Adaptive Sparsifying Transforms for Signal and Image Processing

Saiprasad Ravishankar and Yoram Bresler

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

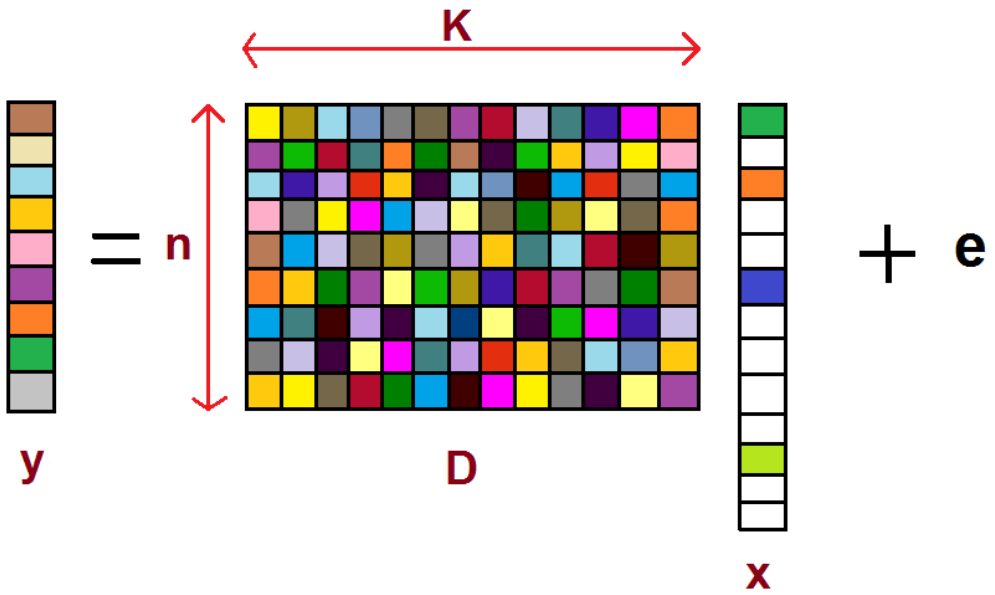
May 20, 2012

Outline

- Synthesis and Analysis models for sparse representations and their limitations
- Transform model for sparse representation
- Our transform learning formulation and properties
- Proposed algorithm and its properties
- Numerical examples and applications
- Conclusions

Synthesis Model for Sparse Representation

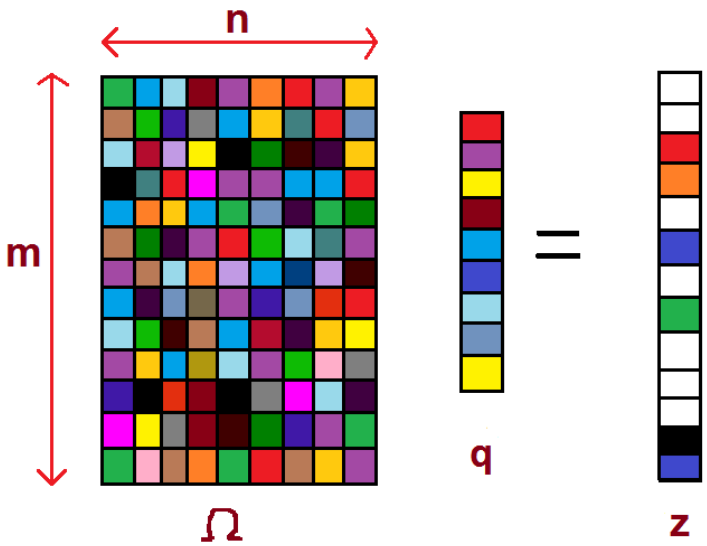
- Given a signal $y \in \mathbb{R}^n$, and dictionary $D \in \mathbb{R}^{n \times K}$, we model $y = Dx + e$ with $\|x\|_0 \ll K$ and $e \sim N(0, \sigma^2 I)$.



- The l_0 quasi norm counts the number of non-zeros in x.
- Analytical dictionaries for sparse signal/image representation - Ridgelet, Contourlet, and Curvelet dictionaries.
- Given D, find x by minimizing $\|y - Dx\|_2^2$ subject to $\|x\|_0 \leq s$. This is sparse coding - NP-hard problem!

