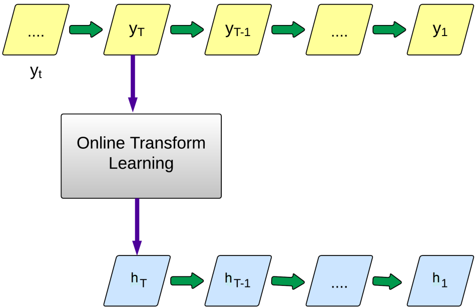
UTMRI Error



UNITE-MRI Error

Online Transform Learning for Dynamic Imaging and Big Data

Online Transform Learning



h_t : Learnt Transform/Sparse Codes/Signal Estimates

- Big data $\Rightarrow$ large training sets $\Rightarrow$ batch learning (using all data) is computationally expensive in time and memory.
- Streaming data $\Rightarrow$ must be processed sequentially to limit latency.
- Online learning involves cheap computations and memory usage.

Online Transform Learning Formulation

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

$$(\mathrm{P3})\ \left\{\hat{W}_t, \hat{x}_t\right\} = \underset{W,\, x_t}{\arg\min}\ \frac{1}{t}\sum_{j=1}^{t}\left\{\|Wy_j - x_j\|_2^2 + \lambda_j v(W)\right\}$$
$$\text{s.t.}\ \ \|x_t\|_0 \leq s,\ x_j = \hat{x}_j,\ 1 \leq j \leq t-1.$$

- $\lambda_j = \lambda_0 \|y_j\|_2^2$. λ_0 controls condition number and scaling of $\hat{W}_t \in \mathbb{R}^{n\times n}$.
- Denoised image estimate $\hat{y}_t = \hat{W}_t^{-1}\hat{x}_t$ is computed efficiently.
- For non-stationary data, use forgetting factor $\rho \in [0, 1]$, to diminish the influence of old data.

$$\frac{1}{t}\sum_{j=1}^{t}\rho^{t-j}\left\{\|Wy_j - x_j\|_2^2 + \lambda_j v(W)\right\} \tag{12}$$

Online Transform Learning Algorithm

- **Sparse Coding** - solve for x_t in (P3) with fixed $W = \hat{W}_{t-1}$: **Cheap Solution:** $\hat{x}_t = H_s(Wy_t)$.

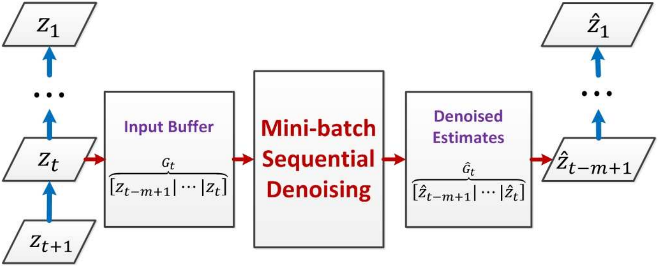
- **Transform Update:** solve for W in (P3) with $x_t = \hat{x}_t$. Cheap, closed-form update using SVD rank-1 update.
- **No matrix-matrix products.** Approx. error bounded, and cheaply monitored.

- **Assumption:** y_t are i.i.d. random samples from the sphere $S^n = \{y \in \mathbb{R}^n : \|y\|_2 = 1\}$.
- Consider the minimization of the expected learning cost:

$$g(W) = \mathbb{E}_y \left[\|Wy - H_s(Wy)\|_2^2 + \lambda_0 \|y\|_2^2 \, v(W) \right] \tag{13}$$

- Mild assumptions: Exact computations, Nondegenerate SVDs.
- **Main Result: $\hat{W}_t$ in OTL converges to the set of stationary points of $g(W)$ almost surely.** $\hat{W}_{t+1} - \hat{W}_t \sim O(1/t)$.

Online Video Denoising by 3D Transform Learning

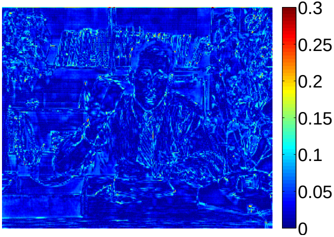


- z_t is a noisy video frame. $\hat{z}_t$ is its denoised version.
- G_t is a tensor with m frames formed using a sliding window scheme.
- Overlapping 3D patches in the G_t's are denoised sequentially.
- Denoised patches averaged at 3D locations to yield frame estimates.

