# Distributed Sensor Perception via Sparse Representation

**Allen Y. Yang**

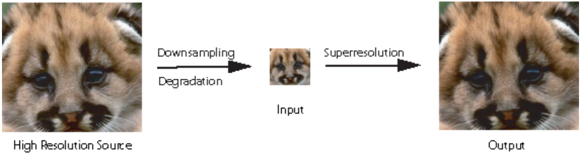
CVPR Tutorial on Sparse Representation and Applications
June 20, 2009

# Outline

1. Hands-on experience how to formulate sparse representation
2. How to implement fast and effective $\ell^1$-min programs in MATLAB
3. Problems
   1. Image Super-Resolution
   2. Wearable Action Recognition via Body Motion Sensors
   3. Distributed Object Recognition via Camera Sensor Networks

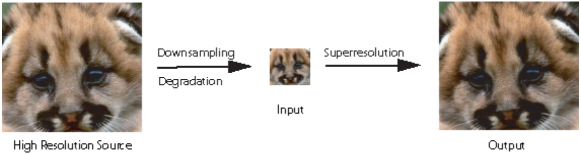
# Problem Definition: Example-based Super-resolution

- Problem: Given a **low-resolution input**, reconstruct a higher-resolution version of the image.



# Problem Definition: Example-based Super-resolution

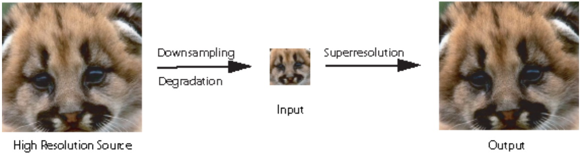
- Problem: Given a **low-resolution input**, reconstruct a higher-resolution version of the image.




- The problem is an inverse problem and **ill-posed**.
  1. Subpixel alignment
  2. Markov random field [Freeman 2000, Tipping 2003]
  3. Neighbor embedding [Chang 2004]

# Problem Definition: Example-based Super-resolution

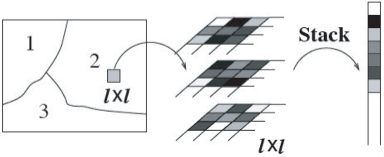
- Problem: Given a **low-resolution input**, reconstruct a higher-resolution version of the image.




- The problem is an inverse problem and **ill-posed**.
  1. Subpixel alignment
  2. Markov random field [Freeman 2000, Tipping 2003]
  3. Neighbor embedding [Chang 2004]
- **Assumption:** A high-resolution image library is provided (example-based).

# Our Approach: Sparse Linear Representation

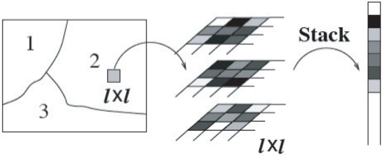
1. Randomly sample patches of the image database as **an overcomplete dictionary**




$$D_h = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n] \in \mathbb{R}^{D \times n}.$$

# Our Approach: Sparse Linear Representation

1. Randomly sample patches of the image database as **an overcomplete dictionary**



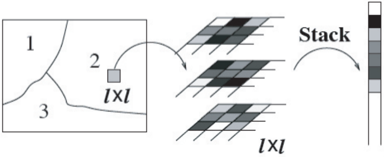

$$D_h = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n] \in \mathbb{R}^{D \times n}.$$

2. A high-resolution source patch $\mathbf{y}$ has a **sparse linear representation** [Perrett & Oram 1993, Olshausen & Field 1997]

$$\boxed{\mathbf{y} = D_h \cdot \mathbf{x}_0 \quad \text{with} \quad \|\mathbf{x}_0\|_0 \ll n.}$$

# Our Approach: Sparse Linear Representation

1. Randomly sample patches of the image database as **an overcomplete dictionary**




$$D_h = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n] \in \mathbb{R}^{D \times n}.$$

2. A high-resolution source patch **y** has a **sparse linear representation** [Perrett & Oram 1993, Olshausen & Field 1997]

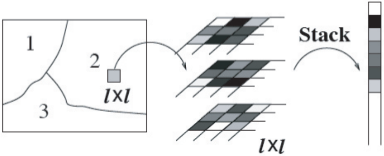
$$\boxed{\mathbf{y} = D_h \cdot \mathbf{x}_0 \quad \text{with} \quad \|\mathbf{x}_0\|_0 \ll n.}$$

3. Downsampling/blurring is a linear transformation $H$

$$D_l \doteq H \cdot D_h \in \mathbb{R}^{d \times n}.$$

# Our Approach: Sparse Linear Representation

1. Randomly sample patches of the image database as **an overcomplete dictionary**




$$D_h = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_n] \in \mathbb{R}^{D \times n}.$$

2. A high-resolution source patch $\mathbf{y}$ has a **sparse linear representation** [Perrett & Oram 1993, Olshausen & Field 1997]

$$\boxed{\mathbf{y} = D_h \cdot \mathbf{x}_0 \quad \text{with} \quad \|\mathbf{x}_0\|_0 \ll n.}$$

3. Downsampling/blurring is a linear transformation $H$

$$D_l \doteq H \cdot D_h \in \mathbb{R}^{d \times n}.$$

4. Input patch $\mathbf{z} \in \mathbb{R}^d$ inherits the same sparse representation

$$\mathbf{z} \doteq H\mathbf{y} = HD_h\mathbf{x}_0 = D_l\mathbf{x}_0.$$

# Local Patch Model from Sparse Representation

- **Known:** $D_h$, $H$, $D_l \doteq HD_h$, low-res image $I_l$ and patches $\mathbf{z}$.
  **Unknown:** $\mathbf{x}_0$ and high-res patches $\mathbf{y} = D_h\mathbf{x}_0$.
  **Output:** High-res image $I_h$ visually coherent as the collection $(\mathbf{y}_1, \cdots, \mathbf{y}_n)$.

