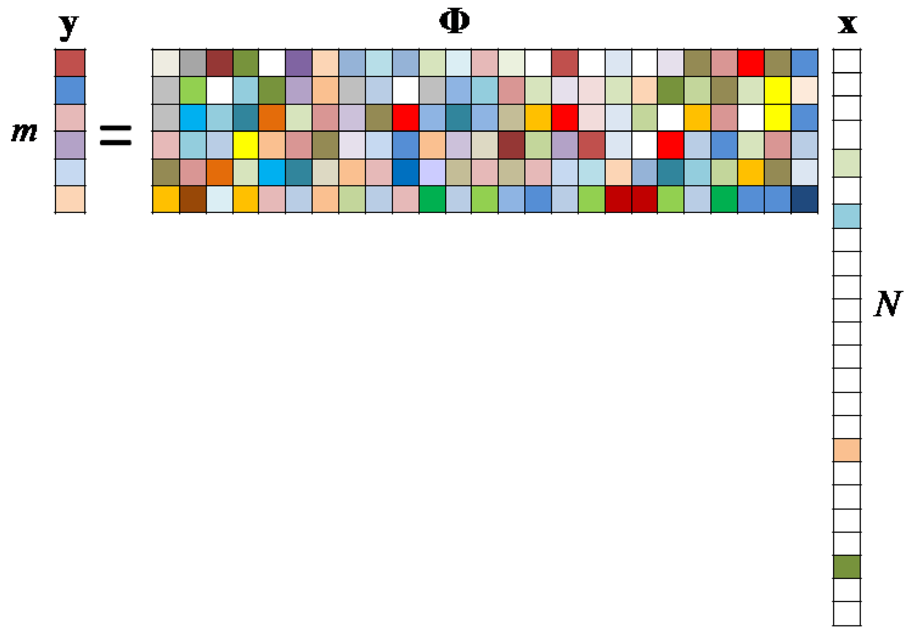
Fast Sparse Representation with Prototypes

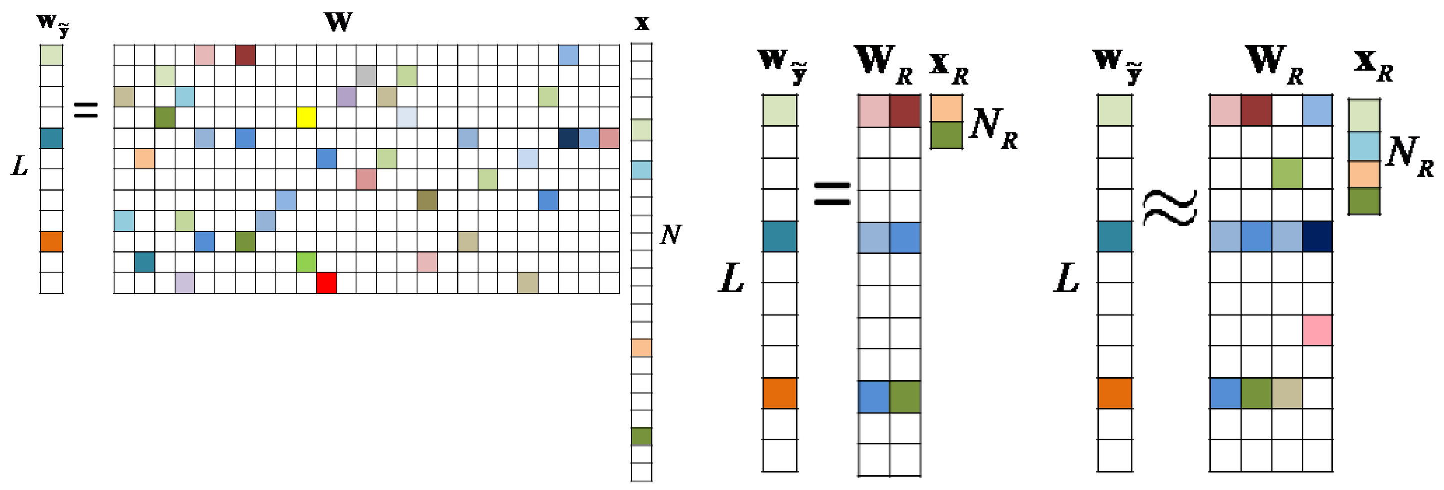
Jia-Bin Huang and Ming-Hsuan Yang
Electrical Engineering and Computer Science, University of California, Merced
jbhuang@ieee.org, mhyang@ieee.org

Overview

- Find the approximated sparse solution of the linear system $\boldsymbol{y} = \boldsymbol{Ax}$
- Original sparse coding problem: find a sparse solution from a *dense* matrix



- Transformed sparse coding problem: find a sparse solution from a *sparse* matrix



Applications

- Sparsity-based classification/clustering
- Image restoration (e.g., denoising, inpainting, demosaicking, super-resolution)
- Compressive sensing

Contributions

- Exploit the fact that signals can be well represented by a sparse linear combination of atom signals
- Reduce the original *dense* and *large* problem to a *sparse* and *small* problem

Algorithm

- **Original problem**

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\tilde{\mathbf{y}} - \mathbf{Fx}\|_2 \le \epsilon. \tag{1}$$

- **Dictionary learning from the K-SVD algorithm**

$$\min_{\mathbf{D},\mathbf{W}} \|\mathbf{F} - \mathbf{DW}\|_F^2 = \sum_{i=1}^{K}\sum_{j=1}^{n_i} \|\mathbf{f}_{i,j} - \mathbf{Dw}_{i,j}\|_2^2$$
$$\text{subject to} \quad \|\mathbf{w}_{i,j}\|_0 \le S_0, \tag{2}$$