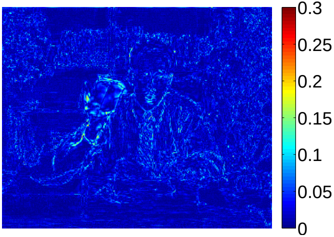
Video Denoising Example: Salesman

Noisy frame

VIDOLSAT (PSNR $=$ 30.97 dB)



VBM4D[12] Error (PSNR $=$ 27.20 dB)

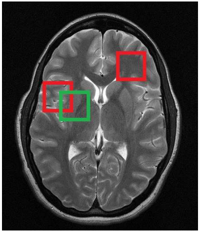


VIDOLSAT Error

12 [Maggioni et al. '12]

From Patches To Filter Banks

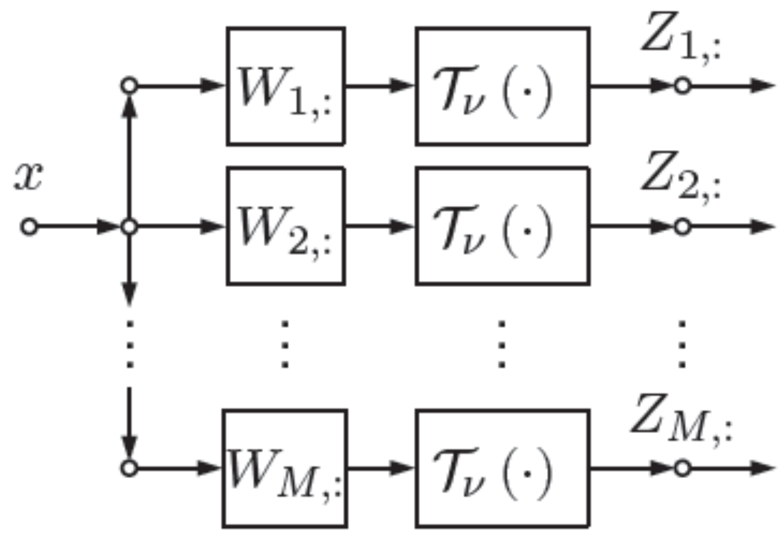
Learning on Image Patches



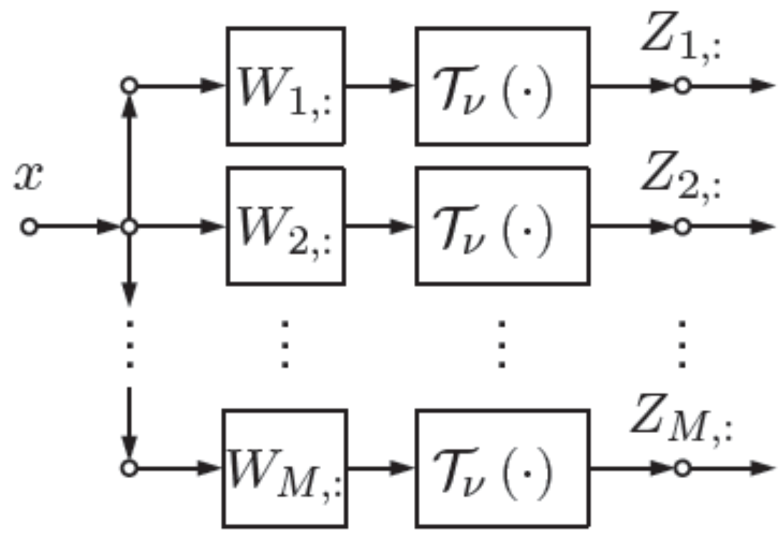
Patches of image

- $Y_j = R_j y$, $j = 1,..N$: jth image patch, vectorized.
- $Y = [\, Y_1 \,|\, Y_2 \,|\, ... \,|\, Y_N \,] \in \mathbb{R}^{n \times N}$: matrix of vectorized patches - training signals