Analysis Model for Sparse Representation

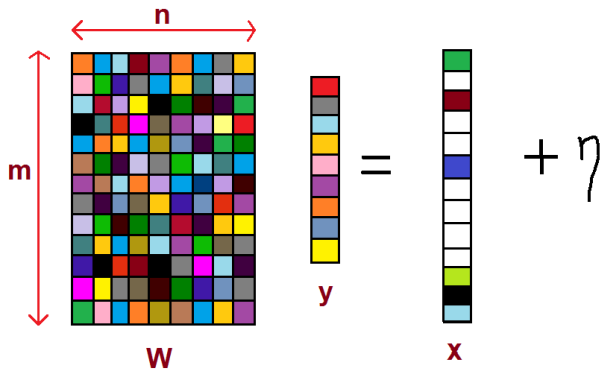
- Given a signal $y \in \mathbb{R}^n$, and analysis dictionary $\Omega \in \mathbb{R}^{m \times n}$, $y = q + e$ where $\Omega q = z$ with $\|z\|_0 \ll m$ and $e \sim N(0, \sigma^2 I)$.



- For given Ω, Rubinstein et al. (ICASSP '12) use backward-greedy algorithm for finding q that minimizes $\|y - q\|_2^2$ subject to $\|\Omega q\|_0 \leq s$.
- When Ω is square and has full rank, $q = \Omega^{-1} z$, and the model is identical to the synthesis model $\Rightarrow$ finding z or q is NP-hard!

Our Transform Model 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.
- These transforms are square, i.e., $m = n$, and orthonormal. When $m > n$ - 'Tall' transform.
- Given W, the sparse code x of sparsity s obtained by minimizing $\|Wy - x\|_2^2$ subject to $\|x\|_0 \leq s$.
- x computed exactly by thresholding Wy. Sparse coding is cheap!
- Sparsifying transforms used in compression (JPEG2000), etc.

Summary of the Models

- Synthesis model -

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

  - e in signal domain
  - finding x with given D - NP-hard.

- Analysis model -

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

  - e in signal domain
  - finding q with given Ω - NP-hard.

- Transform model -

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

  - η in transform domain
  - finding x with given W - easy.

Learning Synthesis and Analysis Dictionaries

- Adapting dictionaries to a class of data advantageous in applications.
- Synthesis dictionary learning formulations - non-convex and NP-hard.
- Approximate algorithms - MOD, K-SVD, online dictionary learning, etc. - may converge to bad local minima.
- For learning analysis dictionaries, Rubinstein et al. (ICASSP '12) propose a K-SVD type algorithm (analysis K-SVD).
- No convergence guarantees. Usefulness in applications unexplored.
- Yaghoobi et al. (ICASSP '12) learn analysis operators with l_1 norm sparsity penalties. However, learnt operators denoise worse than even a fixed finite difference operator.

Transform Learning - Our Formulation

$$(\mathrm{P1}) \ \min_{W,X} \ \|WY - X\|_F^2 \ \ s.t. \ \ \|X_i\|_0 \leq s \ \ \forall \ i \tag{4}$$

- $Y \in \mathbb{R}^{n \times N}$ - training matrix with signals as columns, $X \in \mathbb{R}^{m \times N}$ - sparse codes of Y with X_i as columns.
- $\|WY - X\|_F^2$ 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

$$(\mathrm{P2}) \ \min_{W,X} \ \|WY - X\|_F^2 - \lambda \ \log\det W + \mu \|W\|_F^2 \tag{5}$$
$$s.t. \ \ \|X_i\|_0 \leq s \ \ \forall \ i$$

Transform Learning - Our Formulation

- μ, $\lambda > 0$. $\log \det W$ restricts solution to full rank transforms.
- $\|W\|_F^2$ penalty prevents the objective function from being unbounded from below. The cost in (P2) has explicit lower bound.
- (P2) is non-convex.
- Only require $\det W > 0$ - non-restrictive since sign of $\det W$ can be trivially changed by row permutation.
- Although $\det W > 0$ is required, we do not enforce it, except to initialize algorithms with W_0 such that $\det W_0 > 0$.
- Log-barriers prevent optimization algorithms that minimize the objective from getting into infeasible regions.

