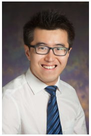
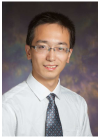
When Sparsity Meets Low-Rankness: Transform Learning With Non-Local Low-Rank Constraint For Image Restoration

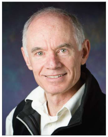
Bihan Wen, Yanjun Li and Yoram Bresler

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

March 9, 2017







Outline of Talk

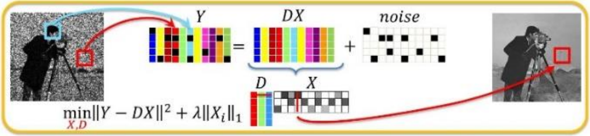
- Local sparsity v.s Non-local low-rankness
- **STROLLR** - Sparsifying TRansfOrm Learning and Low-Rank model
- STROLLR Image Restoration: formulation & algorihms
- Applications in image denoising and inpainting.

Local and Non-Local Image Properties

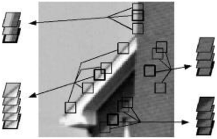
Image Properties: to differentiate **Signal** from corruptions.

1. Local properties - Sparsity
   - Natural patches are sparsifiable.

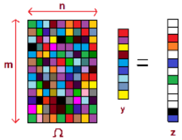
   - Synthesis model
   - Analysis model
   - => Transform model



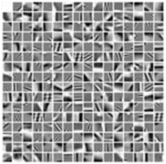
2. Non-local properties – Low-rankness
   - Image contains “similar” patches.
   - Group & process
   - Low-rank approximation



Sparse Coding



Transform Learning



Non-local Low-Rankness

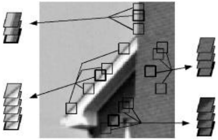
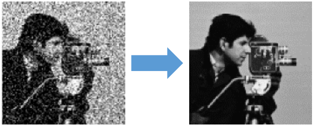
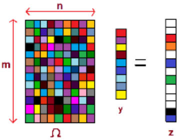


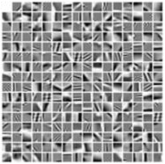
Image Restoration



Sparse Coding



Transform Learning



Non-local Low-Rankness

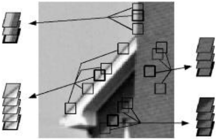
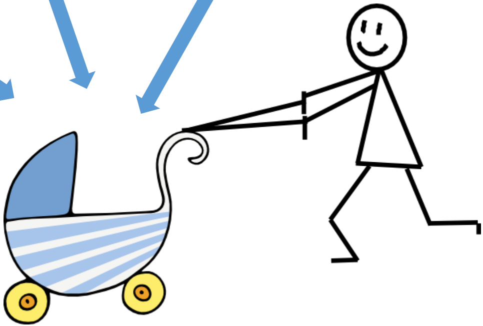