# Local Patch Model from Sparse Representation

- **Known:** $D_h$, $H$, $D_l \doteq HD_h$, low-res image $I_l$ and patches $\mathbf{z}$.
  **Unknown:** $\mathbf{x}_0$ and high-res patches $\mathbf{y} = D_h\mathbf{x}_0$.
  **Output:** High-res image $I_h$ visually coherent as the collection $(\mathbf{y}_1, \cdots, \mathbf{y}_n)$.
- $\ell^1$-Minimization:

$$\mathbf{x}^* = \arg\min \|\mathbf{x}\|_1 \quad \text{subj to} \quad \|\mathbf{z} - D_l\mathbf{x}\|_2 \leq \epsilon_1$$
$$\mathbf{y}^* = D_h\mathbf{x}^*$$

# Local Patch Model from Sparse Representation

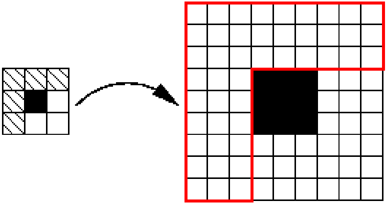
- **Known:** $D_h$, $H$, $D_l \doteq HD_h$, low-res image $I_l$ and patches $\mathbf{z}$.
  **Unknown:** $\mathbf{x}_0$ and high-res patches $\mathbf{y} = D_h\mathbf{x}_0$.
  **Output:** High-res image $I_h$ visually coherent as the collection $(\mathbf{y}_1, \cdots, \mathbf{y}_n)$.
- $\ell^1$-Minimization:

$$\mathbf{x}^* = \arg\min \|\mathbf{x}\|_1 \quad \text{subj to} \quad \|\mathbf{z} - D_l\mathbf{x}\|_2 \leq \epsilon_1$$
$$\mathbf{y}^* = D_h\mathbf{x}^*$$

- **Enforcing boundary compatibility**
  1. Define $\mathbf{w}$ as high-res patch that contains overlapping pixels
  2. Define $P$ as a mask that extracts pixels from previously reconstructed area

$$\mathbf{x}^* = \arg\min \|\mathbf{x}\|_1 \quad \text{subj to} \quad \|P\mathbf{w} - PD_h\mathbf{x}\|_2 \leq \epsilon_2$$

Mask P



# Enforcing Global Constraints

- Combine the two $\ell^1$-min in a single system:

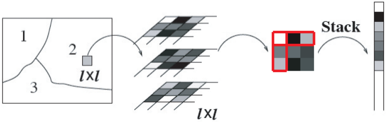
$$\mathbf{x}^* = \arg\min \|\mathbf{x}\|_1 \quad \text{subj to} \quad \| \begin{bmatrix} \mathbf{z} \\ \tilde{\mathbf{w}} \end{bmatrix} - \begin{bmatrix} D_l \\ PD_h \end{bmatrix} \mathbf{x}\|_2 \leq \epsilon$$

# Enforcing Global Constraints

- Combine the two $\ell^1$-min in a single system:

$$\mathbf{x}^* = \arg\min \|\mathbf{x}\|_1 \quad \text{subj to} \quad \| \begin{bmatrix} \mathbf{z} \\ \tilde{\mathbf{w}} \end{bmatrix} - \begin{bmatrix} D_l \\ PD_h \end{bmatrix} \mathbf{x}\|_2 \leq \epsilon$$

- Selecting **high-frequency** feature vectors: High-frequency components are more important to predict the target high-resolution image.



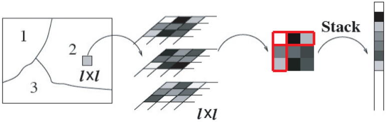

$$F\mathbf{z} = FD_l\mathbf{x} \quad \text{where} \quad f_1 = [-1, 0, 1], f_2 = [1, 0, -2, 0, 1], \ldots$$

# Enforcing Global Constraints

- Combine the two $\ell^1$-min in a single system:

$$\mathbf{x}^* = \arg\min \|\mathbf{x}\|_1 \quad \text{subj to} \quad \| \left[\begin{smallmatrix} \mathbf{z} \\ \tilde{\mathbf{w}} \end{smallmatrix}\right] - \begin{bmatrix} D_l \\ PD_h \end{bmatrix} \mathbf{x}\|_2 \leq \epsilon$$

- Selecting **high-frequency** feature vectors: High-frequency components are more important to predict the target high-resolution image.




$$F\mathbf{z} = FD_l\mathbf{x} \quad \text{where} \quad f_1 = [-1, 0, 1], f_2 = [1, 0, -2, 0, 1], \ldots$$

- Enforcing global constraint: Collect solutions $(\mathbf{y}_1, \cdots, \mathbf{y}_n) \Rightarrow I_0$.

$$I_h^* = \arg\min_I \|I - I_0\|_2 \quad \text{subj to} \quad HI = I_l.$$

Back-projection methods: [Irani 1993, Capel 2001]

# Visualization

Input

Orignial

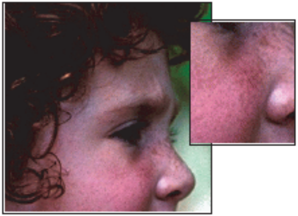
Neighbor Embedding

Sparse Representation

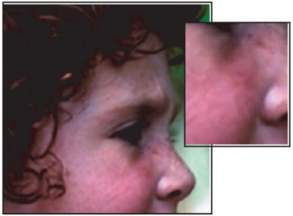
# Visualization

Input

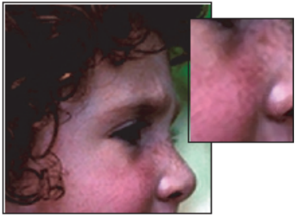
Orignial



Neighbor Embedding



Sparse Representation



# Comparison: Root Mean Square Error

| Image | | Bicubic | Neighborhood embedding | Our method |
|---|---|---|---|---|
| Flower |  | 3.51 | 4.20 | ***3.23*** |
| Girl |  | 5.90 | 6.66 | ***5.61*** |
| Parthenon |  | 12.74 | 13.56 | ***12.25*** |
| Raccoon |  | 9.74 | 9.85 | ***9.19*** |

**Reference:**

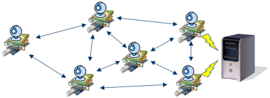
J. Yang, *et al.*, *Image Super-Resolution as Sparse Represetation of Raw Image Patches*, CVPR, 2008.

# Sensing and Perception in Resource-Constrained Distributed Networks

### Centralized Perception

| | |
|---|---|
| **Up:** | powerful processors |
| **Up:** | unlimited memory |
| **Up:** | unlimited bandwidth |
| **Down:** | single modality |

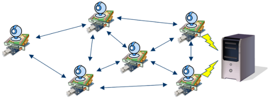
### Distributed Perception



| | |
|---|---|
| **Down:** | mobile processors |
| **Down:** | limited onboard memory |
| **Down:** | band-limited communications |
| **Up:** | distributed, multi-modality |

# Sensing and Perception in Resource-Constrained Distributed Networks

| Centralized Perception | |
|---|---|
|  | |
| **Up:** | powerful processors |
| **Up:** | unlimited memory |
| **Up:** | unlimited bandwidth |
| **Down:** | single modality |

| Distributed Perception | |
|---|---|
|  | |
| **Down:** | mobile processors |
| **Down:** | limited onboard memory |
| **Down:** | band-limited communications |
| **Up:** | distributed, multi-modality |

## If the total network is greater than the sum of its parts?

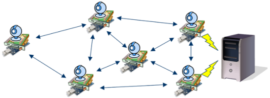
Question I: Recognize and track multiple **human faces** in a crowded camera network?