Sparsifying Transforms as Filter Banks for Maximally Overlapping Patches

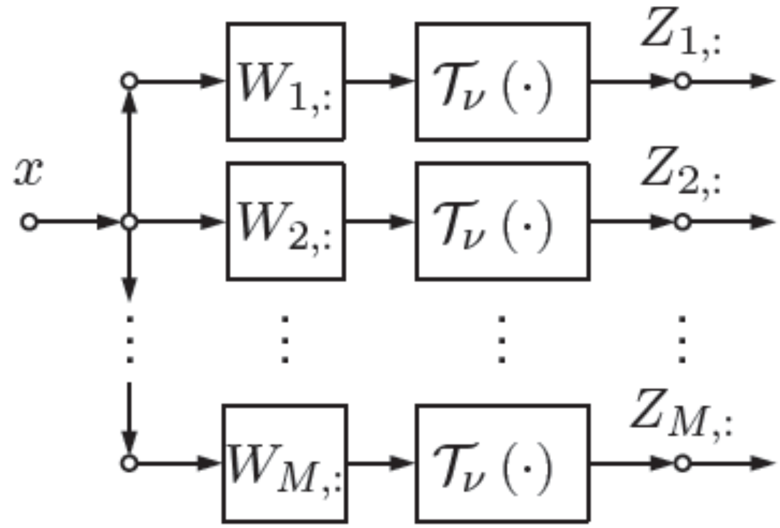


Sparsifying Transforms as Filter Banks for Maximally Overlapping Patches



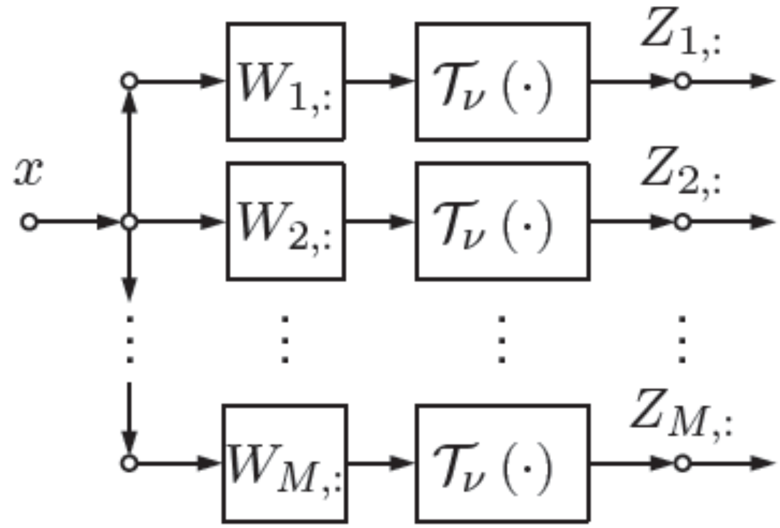
- $\mathrm{vec}(WX) = \mathcal{H}_W x$

Sparsifying Transforms as Filter Banks for Maximally Overlapping Patches



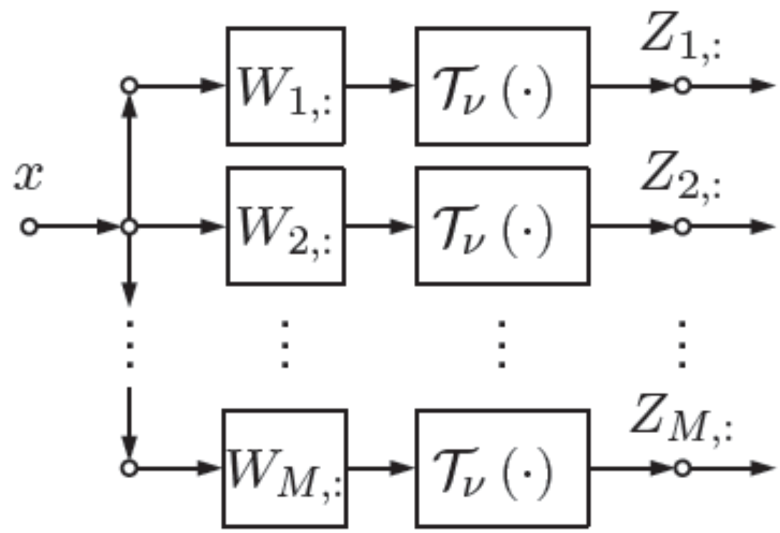
- $\mathrm{vec}(WX) = \mathcal{H}_W x$
- Defn: $\mathcal{H}_W$ is perfect reconstruction (PR) if $\mathcal{H}_W$ left invertible (LI).