Condition Number and Equivalent Solutions

- (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 $\kappa(\hat{W}) = 1$.
- (P2) favors both a low sparsification error and good conditioning.
- Moreover, with fixed $\frac{\mu}{\lambda}$, the solution to Problem (P2) is perfectly conditioned ($\kappa = 1$) in the limit of $\lambda \to \infty$.
- 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 Update Step solves (P2) with fixed W.

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

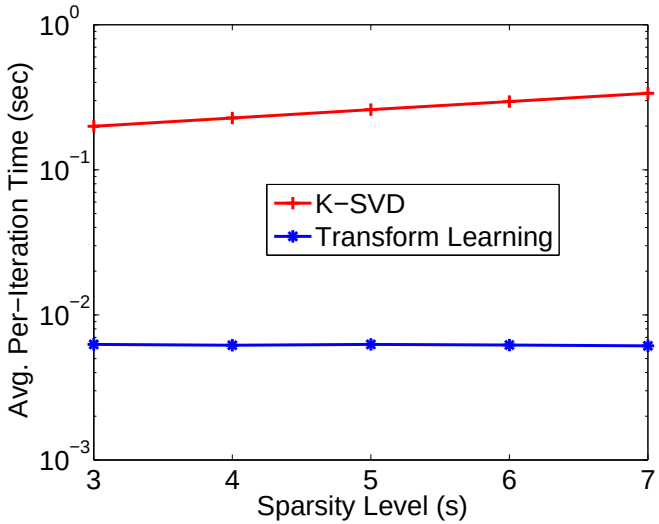
- Easy problem - Solution $\hat{X}$ computed exactly by thresholding WY, and retaining s largest coefficients in each column.
- Transform Update Step solves (P2) with fixed X.

$$\min_{W} \|WY - X\|_F^2 - \lambda \ \log\det W + \mu \|W\|_F^2 \tag{7}$$

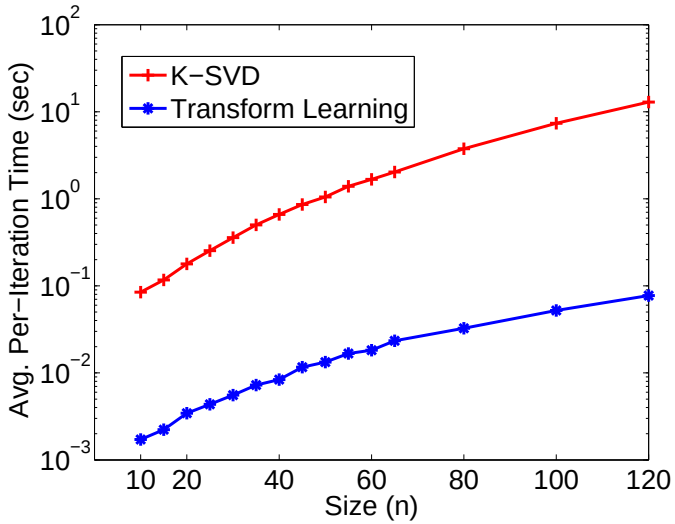
- Solved using iterative methods - Conjugate Gradients (CG) with Armijo step size rule guarantees reduction of cost function.
- Fixed step size rules also observed to work well and faster.
- The cost function is monotone decreasing. Moreover, since it is lower bounded, it converges.