# Sensing and Perception in Resource-Constrained Distributed Networks

Centralized Perception

| | |
|---|---|
| **Up:** | powerful processors |
| **Up:** | unlimited memory |
| **Up:** | unlimited bandwidth |
| **Down:** | single modality |

Distributed Perception



| | |
|---|---|
| **Down:** | mobile processors |
| **Down:** | limited onboard memory |
| **Down:** | band-limited communications |
| **Up:** | distributed, multi-modality |

If the total network is greater than the sum of its parts?

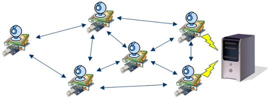
Question I: Recognize and track multiple **human faces** in a crowded camera network?
Question II: **Multiple-view super-resolution** using low-bandwidth camera sensors?

# Sensing and Perception in Resource-Constrained Distributed Networks

## Centralized Perception

| | |
|---|---|
| **Up:** | powerful processors |
| **Up:** | unlimited memory |
| **Up:** | unlimited bandwidth |
| **Down:** | single modality |

## Distributed Perception



| | |
|---|---|
| **Down:** | mobile processors |
| **Down:** | limited onboard memory |
| **Down:** | band-limited communications |
| **Up:** | distributed, multi-modality |

### If the total network is greater than the sum of its parts?

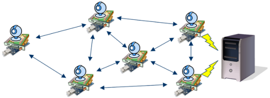
Question I: Recognize and track multiple **human faces** in a crowded camera network?
Question II: **Multiple-view super-resolution** using low-bandwidth camera sensors?
Question III: Use distributed physiological sensors to persistently **monitor patients**?

# Sensing and Perception in Resource-Constrained Distributed Networks

Centralized Perception

| | |
|---|---|
| **Up:** | powerful processors |
| **Up:** | unlimited memory |
| **Up:** | unlimited bandwidth |
| **Down:** | single modality |

Distributed Perception

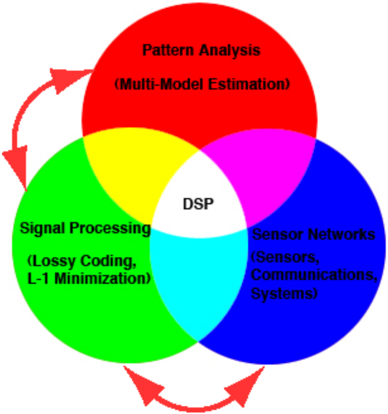


| | |
|---|---|
| **Down:** | mobile processors |
| **Down:** | limited onboard memory |
| **Down:** | band-limited communications |
| **Up:** | distributed, multi-modality |

## If the total network is greater than the sum of its parts?

Question I: Recognize and track multiple **human faces** in a crowded camera network?
Question II: **Multiple-view super-resolution** using low-bandwidth camera sensors?
Question III: Use distributed physiological sensors to persistently **monitor patients**?
Question IV: Use mobile sensors to control the formation of **air/ground vehicles**?

# Distributed Sensing and Perception (DSP)



# DexterNet: A Wearable Body Sensor Platform

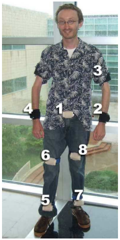
# Wearable Action Recognition

Goals

- Persistently monitor patient activities
- Alert dangerous actions
- Important social information for preventive healthcare in larger scale

Architecture

- 8 sensors distributed on human body.
- Location of the sensors are given and fixed.
- Each sensor carries triaxial accelerometer and biaxial gyroscope.



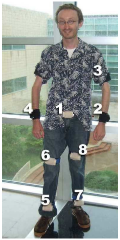
# Wearable Action Recognition

Goals

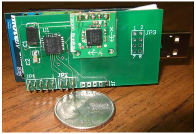
- Persistently monitor patient activities
- Alert dangerous actions
- Important social information for preventive healthcare in larger scale

Architecture

- 8 sensors distributed on human body.
- Location of the sensors are given and fixed.
- Each sensor carries triaxial accelerometer and biaxial gyroscope.




Interested Actions: Walking, Running, Jumping, Turning, Going upstars/downstairs, ...



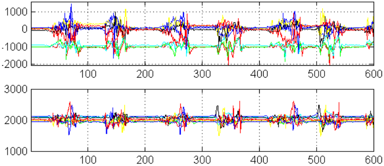
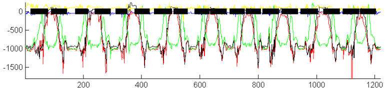



Figure: 8 *x*-axis accelerometers and *x*-axis gyroscopes for a *stand-kneel-stand* action sequence.

# Distributed Action Recognition

1. **Training samples** are segmented manually with correct labels.

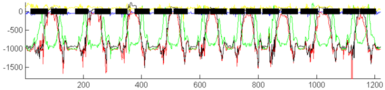



On each sensor node $i$, **normalize the vector form** (via stacking)

$$\mathbf{v}_i = [x(1), \cdots, x(h), y(1), \cdots, y(h), z(1), \cdots, z(h), \theta(1), \cdots, \theta(h), \rho(1), \cdots, \rho(h)]^T \in \mathbb{R}^{5h}$$

# Distributed Action Recognition

1. **Training samples** are segmented manually with correct labels.




On each sensor node $i$, **normalize the vector form** (via stacking)

$$\mathbf{v}_i = [x(1), \cdots, x(h), y(1), \cdots, y(h), z(1), \cdots, z(h), \theta(1), \cdots, \theta(h), \rho(1), \cdots, \rho(h)]^T \in \mathbb{R}^{5h}$$

2. **Full body motion**

$$\text{Training sample: } \mathbf{v} = \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_8 \end{pmatrix} \quad \text{Test sample: } \mathbf{y} = \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_8 \end{pmatrix} \in \mathbb{R}^{8 \cdot 5h}$$

3. **Sparse representation classifier**

$$\mathbf{y} = \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_8 \end{pmatrix} = \left( \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_8 \end{pmatrix}_1, \cdots, \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_8 \end{pmatrix}_n \right) \mathbf{x} = A\mathbf{x}.$$

# Local/Global Classifiers

## Distributed Sparse Representation

$$\begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_8 \end{pmatrix} = \left( \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_8 \end{pmatrix}_1, \cdots, \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_8 \end{pmatrix}_n \right) \mathbf{x} \Leftrightarrow \begin{cases} \mathbf{y}_1 = (\mathbf{v}_{1,1}, \cdots, \mathbf{v}_{1,n})\mathbf{x} \\ \vdots \\ \mathbf{y}_8 = (\mathbf{v}_{8,1}, \cdots, \mathbf{v}_{8,n})\mathbf{x} \end{cases}$$

