Learning Sparsifying Transforms for Image Processing

Saiprasad Ravishankar and Yoram Bresler

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

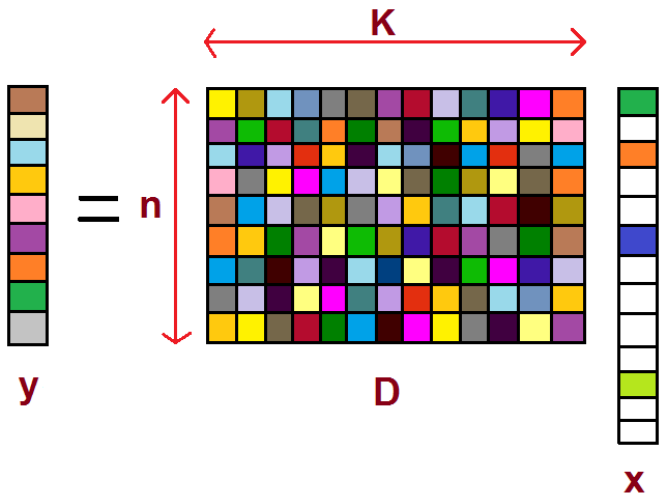
October 1, 2012

Outline

- Synthesis and Analysis models for sparse representation
- Transform model - A generalized Analysis model
- Synthesis/Analysis dictionary learning and drawbacks
- Transform learning
  - Formulations, Algorithms, and Properties
  - Numerical examples
- Conclusions and Future Work

Synthesis Model (SM) for Sparse Representation

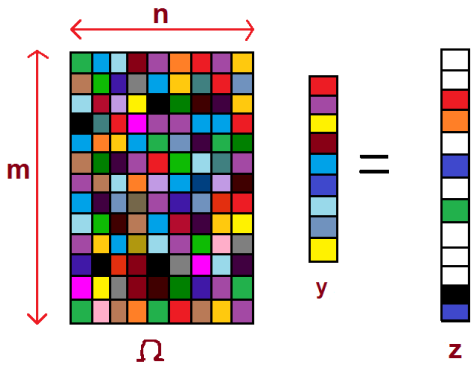
- Given a signal $y \in \mathbb{R}^n$, and dictionary $D \in \mathbb{R}^{n \times K}$, we assume $y = Dx$ with $\|x\|_0 = |supp(x)| \ll K$.



- Real world signals modeled as $y = Dx + e$, e is deviation term.
- Analytical overcomplete ($K > n$) dictionaries for sparse signal representation - Ridgelet, Contourlet, and Curvelet dictionaries.
- Given D, $\hat{x} = \arg\min \|y - Dx\|_2^2$ subject to $\|x\|_0 \leq s$, s is *sparsity level*. This is *synthesis sparse coding* - NP-hard problem!
- Greedy (e.g. Subspace Pursuit) and ℓ_1-relaxation algorithms (e.g. Lasso) exist but are computationally expensive.

Analysis Model (AM) for Sparse Representation

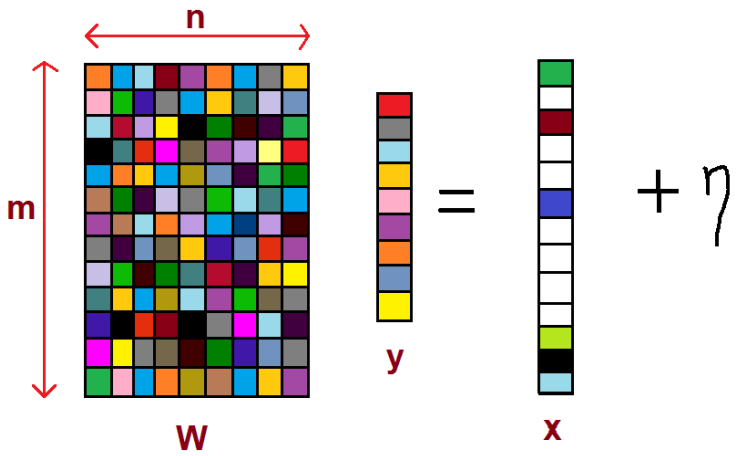
- (Strict) AM : Given a signal $y \in \mathbb{R}^n$, and analysis dictionary $\Omega \in \mathbb{R}^{m \times n}$, $\|\Omega y\|_0 \ll m$.