Computational Cost and Run Time Advantages

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



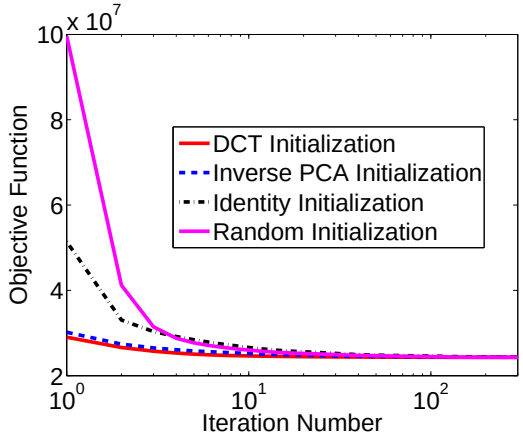
Run time vs. sparsity level



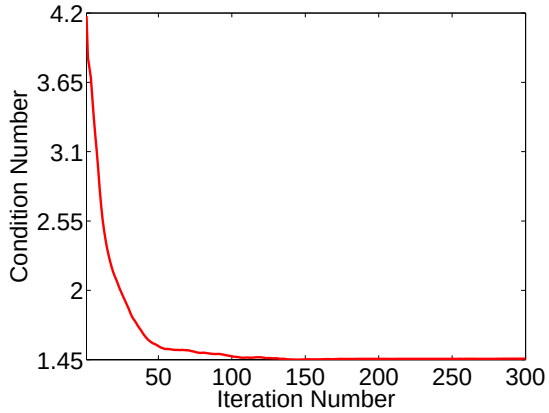
Run time vs. transform size

Example 1 - Convergence and Insensitivity to Initialization

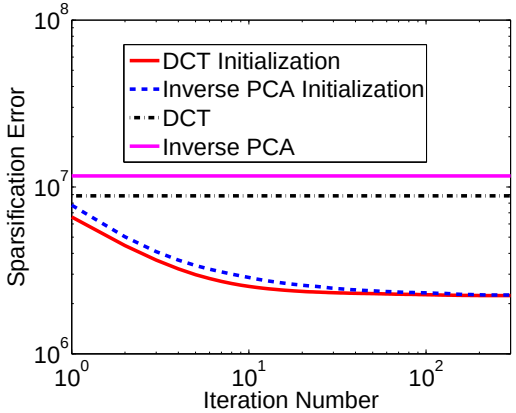
Barbara - 8×8 patches



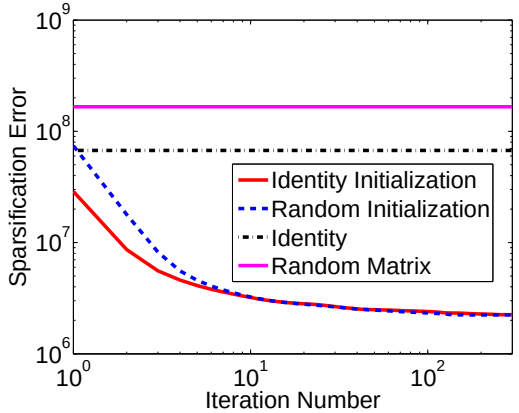
Objective Function



$\kappa(W)$ - 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.

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

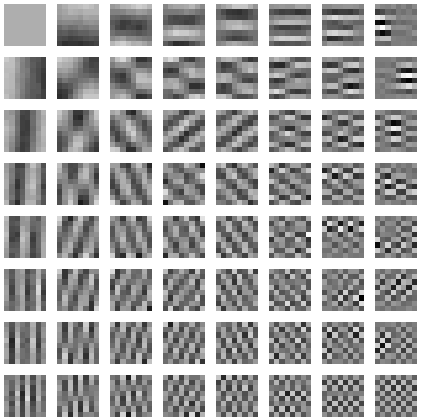
- recovery Peak Signal to Noise Ratio (rPSNR) defined in dB as

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

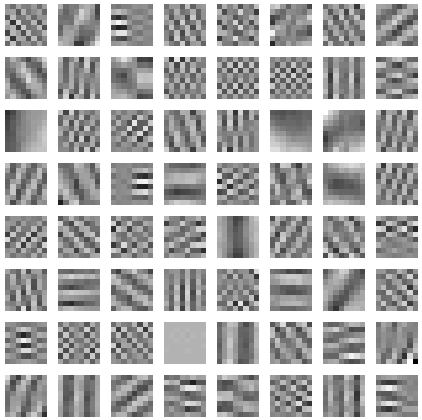
  P - # of image pixels. W^{-1} acts as equivalent synthesis dictionary.

- 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 DCT.