# Local/Global Classifiers

## Distributed Sparse Representation

$$\begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_8 \end{pmatrix} = \left( \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_8 \end{pmatrix}_1, \cdots, \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_8 \end{pmatrix}_n \right) \mathbf{x} \Leftrightarrow \begin{cases} \mathbf{y}_1 = (\mathbf{v}_{1,1}, \cdots, \mathbf{v}_{1,n})\mathbf{x} \\ \vdots \\ \mathbf{y}_8 = (\mathbf{v}_{8,1}, \cdots, \mathbf{v}_{8,n})\mathbf{x} \end{cases}$$

- On each sensor node $i$: Choose **random projection** $R_i \in \mathbb{R}^{d \times 5h}$:

$$\tilde{\mathbf{y}}_i = R_i \mathbf{y}_i = R_i A_i \mathbf{x} = \tilde{A}_i \mathbf{x} \in \mathbb{R}^d$$

# Local/Global Classifiers

## Distributed Sparse Representation

$$\begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_8 \end{pmatrix} = \left( \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_8 \end{pmatrix}_1, \cdots, \begin{pmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_8 \end{pmatrix}_n \right) \mathbf{x} \Leftrightarrow \begin{cases} \mathbf{y}_1 = (\mathbf{v}_{1,1}, \cdots, \mathbf{v}_{1,n})\mathbf{x} \\ \vdots \\ \mathbf{y}_8 = (\mathbf{v}_{8,1}, \cdots, \mathbf{v}_{8,n})\mathbf{x} \end{cases}$$

- On each sensor node $i$: Choose **random projection** $R_i \in \mathbb{R}^{d \times 5h}$:

$$\tilde{\mathbf{y}}_i = R_i \mathbf{y}_i = R_i A_i \mathbf{x} = \tilde{A}_i \mathbf{x} \in \mathbb{R}^d$$

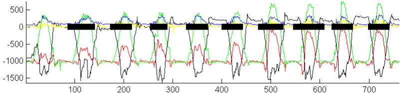
- Adaptive classification for a subset of active sensors (Suppose $1, \ldots, L$ at time $t$ and $h_i$)

  Define **global feature matrix** $R' = \begin{pmatrix} R_1 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & & \vdots \\ 0 & \cdots & R_L & \cdots & 0 \end{pmatrix}$:

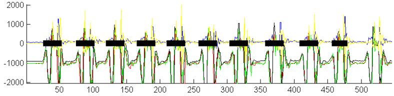
$$\begin{pmatrix} \tilde{\mathbf{y}}_1 \\ \vdots \\ \tilde{\mathbf{y}}_L \end{pmatrix} = R' \begin{pmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_8 \end{pmatrix} = R' \begin{pmatrix} A_1 \\ \vdots \\ A_8 \end{pmatrix} \mathbf{x} = R' A \mathbf{x}$$

# Experiment

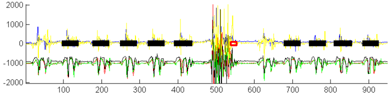
- Visualization



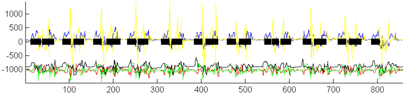

(a) Bend




(b) Jump



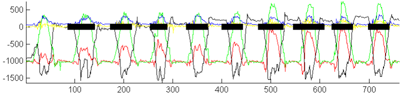

(c) Going downstairs



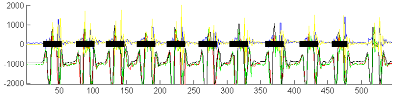

(d) Turning left/right

# Experiment

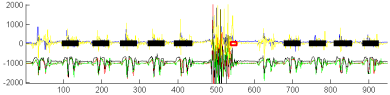
- Visualization



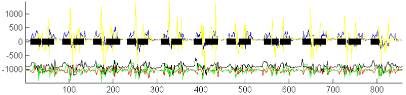
(a) Bend



(b) Jump



(c) Going downstairs

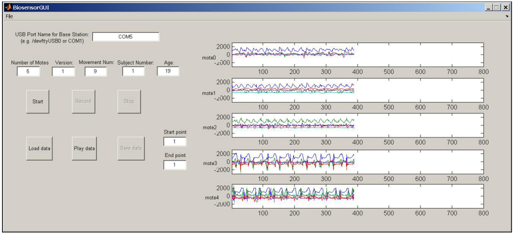


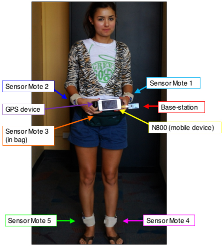
(d) Turning left/right

- Precision vs Recall

| Sensors | 2 | 7 | 2,7 | 1,2,7 | 1- 3, 7,8 | 1- 8 |
|---|---|---|---|---|---|---|
| Prec [%] | 89.8 | 94.6 | 94.4 | 92.8 | 94.6 | 98.8 |
| Rec [%] | 65 | 61.5 | 82.5 | 80.6 | 89.5 | 94.2 |

# Wearable Action Recognition Database (WARD)



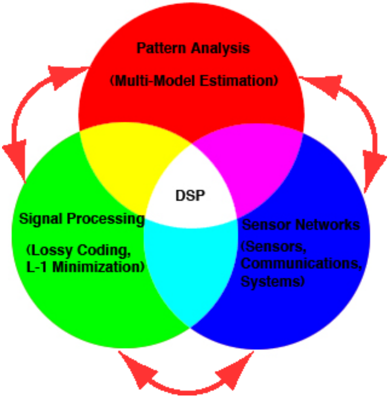

- **Free** for noncommercial users.
- **5 motion sensors**, each carries an accelerometer and gyroscope sampled at 30 Hz.
- **20 test subjects** (13 male & 7 female) ages 19-75.
- **Data processed in Matlab**. Visualization tool is included.

**References:**

Yang, *et al.*, DexterNet: An open platform for heterogeneous body sensor networks and its applications, BSN 2009.

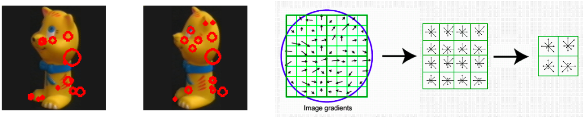
Yang, *et al.*, Distributed Recognition of Human Actions Using Wearable Motion Sensor Networks, JAISE 2009.