- Noisy Signal Analysis Model (NSAM) : $y = q + e$, $\Omega q = z$ sparse.
- Given Ω, $\hat{q} = \arg\min \|y - q\|_2^2$ subject to $\|\Omega q\|_0 \leq m - t$. This is *analysis sparse coding*, t is *co-sparsity level*.
- When Ω is square and full rank, $q = \Omega^{-1} z$, and the model is identical to the synthesis model $\Rightarrow$ finding z or q is NP-hard!
- Rubinstein et al. (2012) use a backward-greedy algorithm. Yaghoobi et al. (2012) solve Lagrangian version with ℓ_1-relaxation.
- However, the algorithms are computationally expensive.

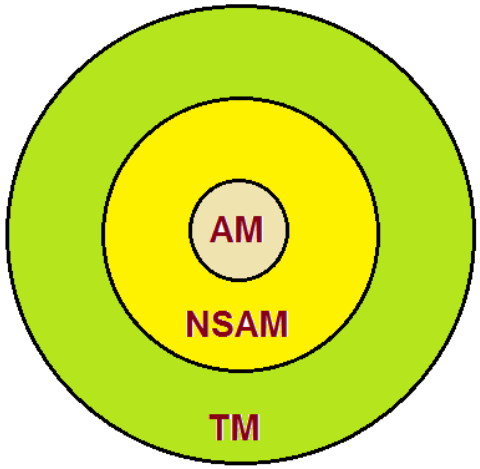
Transform Model (TM) for Sparse Representation

- 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 and images are approximately sparse in analytical transforms - Wavelets and DCT.
- Given W, $\hat{x} = \arg\min \|Wy - x\|_2^2$ subject to $\|x\|_0 \leq s$. This is *transform sparse coding*.
- x computed exactly by thresholding Wy. Sparse coding is cheap! - just like for classical transforms. Signal recovered as $W^\dagger x$.
- Sparsifying transforms used in compression (JPEG2000), etc.

Generality of Transform Model



- TM more general than AM : $y \in \mathrm{AM} \Rightarrow y \in \mathrm{TM}$.
  $y \in \mathrm{TM} \not\Rightarrow y \in \mathrm{AM}$.
- TM more general than the notion of compressibility of Wy.
- TM more general than NSAM : $y \in \mathrm{NSAM} \Rightarrow y \in \mathrm{TM}$.
  $y \in \mathrm{TM} \not\Rightarrow y \in \mathrm{NSAM}$.
  - $y = q + e$ with $Wq = z$ sparse $\Rightarrow Wy = Wq + We = z + We$.
  - However, $Wy = x + \eta$ with x sparse $\not\Rightarrow y = q' + e'$ with $Wq' = x$.
  - TM does not enforce $x \in R(W)$!

Summary of the Models

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

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

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

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

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

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

Learning Synthesis and Analysis Dictionaries

- Adapting dictionaries to a class of data advantageous in applications.
- Synthesis and Analysis dictionary learning formulations - typically non-convex and NP-hard.
- Approximate algorithms for Synthesis : MOD[1], K-SVD[2], online dictionary learning[3], etc.
- Heuristics for Analysis :
  - (Strict) Analysis: Sequential Minimal Eigenvalues[4], AOL[5].
  - Noisy Analysis: Analysis K-SVD[6], NAAOL[7].
- Algorithms Computationally expensive. No global convergence guarantees. Algorithms may converge to bad local minima.
- Yaghoobi et al. (2012) show learnt analysis operators denoise not much better than fixed finite difference operator.