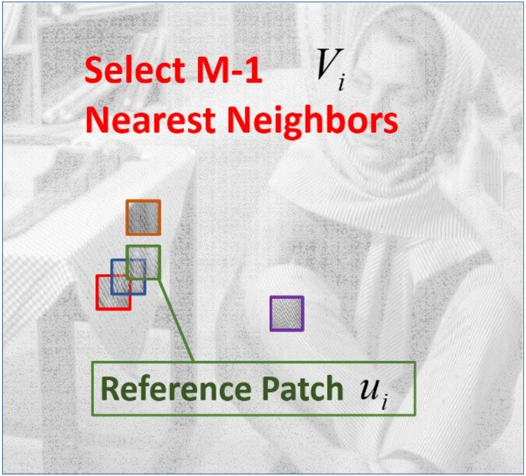
Image Restoration



STROLLR Modeling

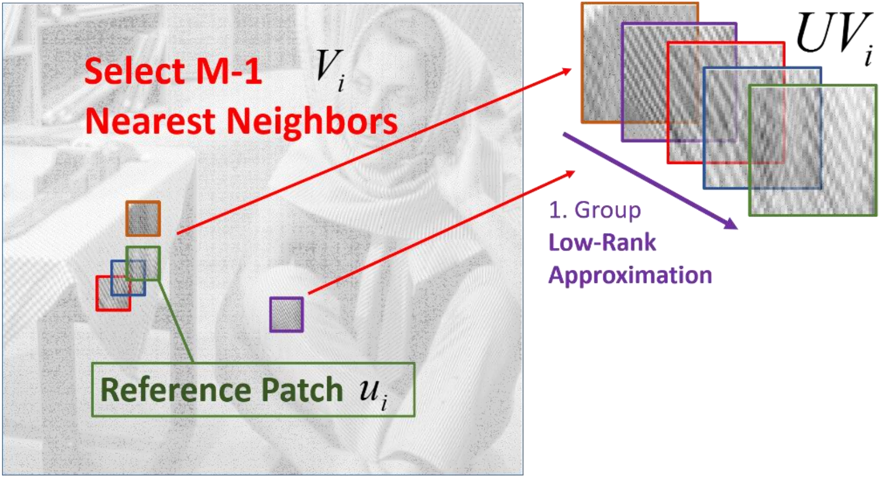
Example: Barbara

STROLLR Modeling



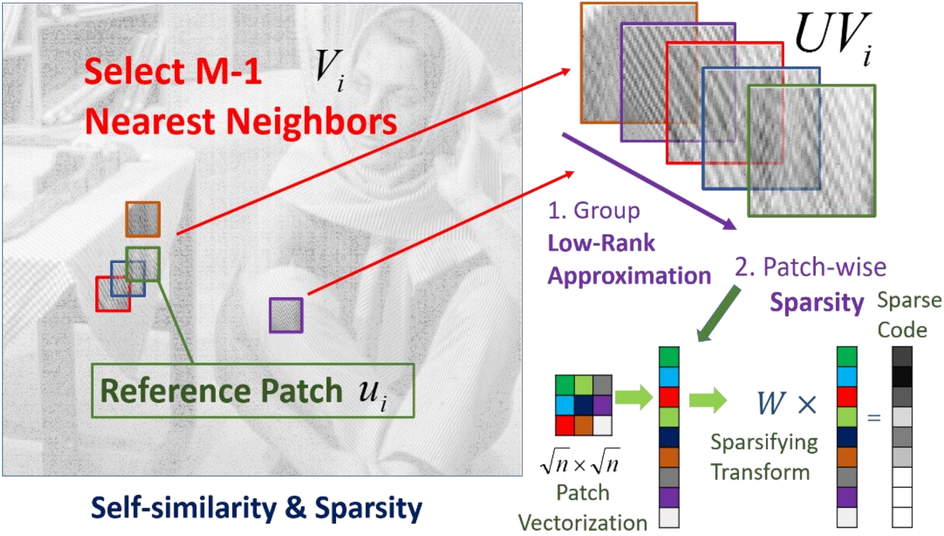
Example: Barbara

STROLLR Modeling



Example: Barbara

STROLLR Modeling

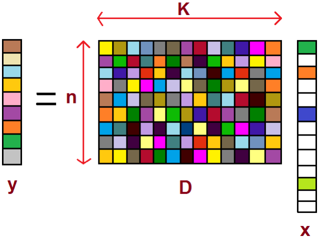


Self-similarity & Sparsity

Example: Barbara

Sparse Model: Synthesis Model

- **Synthesis Model (SM)**: Given synthesis dictionary $D \in \mathbb{R}^{n \times K}$, a signal $y \in \mathbb{R}^n$ satisfies $y = Dx$, with sparse x, i.e., $\|x\|_0 \ll$n.
  - General **SM**: $y = Dx + e$, where e is a small deviation term.

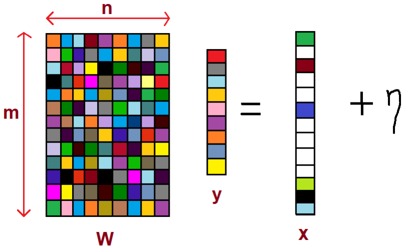


$D \in \mathbb{R}^{n \times K}$: Sparsifying transform $\quad x \in \mathbb{R}^K$ is sparse

- **Dictionary Learning**: popular sparse signal modeling approach.
- Sparse coding in SM is NP-hard!
  - Approximate methods: Greedy algorithms / l_1 norm relaxation.
  - Not efficient enough.

Sparse Model: Transform Model

- Transform Model - generalization of analysis model.
- Given transform $W \in \mathbb{R}^{m \times n}$, signal $y \in \mathbb{R}^n$ satisfies $Wy = x + \eta$, with $\|x\|_0 \ll m$, and a small deviation η in the **transform domain**.



$W \in \mathbb{R}^{m \times n}$: Sparsifying transform $\quad x \in \mathbb{R}^m$ is sparse

- **Transform sparse coding:**

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

- **Exact and cheap solution:** $\hat{x} = H_s(Wy)$ computed by thresholding Wy to the s largest magnitude elements (projection onto ℓ_0 ball).
    - A least squares signal estimate: $\hat{y} = W^{\dagger}\hat{x}$.