# Distributed Object Recognition



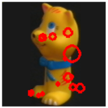
# Motivation: Object Recognition

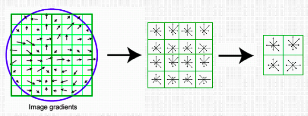
- Affine invariant features, SIFT.



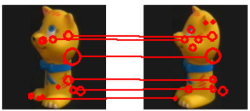
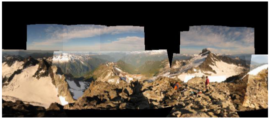
# Motivation: Object Recognition

- Affine invariant features, SIFT.

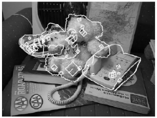





- SIFT Feature Matching [Lowe 1999, van Gool 2004]

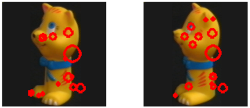
(a) Autostitch

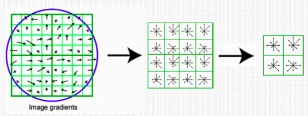


(b) Recognition

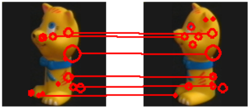
# Motivation: Object Recognition

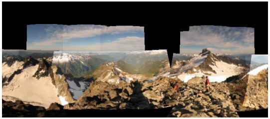
- Affine invariant features, SIFT.



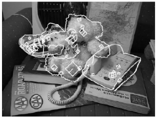



- SIFT Feature Matching [Lowe 1999, van Gool 2004]



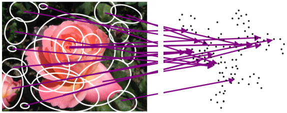


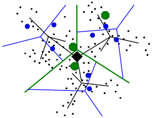
(a) Autostitch



(b) Recognition

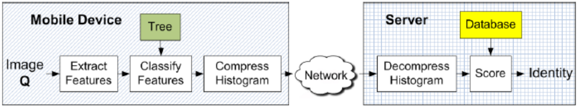
- Bag of Words [Nister 2006]





# Object Recognition in Band-Limited Sensor Networks

1. Compress scalable SIFT tree [Girod et al. 2009]

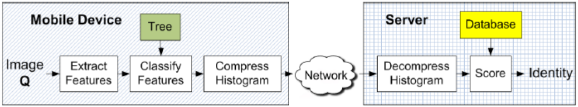



**Observation:** SIFT histogram is largely sparse (up to $10^6$-dim)

$$\begin{array}{lcl} R & : & \text{Sequence of consecutive zero bins.} \\ S & : & \text{Sequence of nonzero bin values.} \end{array}$$

# Object Recognition in Band-Limited Sensor Networks

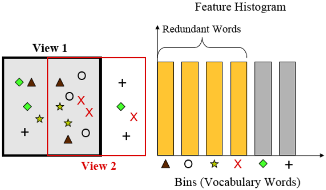
1. Compress scalable SIFT tree [Girod et al. 2009]



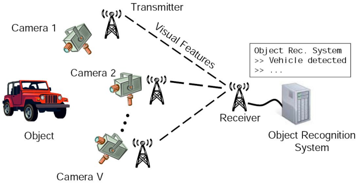

**Observation:** SIFT histogram is largely sparse (up to $10^6$-dim)

$$\begin{aligned} R &: \quad \text{Sequence of consecutive zero bins.} \\ S &: \quad \text{Sequence of nonzero bin values.} \end{aligned}$$

2. Multiple-view SIFT feature selection [Darrell et al. 2008]



# Problem Statement




1. $L$ camera sensors observe a single object in 3-D.
2. The mutual information between cameras are unknown, cross-sensor communication is prohibited.
3. On each camera, seek an encoding function for a **nonnegative, sparse** histogram $\mathbf{x}_i$

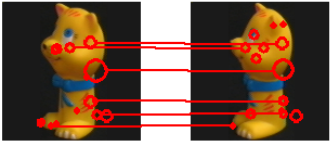
$$f : \mathbf{x}_i \in \mathbb{R}^D \mapsto \mathbf{y}_i \in \mathbb{R}^d$$

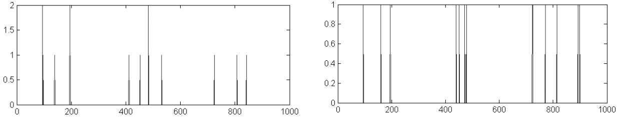
4. On the base station, upon receiving $\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_L$, **simultaneously recover**

$$\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_L,$$

and classify the object class in space.

# Key Observations

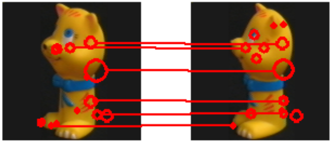


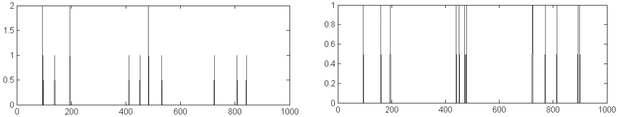



(a) Histogram 1 (b) Histogram 2

- All histograms are **nonnegative** and **sparse**.

# Key Observations

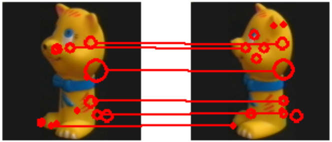


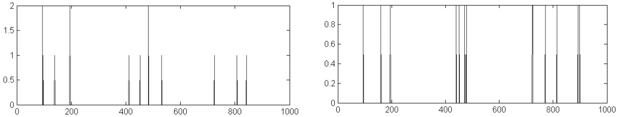



(a) Histogram 1 (b) Histogram 2

- All histograms are **nonnegative** and **sparse**.
- Multiple-view histograms share **joint sparse patterns**.

# Key Observations





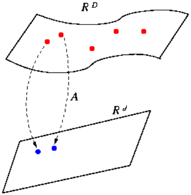

(a) Histogram 1 (b) Histogram 2

- All histograms are **nonnegative** and **sparse**.
- Multiple-view histograms share **joint sparse patterns**.
- Classification is based on the similarity measure in $\ell^2$-norm (linear kernel) or $\ell^1$-norm (intersection kernel).

# Random Projection as Encoding Function

$$\mathbf{y} = A\mathbf{x}$$

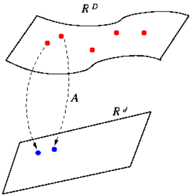
Coefficients of $A \in \mathbb{R}^{d \times D}$ are drawn from zero-mean Gaussian distribution.



# Random Projection as Encoding Function

$$\mathbf{y} = A\mathbf{x}$$

Coefficients of $A \in \mathbb{R}^{d \times D}$ are drawn from zero-mean Gaussian distribution.