[1] [Engan et al. '99] , [2] [Aharon et al. '06], [3] [Mairal et al. '09], [4] [Ophir et al. '11], [5] [Yaghoobi et al. '11], [6] [Rubinstein et al. '12], [7] [Yaghoobi et al. '12].

Transform Learning - Our Formulation

$$(\mathrm{P1}) \quad \min_{W,X} \|WY - X\|_F^2 \;\; s.t. \;\; \|X_i\|_0 \leq s \;\; \forall \; i$$

- $Y = [\, Y_1 \,|\, Y_2 \,|\, ... \,|\, Y_N \,] \in \mathbb{R}^{n \times N}$: matrix of training signals.
- $X = [\, X_1 \,|\, X_2 \,|\, ... \,|\, X_N \,] \in \mathbb{R}^{m \times N}$: matrix of sparse codes of Y_i.
- $\|WY - X\|_F^2$ is the sparsification error - measures deviation of data in transform domain from perfect sparsity at sparsity level s.
- Problem (P1) has the trivial solution $W = 0$, $X = 0$!
- Need to avoid transforms with zero rows, repeated rows, etc. - Transform should convey maximum information.
- For the square transform case ($m = n$), we reformulate (P1) as

$$(\mathrm{P2}) \quad \min_{W,X} \|WY - X\|_F^2 - \lambda \; \log\det W + \mu \, \|W\|_F^2$$
$$s.t. \;\; \|X_i\|_0 \leq s \;\; \forall \; i$$

Transform Learning - Our Formulation

$$(\mathrm{P2}) \quad \min_{W,X} \|WY - X\|_F^2 - \lambda \log\det W + \mu \|W\|_F^2$$
$$s.t. \quad \|X_i\|_0 \leq s \ \forall \ i$$

- $\mu, \lambda > 0$. $\log\det W$ restricts solution to full rank transforms.
- $\|W\|_F^2$ penalty prevents objective function from being unbounded from below. The cost in (P2) has explicit lower bound.
- (P2) is non-convex.
- (P2) requires $\det W > 0$: non-restrictive since sign of $\det W$ trivially changed by row permutation.
- Although $\det W > 0$ required, we do not enforce it, except to initialize minimization algorithms with W_0 such that $\det W_0 > 0$.

Properties of Formulation - Condition Number

$$(\mathrm{P2}) \quad \min_{W,X} \ \|WY - X\|_F^2 - \lambda \ \log\det W + \mu \, \|W\|_F^2$$
$$s.t. \quad \|X_i\|_0 \leq s \ \ \forall \ i$$

- (P2) 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 singular values $\sigma_i(\hat{W}) = \sqrt{\frac{\lambda}{2\mu}} \ \forall i \Rightarrow$ condition number $\kappa(\hat{W}) = 1$.
- Thus, (P2) favors both a low sparsification error and good conditioning.
- Minimizing the $-\lambda \ \log\det W + \mu \, \|W\|_F^2$ penalty encourages reduction of condition number.
- Moreover, with fixed $\frac{\mu}{\lambda}$, the solution to Problem (P2) is perfectly conditioned ($\kappa = 1$) in the limit of $\lambda \to \infty$.

Properties of Formulation - Equivalent Solutions

$$(\mathrm{P2}) \quad \min_{W,X} \ \|WY - X\|_F^2 - \lambda \ \log\det W + \mu \, \|W\|_F^2$$
$$s.t. \ \ \|X_i\|_0 \leq s \ \ \forall \ i$$

- Given a minimizer $(\tilde{W}, \tilde{X})$, we can form equivalent minimizers by
  - simultaneously permuting rows of $\tilde{W}$ and $\tilde{X}$.
  - pre-multiplying $\tilde{W}$ and $\tilde{X}$ with a diagonal matrix Γ with ± 1 entries.
  - Operations must retain the sign of the $\det(\tilde{W})$.