STROLLR Modeling

$$(\text{P1}) \quad \min_{\{W,X,\{D_i\}\}} \|W\,U - X\|_F^2 + \gamma_s^2 \|X\|_0 + \gamma_l \sum_{i=1}^{N} \left\{ \|U\,V_i - D_i\|_F^2 + \theta^2 \,\mathrm{rank}(D_i) \right\} \;\; s.t. \;\; W^T W = I_n$$

- $U = [\, u_1 \,|\, u_2 \,|\, \ldots.. \,|\, u_N \,] \in \mathbb{R}^{n \times N}$: matrix of image patch vectors.
- $X = [\, x_1 \,|\, x_2 \,|\, \ldots.. \,|\, x_N \,] \in \mathbb{R}^{n \times N}$: matrix of sparse codes of u_i's.
- $UV_i \in \mathbb{R}^{n \times M}$: matrix of patch-vectors via block matching (BM) with reference patch u_i.
- $D_i \in \mathbb{R}^{n \times M}$: the low-rank approximation of UV_i.
- **STROLLR Modeling:**
  - Local patch sparsity $\Longrightarrow$ Sparse coding for U with adaptive W.
  - Non-local low-rankness $\Longrightarrow$ Low-rank approximation of UV_i .

STROLLR Restoration

$$(\mathrm{P2}) \quad \min_{\{W, X, \{D_i\}, U\}} \|W\,U - X\|_F^2 + \gamma_s^2 \|X\|_0 + \gamma_f \sum_{i=1}^{N} \left\{ \|A_i\,u_i - y_i\|_2^2 \right\}$$
$$+ \gamma_l \sum_{i=1}^{N} \left\{ \|U\,V_i - D_i\|_F^2 + \theta^2 \,\mathrm{rank}(D_i) \right\} \quad s.t. \;\; W^T W = I_n$$

- Corrupted measurement $y_i = A_i u_i + h_i$
  - $h_i \in \mathbb{R}^n$: additive noise, and $A_i \in \mathbb{R}^{n \times n}$: corruption operator.
- $U = [\, u_1 \,|\, u_2 \,|\, \ldots.. \,|\, u_N \,] \in \mathbb{R}^{n \times N}$, where u_i is i-th overlapping image patch.
- $UV_i \in \mathbb{R}^{n \times M}$: block matching within $Q \times Q$ search window, centered at u_i.
- **Simple algorithm via Block Coordinate Descent:**
  - Exact and closed-form solution within each step.

STROLLR Algorithm

Four Major Steps:

1. Sparse Coding
2. Transform Update
3. Low-rank Approximation
4. Patch Restoration

STROLLR Image Restoration Algorithm Framework
Input: The corrupted image Y, the initial transform W_0.
Initialize: $\hat{W}_0 = W_0$, $\hat{U}_0 = [R_1 Y \mid R_2 Y \mid ... \mid R_N Y]$.
For $t = 1, 2, ..., T$ **Repeat**
1. **Sparse Coding:** $\hat{X}_t = H_{\gamma_s}(\hat{W}_{t-1} \hat{U}_{t-1})$.
2. **Transform Update:** Compute $S_t \Sigma_t G_t^T = \mathrm{SVD}(\hat{U}_{t-1} \hat{X}_t^T)$ as the full SVD, then update $\hat{W}_t = G_t S_t^T$.
3. **Low-rank Approximation for all** $i = 1, ...N$**:**
(a) Form $\{U_{t-1} V_i\}$ using BM.
(b) Compute SVD $\Phi_t \Omega_t \Psi_t^T = \mathrm{SVD}(U_{t-1} V_i)$.
(c) Update $\hat{D}_{i,t} = \Phi_t H_\theta(\Omega_t) \Psi_t^T$.
4. **Patch Restoration:** Restore the patch with closed-form solution for denoising or inpainting, to update $\hat{U}_t$
End
Aggregate $\{\hat{u}_i\}_{i=1}^N$ to restore the image

STROLLR Algorithm

$$\hat{X} = \underset{X}{\operatorname{argmin}} \; \|W\,U - X\|_F^2 + \gamma_s^2 \, \|X\|_0 \qquad (1)$$

Step 1: Sparse Coding: update X with fixed W.

- Standard transform-domain sparse coding:

Cheap hard thresholding: $\hat{X} = H_{\gamma_s}(W\,U)$.

$$\hat{W} = \underset{W}{\operatorname{argmin}} \; \|W\,U - X\|_F^2 \quad s.t. \;\; W^T W = I_n \qquad (2)$$

Step 2: Transform Update: update W with fixed X.

- Singular Value Decomposition: $S \Sigma G^T = U X^T$

Exact transform update: $\hat{W} = G S^T$.