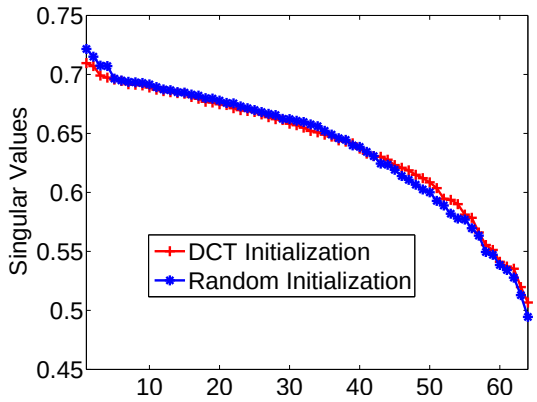
Non-Trivially Equivalent Transforms



Learnt W - DCT Init



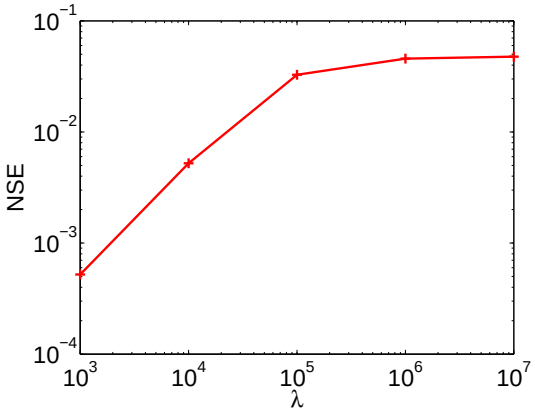
Learnt W - Random Init



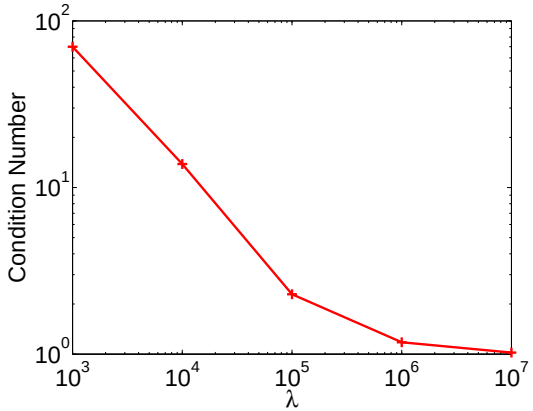
Singular Values

- Atoms 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.

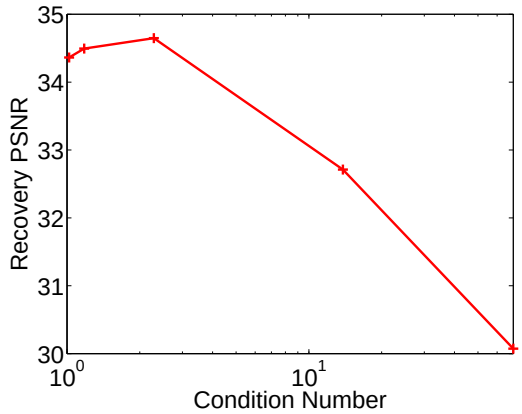
Behavior with respect to Parameters ($\mu = \lambda$, $s = 11$)