Algorithm

- Our algorithm for (P2) alternates between updating X and W.
- Sparse Coding Step solves (P2) with fixed W.

$$\min_X \|WY - X\|_F^2 \quad s.t. \quad \|X_i\|_0 \leq s \ \forall \ i \tag{4}$$

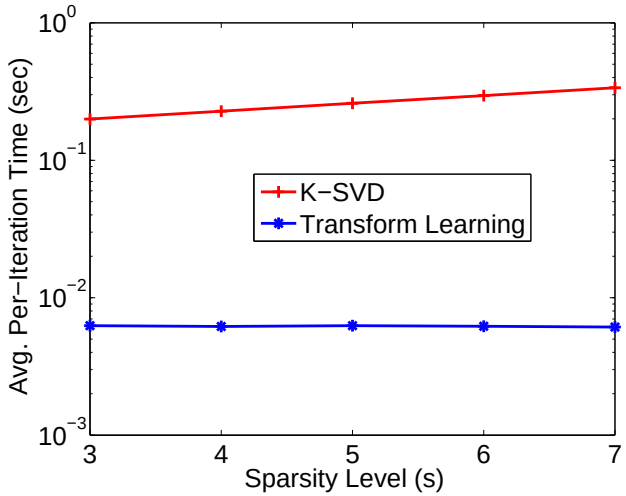
- **Easy** problem - Solution $\hat{X}$ computed exactly by **thresholding** WY, and retaining s largest coefficients in each column.
- Transform Update Step solves (P2) with fixed X.

$$\min_W \|WY - X\|_F^2 - \lambda \log\det W + \mu \|W\|_F^2 \tag{5}$$

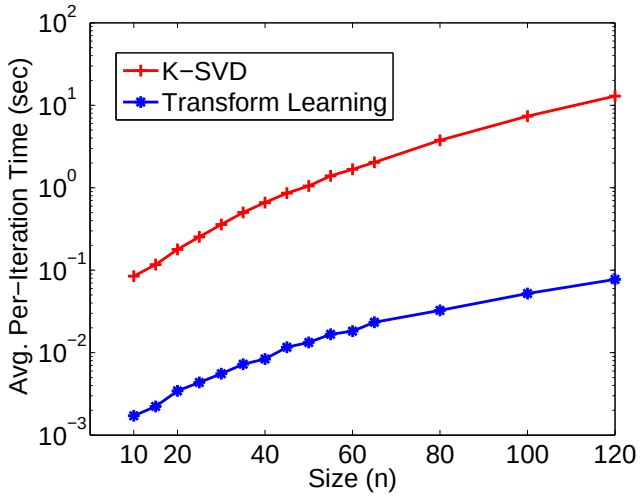
- Solved using Conjugate Gradients (CG).
- The cost function is monotone decreasing in each step. Moreover, since it is lower bounded, it converges.

Computational Cost and Run Time Advantages

- Cost per iteration of proposed algorithm: $O(Nn^2)$ for N training signals and $W \in \mathbb{R}^{n \times n}$.
- Synthesis/Analysis K-SVD cost per iteration : $O(Nn^3)$ – for square case. Cost dominated by sparse coding.
- Faster computations enable larger problem sizes and much lower run times for applications.



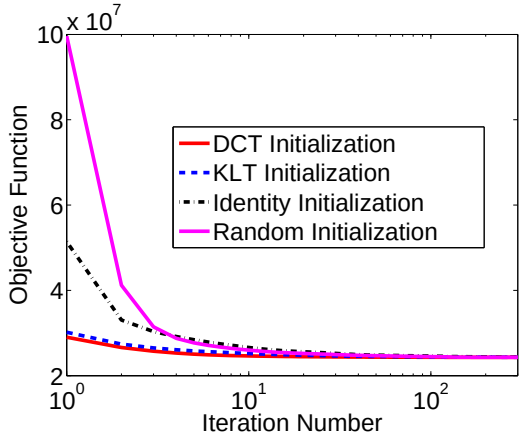
Per-iteration run time vs. s with fixed n