## Johnson-Lindenstrauss Lemma [Frankl 1988, Li 2007, Hedge 2007]

For $n$ number of point cloud in $\mathbb{R}^D$, given distortion threshold $\epsilon$, for any

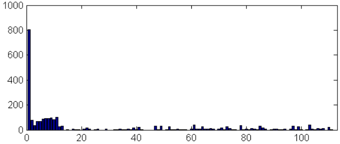
$$d > O(\epsilon^2 \log n),$$

**a Gaussian random projection** $f(\mathbf{x}) = A\mathbf{x} \in \mathbb{R}^d$ **preserves pairwise** $\ell^2$**-distance**

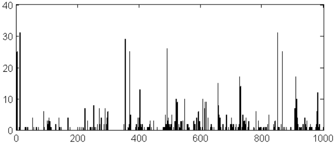
$$(1-\epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|_2^2 \le \|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|_2^2 \le (1+\epsilon)\|\mathbf{x}_i - \mathbf{x}_j\|_2^2.$$

# Classification in Random Projection Space

- Projection only applies to leaf-node histogram $\mathbf{x}_4$




(a) Level 1–3

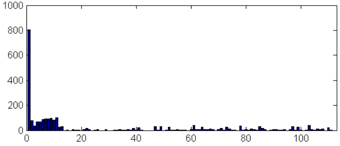



(b) Level 4 (leaf nodes)

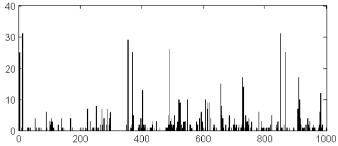
$$\mathbf{x}^T = [\mathbf{x}^{(1)} \in \mathbb{R}, \mathbf{x}^{(2)} \in \mathbb{R}^{10}, \mathbf{x}^{(3)} \in \mathbb{R}^{100}, \mathbf{x}^{(4)} \in \mathbb{R}^{1000}].$$

# Classification in Random Projection Space

- Projection only applies to leaf-node histogram $\mathbf{x}_4$




(a) Level 1–3




(b) Level 4 (leaf nodes)

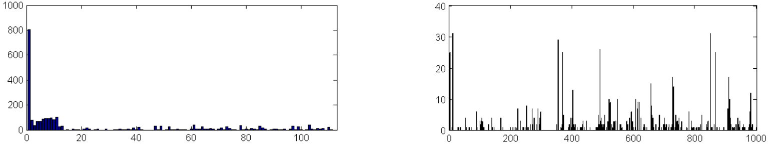
$$\mathbf{x}^T = [\mathbf{x}^{(1)} \in \mathbb{R}, \mathbf{x}^{(2)} \in \mathbb{R}^{10}, \mathbf{x}^{(3)} \in \mathbb{R}^{100}, \mathbf{x}^{(4)} \in \mathbb{R}^{1000}].$$

- Direct classification can be applied using projected leaf histogram (NN or SVM)

$$\mathbf{y} = A\mathbf{x}^{(4)}.$$

# Classification in Random Projection Space

- Projection only applies to leaf-node histogram $\mathbf{x}_4$




(a) Level 1–3 (b) Level 4 (leaf nodes)

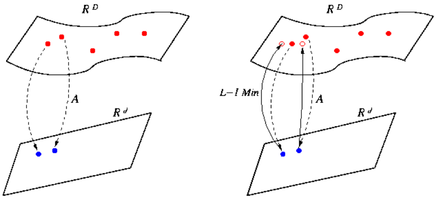
$$\mathbf{x}^T = [\mathbf{x}^{(1)} \in \mathbb{R}, \mathbf{x}^{(2)} \in \mathbb{R}^{10}, \mathbf{x}^{(3)} \in \mathbb{R}^{100}, \mathbf{x}^{(4)} \in \mathbb{R}^{1000}].$$

- Direct classification can be applied using projected leaf histogram (NN or SVM)

$$\mathbf{y} = A\mathbf{x}^{(4)}.$$

- Advantages about Random Projection
  1. Easy to generate and update.
  2. Does not need training prior (universal dimensionality reduction).
  3. faster recognition speed.

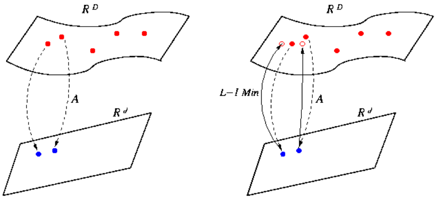
# From J-L Lemma to Compressive Sensing




(a) J-L lemma (b) Compressive sensing

1. **Problem I:** J-L lemma does not provide means to reconstruct **histogram hierarchy**.

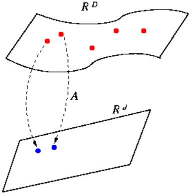
# From J-L Lemma to Compressive Sensing



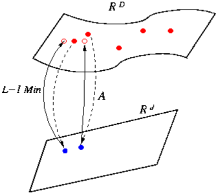

(a) J-L lemma (b) Compressive sensing

1. **Problem I:** J-L lemma does not provide means to reconstruct **histogram hierarchy**.
2. **Problem II:** Gaussian projection **does not preserve $\ell^1$-distance** (for intersection kernels).

# From J-L Lemma to Compressive Sensing



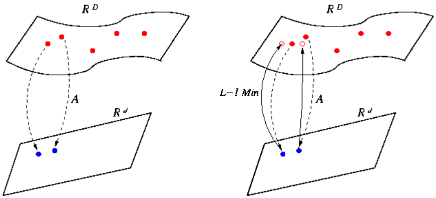

(a) J-L lemma




(b) Compressive sensing

1. **Problem I:** J-L lemma does not provide means to reconstruct **histogram hierarchy**.
2. **Problem II:** Gaussian projection **does not preserve $\ell^1$-distance** (for intersection kernels).
3. **Problem III:** Previous codec's impose explicit mutual information between **fixed camera locations**.

# From J-L Lemma to Compressive Sensing




(a) J-L lemma (b) Compressive sensing

1. **Problem I:** J-L lemma does not provide means to reconstruct **histogram hierarchy**.
2. **Problem II:** Gaussian projection **does not preserve $\ell^1$-distance** (for intersection kernels).
3. **Problem III:** Previous codec's impose explicit mutual information between **fixed camera locations**.

Compressive sensing provides principled solutions to the above problems.

# Compressive Sensing

- **Noise-free case:** Assume $\mathbf{x}_0$ is sufficiently $k$-sparse. Given triplet $(D, d, k)$ and mild condition for $A$,

$$(P_1): \quad \min \|\mathbf{x}\|_1 \text{ subject to } \mathbf{y} = A\mathbf{x}$$

recovers the exact solution.