NSE vs. λ



$\kappa(W)$ vs. λ

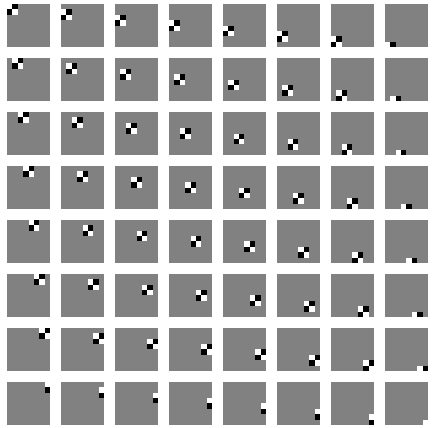


rPSNR vs. $\kappa(W)$

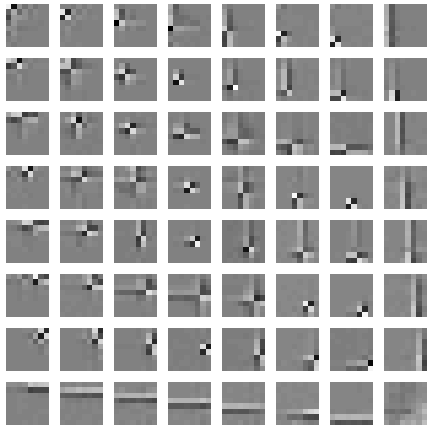
- λ enables complete control over condition number $\kappa(W)$. $\kappa \searrow$ as $\lambda \nearrow$.
- Trade-off between NSE and $\kappa(W)$. NSE $\nearrow$ as $\kappa \searrow$.
- rPSNR better at some intermediate conditioning: $\kappa(W) = 2.3$.
- This indicates that natural images prefer well-conditioning rather than unit-conditioning or bad conditioning.

Example 2 ($s = 5$) : Well-Conditioned Transforms do Well!

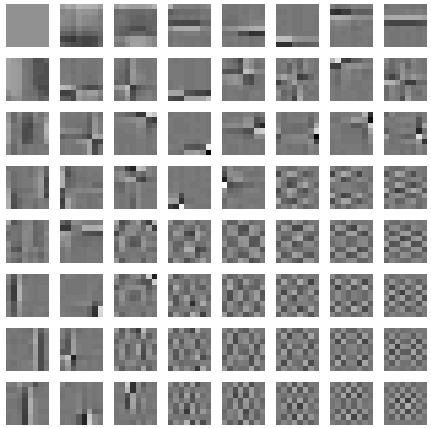
Image



Finite difference
$\kappa(W) = 113.5$



Finite difference init
$\kappa(W) = 5.02$



DCT init
$\kappa(W) = 3.92$

Application - Image Denoising

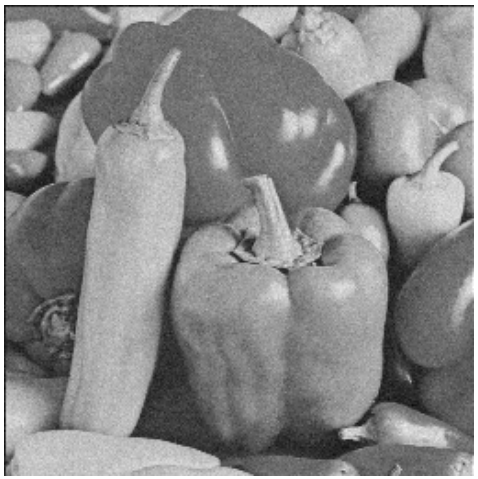
- Goal - estimate an image $x \in \mathbb{R}^P$ from its measurement $y = x + n$, that is corrupted by noise n.

$$(P4) \quad \min_{\{x_i, \alpha_i\}} \sum_{i=1}^{M} \|Wx_i - \alpha_i\|_2^2 + \tau \sum_{i=1}^{M} \|R_i\, y - x_i\|_2^2$$
$$s.t. \quad \|\alpha_i\|_0 \leq s_i \;\; \forall \; i \tag{8}$$

- $R_i \in \mathbb{R}^{n \times P}$ extracts i^{th} patch from y. M overlapping patches assumed.
- Assumption: Noisy $R_i\, y$ approximated by noiseless patch x_i that is sparsifiable.
- $\alpha_i \in \mathbb{R}^n$ - sparse code of x_i; $\tau \propto \frac{1}{\sigma}$ with σ - noise level .
- Problem (P4) can be solved efficiently by alternating minimization.
- W is learnt from patches of noisy image (fixed s).

Image Denoising Example

Original Peppers



Noisy ($\sigma = 10$)

PSNR = 34.45 dB
64×64 Transform

PSNR = 34.28 dB
64×256 Synthesis K-SVD

Conclusions

- We introduced new formulation for adaptive sparse modeling that is highly effective for natural images.
- New regularization functional provides complete control of condition number - also applicable to synthesis and analysis learning.
- Proposed algorithm
  - insensitive to initialization
  - encourages well-conditioning
  - Low computational cost
- Adaptive transforms provide significantly better representations than analytical ones.
- Denoising better than learnt overcomplete synthesis dictionaries.
- Future work - tall transforms, structured transforms, more applications.