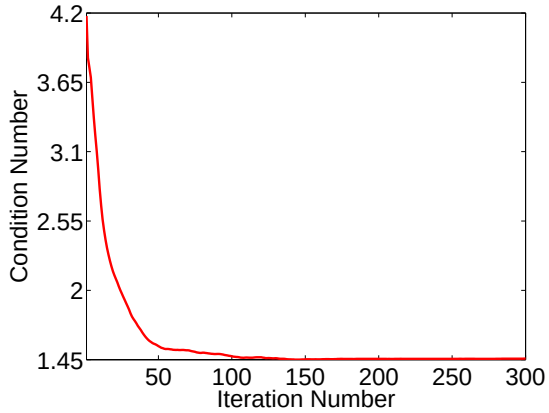
Per-iteration run time vs. n with $s \propto n$

Fast Convergence and Insensitivity to Initialization

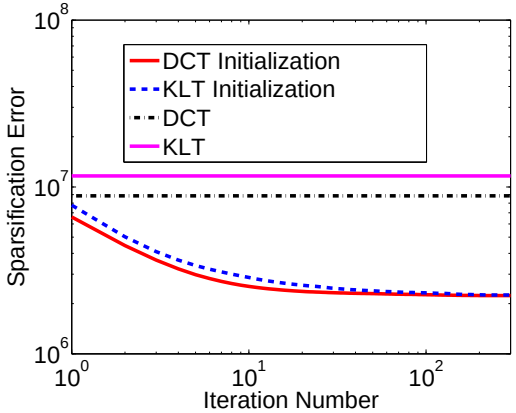
Barbara - 8×8 patches



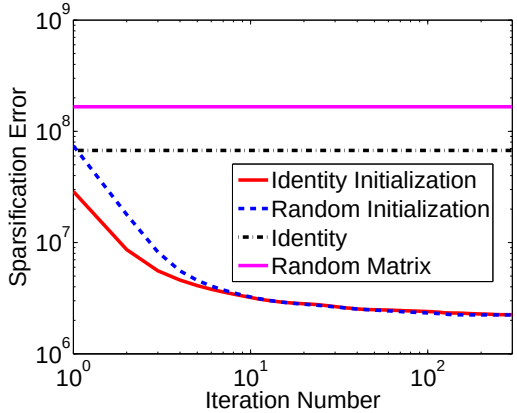
Objective Function



$\kappa(W)$ - Random Init



Sparsification Error ($s = 11$)



Sparsification Error ($s = 11$)

Learnt transforms are better than analytical transforms

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

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

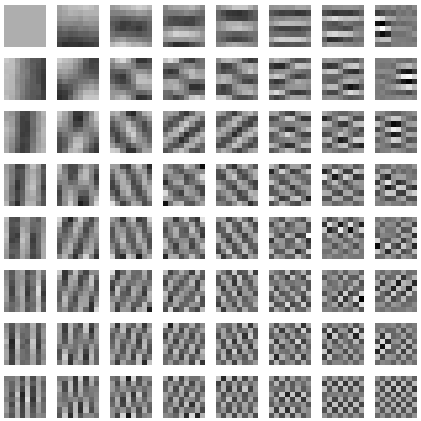
- recovery Peak Signal to Noise Ratio (rPSNR) defined in dB as

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

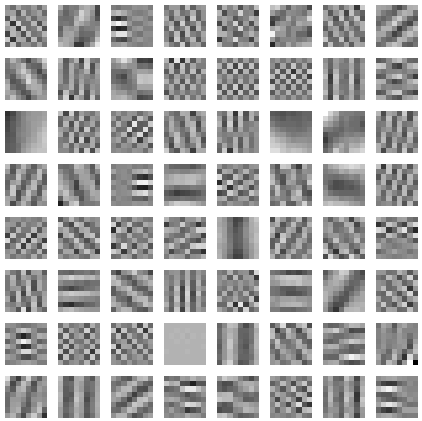
  P is # of image pixels.