Sparsifying Transforms as Filter Banks for Maximally Overlapping Patches



- $\operatorname{vec}(WX) = \mathcal{H}_W x$
- Defn: $\mathcal{H}_W$ is perfect reconstruction (PR) if $\mathcal{H}_W$ left invertible (LI).
- Properties of filter bank controlled by patch extraction and by W
  - Shape of patches $\rightarrow$ shape of filters
  - Rows of W $\rightarrow$ channels of filter bank

Sparsifying Transforms as Filter Banks for Maximally Overlapping Patches



- $\mathrm{vec}(WX) = \mathcal{H}_W x$
- Defn: $\mathcal{H}_W$ is perfect reconstruction (PR) if $\mathcal{H}_W$ left invertible (LI).
- Properties of filter bank controlled by patch extraction and by W
  - Shape of patches $\rightarrow$ shape of filters
  - Rows of W $\rightarrow$ channels of filter bank
  - W is LI $\Rightarrow \mathcal{H}_W$ is PR

Sparsifying transforms as filter banks

- Take away: Existing transform learning algorithms learn perfect reconstruction filter banks!
- ... But, requiring W to be LI is stronger than requiring $\mathcal{H}_W$ to be PR!
- Two questions:
  1. Do we benefit by requiring $\mathcal{H}_W$ to be PR and relaxing the LI condition on W?
  2. Can we find an efficient algorithm to learn such an $\mathcal{H}_W$?

Sparsifying transforms as filter banks

- Take away: Existing transform learning algorithms learn perfect reconstruction filter banks!
- ... But, requiring W to be LI is stronger than requiring $\mathcal{H}_W$ to be PR!
- Two questions:
  1. Do we benefit by requiring $\mathcal{H}_W$ to be PR and relaxing the LI condition on W?
  2. Can we find an efficient algorithm to learn such an $\mathcal{H}_W$?

Sparsifying transforms as filter banks

- Take away: Existing transform learning algorithms learn perfect reconstruction filter banks!
- ... But, requiring W to be LI is stronger than requiring $\mathcal{H}_W$ to be PR!
- Two questions:
  1. Do we benefit by requiring $\mathcal{H}_W$ to be PR and relaxing the LI condition on W?
  2. Can we find an efficient algorithm to learn such an $\mathcal{H}_W$?

Previous Work

- Connection between patch-based analysis operators and convolution previously known
- Convolution often used as a computational tool

The Key Property

- Each frequency must pass through at least one channel!

Diagonalization

$$C_W^H C_W = \Phi^H \operatorname{ddiag}\left(\left| \bar{\Phi} W^T \right|^2 \mathbf{1}_{N_c} \right) \Phi$$

Perfect Recovery Condition

$\mathcal{H}_W$ is PR $\Leftrightarrow$ each entry of $\left| \bar{\Phi} W^T \right|^2 \mathbf{1}_{N_c} > 0$

- Decouples the choice of number of channels N_c and patch size (support of transform) $K \times K$
- Especially attractive for high dimensional data

Learning a sparsifying filter bank

Learning Formulation

- Desiderata
  - Parameterize with few degrees of freedom
  - $\mathcal{H}_W x$ should be (approximately) sparse
  - $\mathcal{H}_W$ should be PR and well conditioned
  - No identically zero filters
  - No duplicated filters

Learning Formulation

- $\mathcal{H}_W x$ should be (approximately) sparse
- $\implies WX$ should be (approximately) sparse

$$F(W, Z, x) \triangleq \frac{1}{2}\|WX - Z\|_F^2 + \nu\|Z\|_0$$

Learning Formulation

- $\mathcal{H}_W$ should be PR and well conditioned
- Let ζ_i be an eigenvalue of $\mathcal{H}_W^H \mathcal{H}_W$

$$\sum_{i=1}^{N^2} f(\zeta_i) = \sum_{i=1}^{N^2} \frac{\zeta_i^2}{2} - \log {\zeta_i}^2$$

$$= 0.5 \sum_{i=1}^{N^2} \sum_{j=1}^{N_c} (\left|\bar{\Phi} W^T\right|^2)_{i,j} - \log \left(\sum_{j=1}^{N_c} (\left|\bar{\Phi} W^T\right|^2)_{i,j} \right)$$