- **Approximation process**

$$\tilde{\mathbf{y}} \approx \mathbf{Dw}_{\tilde{\mathbf{y}}} \approx \mathbf{DWx}. \tag{3}$$
$$\mathbf{Dw}_{\tilde{\mathbf{y}}} + \mathbf{e}_{\tilde{\mathbf{y}}} = \mathbf{DWx} + \mathbf{e}_{\mathbf{F}}\mathbf{x} \Longrightarrow \mathbf{D}(\mathbf{w}_{\tilde{\mathbf{y}}} - \mathbf{Wx}) = \mathbf{e}_{\mathbf{F}}\mathbf{x} - \mathbf{e}_{\tilde{\mathbf{y}}}. \tag{4}$$
$$\|\mathbf{D}(\mathbf{w}_{\tilde{\mathbf{y}}} - \mathbf{Wx})\|_2 \le (s+1)\epsilon. \tag{5}$$
$$\|\mathbf{z}\|_2 = \|\mathbf{w}_{\tilde{\mathbf{y}}} - \mathbf{Wx}\|_2 \le \frac{(s+1)\epsilon}{\sqrt{(1-\rho)}} = \tilde{\epsilon}. \tag{6}$$

- **Reduced ℓ_1 – *norm* minimization problem**

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \|\mathbf{w}_{\tilde{\mathbf{y}}} - \mathbf{Wx}\|_2 \le \tilde{\epsilon}. \tag{7}$$

Experimental Results

- **Face Recognition**

Figure: Sample images from the extended Yale database B

Table: Recognition accuracy and speed using the Extended Yale database B

Feature	Downsampled image		PCA	
Method	Acc (%)	Time (s)	Acc (%)	Time (s)
Original	93.78	20.08	95.01	13.17
Proposed	91.62	0.51	92.28	0.32
Speed-up	39.4		41.2	

- **Multi-view Object Recognition**

Table: Comparison on recognition speed and accuracy on the COIL-100

Number of view	8		16		8			16		
	Recognition accuracy				Execution time					
Feature used	Orig. (%)	Ours (%)	Orig. (%)	Ours (%)	Orig. (s)	Ours (ms)	Speed-up	Orig. (s)	Ours (ms)	Speed-up
Downsample	82.43	80.93	90.01	87.43	6.85	3.2	2140.6	52.73	3.9	13520.5
Downsample	75.08	74.28	84.75	84.00	4.13	3.9	1059.0	48.02	5.2	9234.6
PCA: 256	84.56	82.08	91.03	89.22	3.71	3.3	1124.2	29.58	3.8	7784.2
PCA: 100	81.23	79.23	90.58	87.72	2.54	3.6	705.6	21.00	5.6	3750.0

Experimental Results

- **Single Image Super-Resolution**

Ground-truth Bicubic [Yang et al. 08] Proposed

Table: Execution time and RMSE for sparse coding on four test images (scale factor = 3)

Image	Original		Proposed		
Method	RMSE	Time (s)	RMSE	Time (s)	Speedup
Girl	5.6684	17.2333	6.2837	1.5564	11.07
Flower	3.3649	14.9173	3.8710	1.3230	11.27
Pathenon	12.247	35.1163	13.469	3.1485	11.15
Raccoon	9.3584	27.9819	10.148	2.3284	12.02

- **Human Pose Estimation**

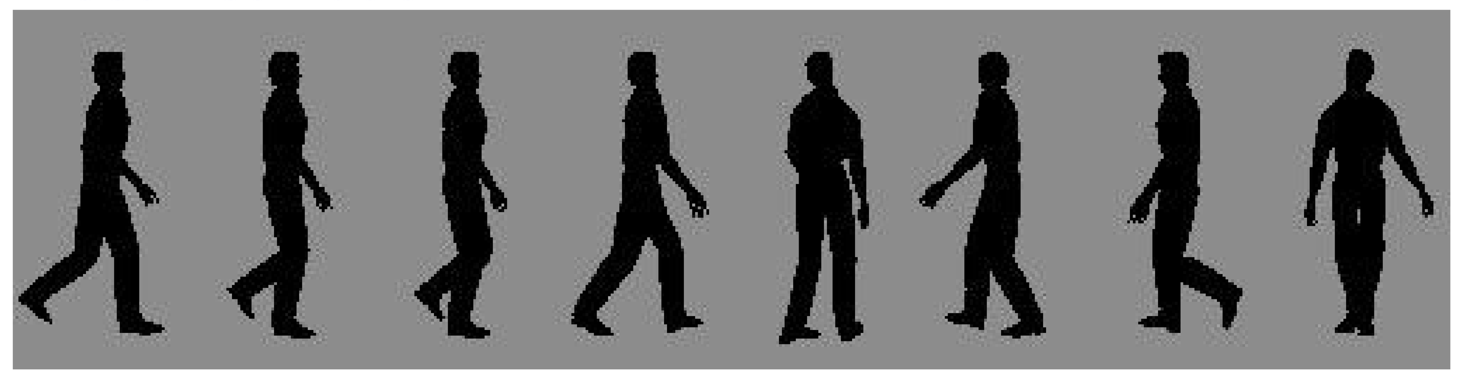


Figure: Sample images from the INRIA data set

Table: Comparison on pose estimation accuracy and speed under different number of prototypes using the INRIA data set

Number of coefficients	3	6	9	12	15	Original
Mean error (in degrees)	9.1348	7.9970	7.4406	7.2965	7.1872	6.6513
Execution time (in seconds)	0.0082	0.0734	0.3663	1.1020	2.3336	24.69
Speed-up	3011.0	336.4	67.4	22.4	10.6	