- rPSNR measures the error in recovering the patches from their sparse codes as $\hat{Y} = W^{-1}X$.
- rPSNRs for the learnt W about 1.7 dB better than for DCT.

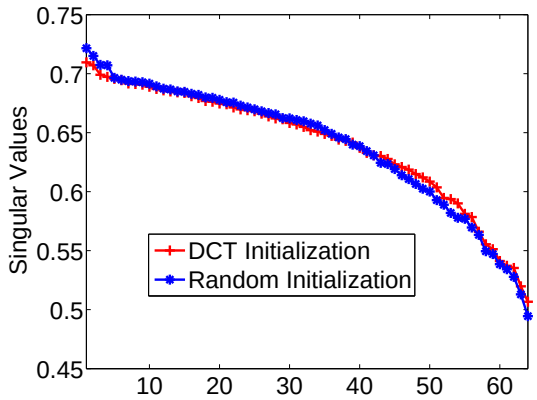
Non-Trivially Equivalent Transforms



Learnt W - DCT Init



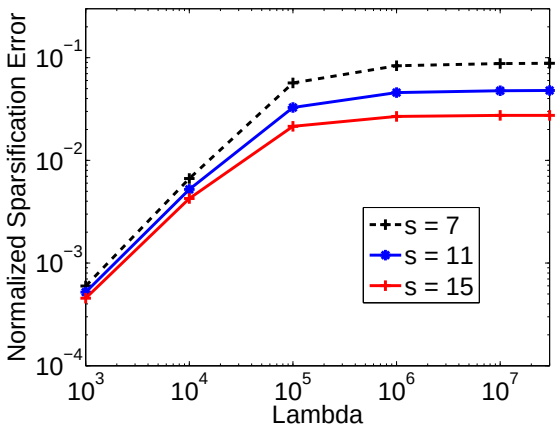
Learnt W - Random Init



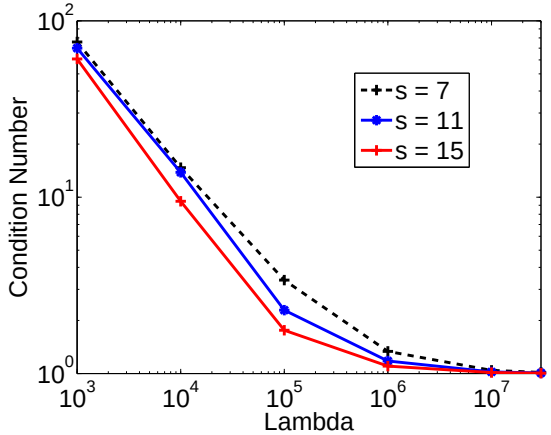
Singular Values

- Atoms/rows of transforms exhibit geometric and frequency like structures.
- Although they appear different, the different learnt transforms perform equally well.
- Additional application-specific performance criteria may be used to select between these essentially equivalent transforms.

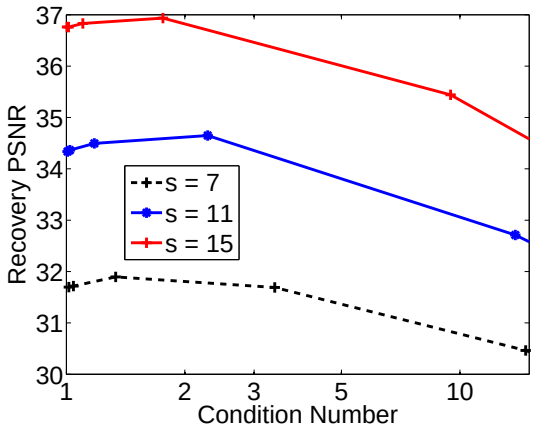
Behavior with respect to Parameters ($\mu = \lambda$)