# Compressive Sensing

- **Noise-free case:** Assume $\mathbf{x}_0$ is sufficiently $k$-sparse. Given triplet $(D, d, k)$ and mild condition for $A$,

$$(P_1): \quad \min \|\mathbf{x}\|_1 \text{ subject to } \mathbf{y} = A\mathbf{x}$$

recovers the exact solution.

- **Noisy case:** Assume $\mathbf{x}_0$ is sufficiently $k$-sparse and bounded noise $\|e\|_2 \leq \epsilon$:

$$\mathbf{y} = A\mathbf{x}_0 + \mathbf{e}.$$

A quadratic program recovers a bounded near solution: $\|\mathbf{x}^* - \mathbf{x}_0\|_2 < C\epsilon$:

$$(P_1'): \quad \min \|\mathbf{x}\|_1 \text{ subject to } \|\mathbf{y} - A\mathbf{x}\|_2 \leq \epsilon$$

# Compressive Sensing

- **Noise-free case:** Assume $\mathbf{x}_0$ is sufficiently $k$-sparse. Given triplet $(D, d, k)$ and mild condition for $A$,

$$(P_1): \quad \min \|\mathbf{x}\|_1 \text{ subject to } \mathbf{y} = A\mathbf{x}$$

recovers the exact solution.

- **Noisy case:** Assume $\mathbf{x}_0$ is sufficiently $k$-sparse and bounded noise $\|e\|_2 \leq \epsilon$:

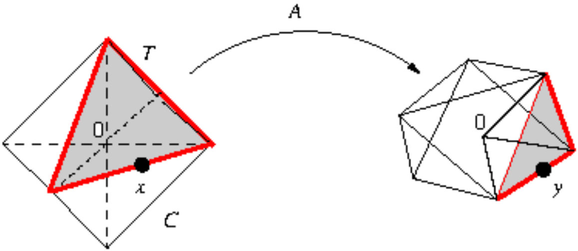
$$\mathbf{y} = A\mathbf{x}_0 + \mathbf{e}.$$

A quadratic program recovers a bounded near solution: $\|\mathbf{x}^* - \mathbf{x}_0\|_2 < C\epsilon$:

$$(P_1'): \quad \min \|\mathbf{x}\|_1 \text{ subject to } \|\mathbf{y} - A\mathbf{x}\|_2 \leq \epsilon$$

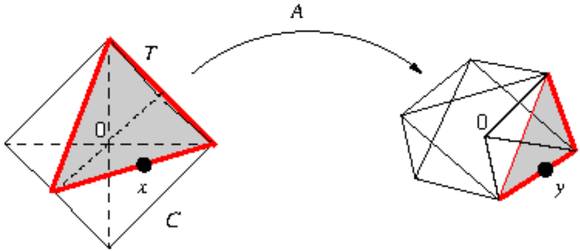
- What are the mild conditions for $A$?
  1. (In)coherence [Gribvonel & Nielsen 2003, Donoho & Elad 2003].
  2. Restricted Isometry Property [Candes & Tao 2005].
  3. $k$-Neighborliness [Donoho 2006].

# $k$-Neighborliness




- Define **cross polytope** $C$ and **quotient polytope** $P$ such that $P = AC$.
- $\mathbf{x}$ is $k$-sparse $\Leftrightarrow$ $\mathbf{x}$ lie in a unique $(k-1)$-face of $C$.

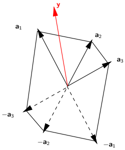
# $k$-Neighborliness




- Define **cross polytope** $C$ and **quotient polytope** $P$ such that $P = AC$.
- $\mathbf{x}$ is $k$-sparse $\Leftrightarrow$ $\mathbf{x}$ lie in a unique $(k-1)$-face of $C$.
- **Necessary and Sufficient:**
  1. If all $(k-1)$-faces of $C$ map to the faces of $P$ on the boundary, $\ell^1/\ell^0$ holds for all $k$-sparse $\mathbf{x}$.
  2. If the $(k-1)$-face where $\mathbf{x}$ lies maps to a face of $P$, then $\ell^1/\ell^0$ holds for this specific $\mathbf{x}$.

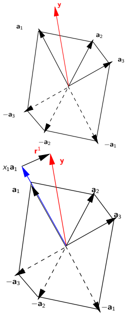
# Matching Pursuit

1. **Initialization:**
   - $\mathbf{y} = [A; -A]\tilde{\mathbf{x}}$, where $\tilde{\mathbf{x}} \geq 0$
   - $k \leftarrow 0$; $\tilde{\mathbf{x}} \leftarrow 0$; $\mathbf{r}^0 \leftarrow \mathbf{y}$; Sparse support $\mathcal{I} = \emptyset$



# Matching Pursuit

1. **Initialization:**
   - $\mathbf{y} = [A; -A]\tilde{\mathbf{x}}$, where $\tilde{\mathbf{x}} \geq 0$
   - $k \leftarrow 0$; $\tilde{\mathbf{x}} \leftarrow 0$; $\mathbf{r}^0 \leftarrow \mathbf{y}$; Sparse support $\mathcal{I} = \emptyset$
2. $k \leftarrow k + 1$:
   - $i = \arg\max_{j \notin \mathcal{I}} \{\mathbf{a}_j^T \mathbf{r}^{k-1}\}$
   - **Update**: $\mathcal{I} = \mathcal{I} \cup \{i\}$; $x_i = \mathbf{a}_i^T \mathbf{r}^{k-1}$; $\mathbf{r}^k = \mathbf{r}^{k-1} - x_i \mathbf{a}_i$



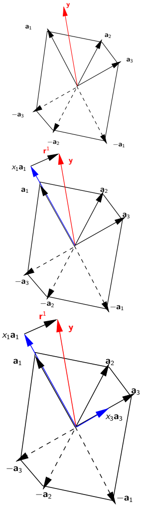
# Matching Pursuit

1. **Initialization:**
   - $\mathbf{y} = [A; -A]\tilde{\mathbf{x}}$, where $\tilde{\mathbf{x}} \geq 0$
   - $k \leftarrow 0$; $\tilde{\mathbf{x}} \leftarrow 0$; $\mathbf{r}^0 \leftarrow \mathbf{y}$; Sparse support $\mathcal{I} = \emptyset$

2. $k \leftarrow k + 1$:
   - $i = \arg\max_{j \notin \mathcal{I}} \{\mathbf{a}_j^T \mathbf{r}^{k-1}\}$
   - **Update**: $\mathcal{I} = \mathcal{I} \cup \{i\}$; $x_i = \mathbf{a}_i^T \mathbf{r}^{k-1}$; $\mathbf{r}^k = \mathbf{r}^{k-1} - x_i \mathbf{a}_i$

3. **If**: $\|\mathbf{r}^k\|_2 > \epsilon$, **go to** STEP 2;
   **Else**: **output** $\tilde{\mathbf{x}}$

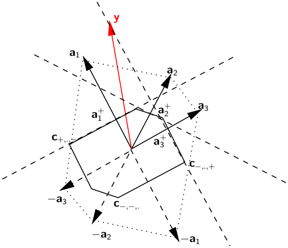
Fail to search sparse solution on the **boundary** of the quotient polytope.