Learning Formulation

- No identically zero filters

$$-\beta \sum_{j=1}^{N_c} \log\left(\|W_{j,:}\|_2^2\right)$$

Learning Formulation

- $\mathcal{H}_W$ should be PR and well conditioned
- No identically zero filters

$$J_1(W) = 0.5 \sum_{i=1}^{N^2} \sum_{j=1}^{N_c} (\left|\bar{\Phi} W^T\right|^2)_{i,j} - \log\left(\sum_{j=1}^{N_c} (\left|\bar{\Phi} W^T\right|^2)_{i,j} \right)$$
$$- \beta \sum_{j=1}^{N_c} \log\left(\|W_{j,:}\|_2^2 \right)$$

Learning Formulation

- No duplicated filters

$$J_2(W) = \sum_{1 \le i < j \le N_c} - \log \left(1 - \left(\frac{\langle W_{i,:}, W_{j,:} \rangle}{\|W_{i,:}\|_2 \|W_{j,:}\|_2} \right)^2 \right)$$

Learning Formulation

$$\min_{W,Z} \frac{1}{2}\|WX - Z\|_F^2 + \alpha J_1(W) + \gamma J_2(W) + \nu\|Z\|_0$$

Alternating minimization:

- $Z^{k+1} = \arg\min_Z \frac{1}{2}\|W^k X - Z\|_F^2 + \nu\|Z\|_0$
- $W^{k+1} = \arg\min_W \frac{1}{2}\|WX - Z^{k+1}\|_F^2 + \alpha J_1(W) + \gamma J_2(W)$

Application to Magnetic Resonance Imaging

Imaging Model

- Imaging Model: Undersampled Fourier measurements

$$y = \Gamma\Phi x + e$$

- $x \in \mathbb{R}^{N^2}$: Input image
- $\Phi \in \mathbb{C}^{N^2 \times N^2}$: DFT matrix
- $\Gamma \in \mathbb{C}^{M \times N^2}$: Row selection matrix
- $e \in \mathbb{C}^M$: Zero mean Gaussian noise

Image Reconstruction - Transform Blind Compressed Sensing

$$\min_{x,\mathcal{H}_W,z} \frac{1}{2}\|y - \Gamma\Phi x\|_2^2 + \lambda \left(\frac{1}{2}\|\mathcal{H}_W x - z\|_2^2 + \nu\|z\|_0 + \alpha J_1(\mathcal{H}_W) + \gamma J_2(\mathcal{H}_W) \right)$$

- Data fidelity
- Transform learning
- Solve using alternating minimization

Image Reconstruction - Transform Blind Compressed Sensing

$$\min_{x,\mathcal{H}_W,z} \frac{1}{2}\|y - \Gamma\Phi x\|_2^2 + \lambda\left(\frac{1}{2}\|\mathcal{H}_W x - z\|_2^2 + \nu\|z\|_0 + \alpha J_1(\mathcal{H}_W) + \gamma J_2(\mathcal{H}_W)\right)$$

- Data fidelity
- Transform learning
- Solve using alternating minimization

Image Reconstruction - Transform Blind Compressed Sensing

$$\min_{x,\mathcal{H}_W,z} \frac{1}{2}\|y - \Gamma\Phi x\|_2^2 + \lambda\left(\frac{1}{2}\|\mathcal{H}_W x - z\|_2^2 + \nu\|z\|_0 + \alpha J_1(\mathcal{H}_W) + \gamma J_2(\mathcal{H}_W)\right)$$

- Data fidelity
- Transform learning
- Solve using alternating minimization

Image Reconstruction - Transform Blind Compressed Sensing

$$\min_{x,\mathcal{H}_W,z} \frac{1}{2}\|y - \Gamma\Phi x\|_2^2 + \lambda\left(\frac{1}{2}\|\mathcal{H}_W x - z\|_2^2 + \nu\|z\|_0 + \alpha J_1(\mathcal{H}_W) + \gamma J_2(\mathcal{H}_W)\right)$$