NSE vs. λ



$\kappa(W)$ vs. λ

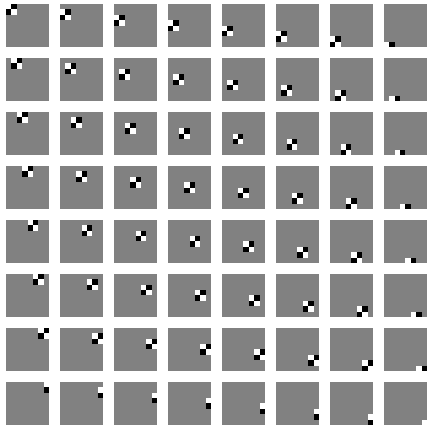


rPSNR vs. $\kappa(W)$

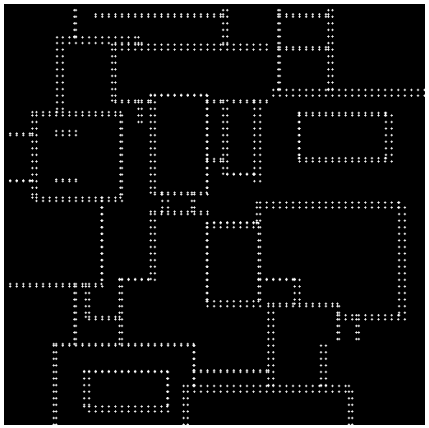
- For fixed s, λ enables complete control over $\kappa(W)$.
- Trade-off between NSE and $\kappa(W)$.
- rPSNR best at intermediate κ : e.g. $\kappa(W) = 2.3$ for $s = 11$.
- Natural images prefer well-conditioning rather than unit or bad κ.
- All metrics degrade as s decreases. Lower s values $\Rightarrow$ fewer degrees of freedom to represent data.

Piecewise-Constant Images

Image



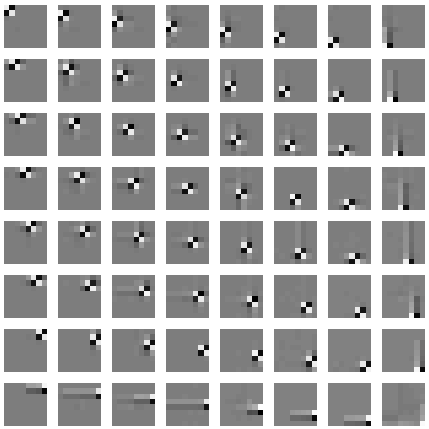
Finite difference (FD)
$\kappa(W) = 113.5$



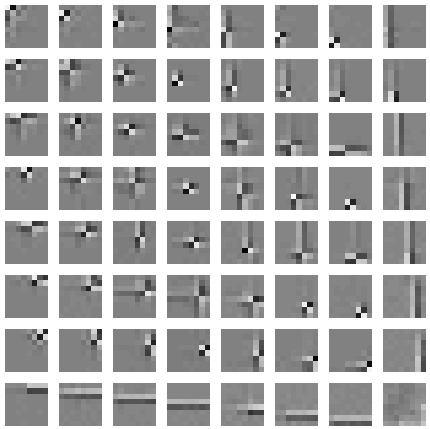
Sparse Result
$s = 5$

- 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.
- (P2) solved to learn transforms at various λ ($\mu = \lambda$), with $s = 5$.

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.

Conclusions

- We proposed formulations for learning sparsifying transforms that are highly effective for representing natural images.
- New regularization functional provides complete control of condition number - also applicable to synthesis and analysis learning.
- Proposed algorithms
  - encourage well-conditioning
  - low computational cost
  - insensitive to initialization
- Adaptive transforms provide significantly better representations than analytical ones.
- For more details, see "Learning Sparsifying Transforms", IEEE TSP, 2012 (to appear).
- Future work - tall transforms, structured transforms.