STROLLR Algorithm

$$\hat{D}_i = \underset{D_i}{\operatorname{argmin}} \; \|U\,V_i - D_i\|_F^2 + \theta^2 \,\mathrm{rank}(D_i) \tag{3}$$

Step 3: Low-Rank Approximation: solve for D_i $\forall i$:

- Block matching to form UV_i.
- Apply SVD: $\Phi\Omega\Psi^T = UV_i$:

Low-rank Approximation: $\hat{D}_i = \Phi H_\theta(\Omega)\Psi^T$.

$$\hat{u}_i = \underset{u_i}{\operatorname{argmin}} \; \|W\,u_i - x_i\|_2^2 + \gamma_f \,\|A_i\,u_i - y_i\|_2^2 + \gamma_l \sum_{j \in C_i} \|u_i - D_{j,i}\|_2^2 \tag{4}$$

Step 4: Patch Reconstruction: solve for u_i, with fixed W, X, and $\{D_i\}$.

- Different restoration problems apply different A_i.

Image Inpainting and Denoising

$$\hat{u}_i = \{(1 + |C_i|\gamma_l)I_n + \gamma_f A_i\}^{-1} \left(W^T x_i + \gamma_l \sum_{j \in C_i} D_{j,i} + \gamma_f A_i y_i\right) \qquad (5)$$

Inpainting: A_i is diagonal binary matrix $\forall i$.

- Least squares solution to (4), with cheap inversion of diagonal matrix.

$$\hat{u}_i = \underset{u_i}{\operatorname{argmin}} \; \|W\, u_i - x_i\|_2^2 + \gamma_f \|u_i - y_i\|_2^2 + \gamma_l \sum_{j \in C_i} \|u_i - D_{j,i}\|_2^2 \qquad (6)$$

Denoising: special case when $A_i = I \; \forall i$.

- Simple solution as weighted average:

$$\hat{u}_i = (W^T x_i + \gamma_f y_i + \gamma_l \textstyle\sum_{j \in C_i} D_{j,i})/(1 + \gamma_f + |C_i|\gamma_l).$$

Application: Inpainting

- Example:

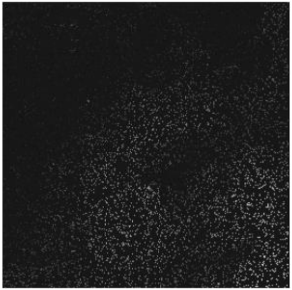


Image ***Face***

Corrupted measurement with 90% pixels missing

Inpainted image using ***STORLLR*** PSNR = 28.1 dB

Testing Images (size)

Barbara 512^2 | Lena 512^2 | Airport 1024^2 | Baboon 512^2 | Face 276^2 | Moon 256^2 | Elaine 512^2 | Sailboat 512^2 | Tank 512^2 | Plane 1024^2

Application: Inpainting

- Inpainting results:

  - Random pixel removal
  - Additive Gaussian noise
  - PSNR (dB)

Available pixels	σ	Smooth	LR	TL	STROLLR
20%	5	28.9	29.0	29.2	**29.3**
	10	27.4	28.2	28.2	**28.3**
	15	26.9	27.3	27.3	**27.4**
	20	25.5	**26.5**	26.2	**26.5**
10%	5	26.9	26.9	27.0	**27.1**
	10	26.0	26.3	26.3	**26.5**
	15	24.8	25.5	25.4	**25.6**
	20	23.7	24.7	24.5	**24.9**
Average		26.3	26.8	26.8	**27.0**

Testing Images (size)

Barbara 512^2 Lena 512^2 Airport 1024^2 Baboon 512^2 Face 276^2 Moon 256^2 Elaine 512^2 Sailboat 512^2 Tank 512^2 Plane 1024^2

Application: Denoising

Denoising PSNR table:

Images	σ	KSVD	LR	TL	BM3D	STROLLR
Barbara	5	38.1	38.4	38.1	38.3	**38.5**
	10	34.4	35.0	34.3	35.0	**35.1**
	15	32.3	33.1	32.1	33.1	**33.2**
	20	30.8	31.8	30.5	31.7	**31.9**
Elaine	5	37.3	37.2	37.2	36.7	**37.4**
	10	34.0	34.1	33.7	33.3	**34.2**
	15	32.3	32.5	32.1	32.2	**32.6**
	20	31.4	31.6	31.2	31.5	**31.7**
Average over 10 testing images		32.9	33.0	32.8	33.1	**33.2**

Testing Images (size)

Barbara 512^2

Lena 512^2

Airport 1024^2

Baboon 512^2

Face 276^2

Moon 256^2

Elaine 512^2

Sailboat 512^2

Tank 512^2

Plane 1024^2

Take-home message

- Prior works on local and non-local image structures
- Sparsifying TRansfOrm Learning and Low-Rank (**STROLLR**) combines both local sparsity and non-local structure in a single variational formulation.
- STROLLR modeling and restoration with efficient algorithms.
- Applications: Image Denoisng, inpainting, etc.

Thank you! Questions??