- Data fidelity
- Transform learning
- Solve using alternating minimization

Image Reconstruction - Transform Blind Compressed Sensing

Image Update

$$\min_{x} \frac{1}{2}\|y-\Gamma\Phi x\|^2+\frac{\lambda}{2}\|\mathcal{H}_W x-z\|_2^2$$

Transform Update

$$\min_{W} \frac{1}{2}\|\mathcal{H}_W x-z\|_F^2+\alpha J_1(W)+\gamma J_2(W)$$

Sparse Code Update

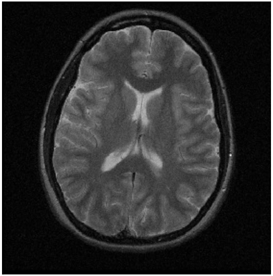
$$\min_{z} \frac{1}{2}\|\mathcal{H}_W x-z\|_2^2+\nu\|z\|_0$$

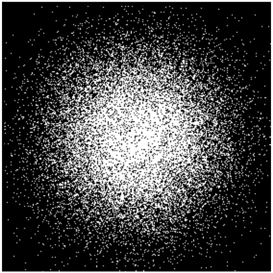
Experiments

- Synthetic MR data from magnitude image
- $\approx$ 5 fold undersampling
- Vary filter size $\&$ number of channels
- Compare against square patch-based transform learning:

$$\min_{W,x,Z} \frac{1}{2}\|y - \Gamma\Phi x\|_2^2 + \frac{\lambda}{2}\|WX - Z\| + \nu\|Z\|_0$$
$$+ \alpha\|W\|_F^2 - \beta \log\det W$$

- Solved using alternating minimization
- Initialized with DCT matrix

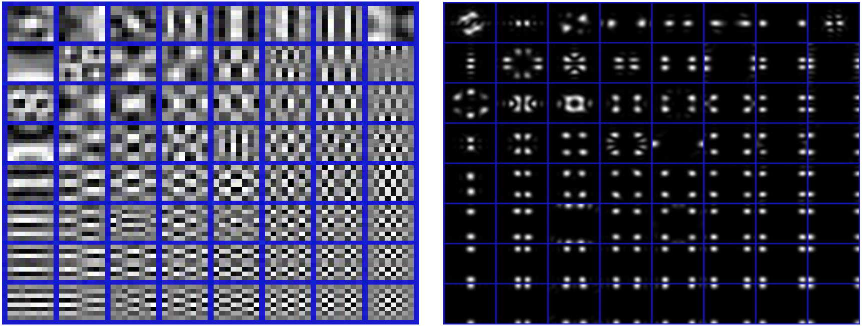




Reconstruction PSNR (dB)

σ / PSNR In	Filter Bank			Patch Based
	$N_c = 64$ $K = 8$	$N_c = 128$ $K = 8$	$N_c = 64$ $K = 12$	64×64
0 / 29.6	35.2	**35.2**	35.1	34.6
$\frac{10}{255}$ / 28.8	32.6	**32.7**	32.6	32.5
$\frac{20}{255}$ / 26.9	**31.6**	31.6	31.2	31.3

Learned filters 8×8



Conclusion

- New framework for learning filter bank sparsifying transforms
- Replace patch recovery conditions with image recovery
- Decouples number of channels from filter length
- Can outperform patch-based transform for MR reconstruction

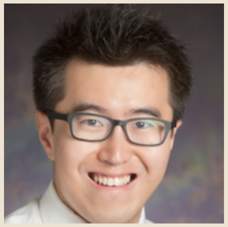
Summary

- We introduced several data-driven sparsifying transform adaptation techniques.
- Proposed learning methods
  - are highly efficient and scalable
  - enjoy good theoretical and empirical convergence behavior
  - are highly effective in many applications
- Highly promising results obtained using transform learning in denoising and compressed sensing.
- Papers and software available for download at http://transformlearning.csl.illinois.edu

Papers and software: http://transformlearning.csl.illinois.edu

Thank You!

Work with
- Sairprasad Ravishankar
- Bihan Wen
- Luke Pfister


Acknowledgements: NSF Grants CCF-1018660 & CCF-1320953
Andrew T. Yang Fellowship