# Polytope Faces Pursuit

- Dual linear program [Chen et al. 1998, Plumbley 2006]

$$\min_{\mathbf{y}=\tilde{A}\tilde{\mathbf{x}}} \mathbf{1}^T\tilde{\mathbf{x}} \Longleftrightarrow \max_{\tilde{A}^T\mathbf{c}\leq\mathbf{1}} \mathbf{y}^T\mathbf{c}$$




- Definition:

$$\mathbf{a}_i^+ \doteq \frac{\mathbf{a}_i}{\|\mathbf{a}_i\|_2^2}$$

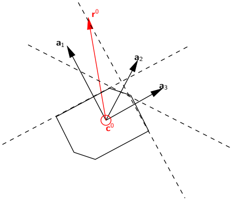
A vertex of the polar polytope at the intersection of hyperplanes $-\mathbf{a}_1^+$ and $-\mathbf{a}_2^+$:

$$\mathbf{c}_{-,-,\cdot} \doteq [-\mathbf{a}_1, -\mathbf{a}_2]^{\dagger\,T}\mathbf{1}$$

# Simulation

$$\max_{\tilde{A}^T \mathbf{c} \leq \mathbf{1}} \mathbf{y}^T \mathbf{c}$$

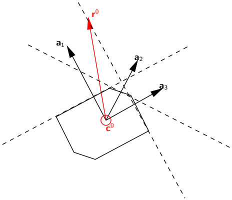
1. Initialization



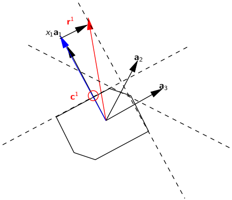
# Simulation

$$\max_{\tilde{A}^T \mathbf{c} \leq \mathbf{1}} \mathbf{y}^T \mathbf{c}$$

1. Initialization



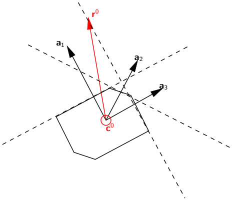

2. Find face: $A^{\mathcal{I}} = \{\mathbf{a}_1\}$.



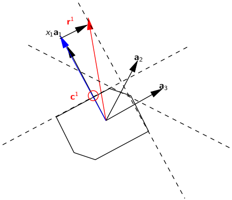
# Simulation

$$\max_{\tilde{A}^T \mathbf{c} \leq \mathbf{1}} \mathbf{y}^T \mathbf{c}$$

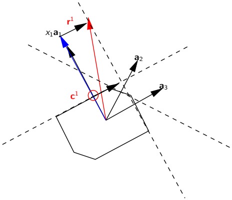
1. Initialization




2. Find face: $A^{\mathcal{I}} = \{\mathbf{a}_1\}$.



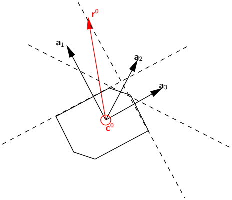

3. Pursuit on the hyperplane



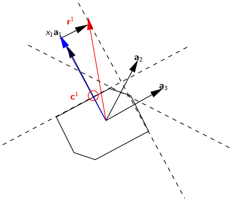
# Simulation

$$\max_{\tilde{A}^T \mathbf{c} \leq \mathbf{1}} \mathbf{y}^T \mathbf{c}$$

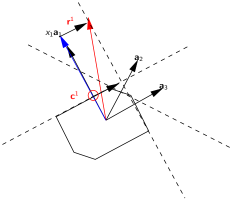
1. Initialization



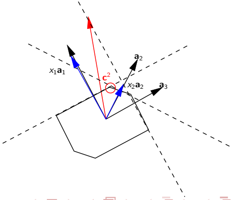

2. Find face: $A^{\mathcal{I}} = \{\mathbf{a}_1\}$.




3. Pursuit on the hyperplane




4. Find face: $A^{\mathcal{I}} = \{\mathbf{a}_1, \mathbf{a}_2\}$.



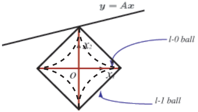
# PFP: Algorithm

1. Convert $\mathbf{y} = A\mathbf{x}$ to
$$\mathbf{y} = [A; -A]\tilde{\mathbf{x}}, \text{ where } \tilde{\mathbf{x}} \geq 0.$$
2. **Initialization:**
   - $k \leftarrow 0$; $\tilde{\mathbf{x}} \leftarrow 0$; $\mathbf{r}^0 \leftarrow \mathbf{y}$; Sparse support $\mathcal{I} = \emptyset$
   - $\mathbf{c}^0 = 0$
3. $k \leftarrow k + 1$:
$$i = \arg\min_{j \notin \mathcal{I}} \{\alpha | \mathbf{a}_j^T (\mathbf{c}^{k-1} + \alpha \mathbf{r}^{k-1}) = 1\}$$
   $\mathcal{I} = \mathcal{I} \cup \{i\}$.
4. **Update**:
   - $\tilde{\mathbf{x}}^{\mathcal{I}} = (\tilde{A}^{\mathcal{I}})^{\dagger}\mathbf{y}$; $\mathbf{r}^k = \mathbf{y} - \tilde{A}\tilde{\mathbf{x}}$
   - $\mathbf{c}^k = ((A^{\mathcal{I}})^{\dagger})^T \mathbf{1}$
5. **If**: $\tilde{\mathbf{x}}^{\mathcal{I}}$ contains negative coefficients, remove indexes from $\mathcal{I}$, **go to** STEP 4.
6. **If**: $\|\mathbf{r}^k\|_2 > \epsilon$, **go to** STEP 3;
   **Else**: **output** $\tilde{\mathbf{x}}$

# Other Nonlinear $\ell^1$ Solvers

- $\ell^p$-min via reweighting [Candes-Wakin-Boyd 2004]

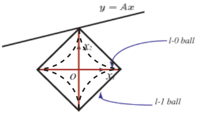
$$\min \sum \log(|x_i| + \epsilon) \text{ subj to } \mathbf{y} = A\mathbf{x}$$



# Other Nonlinear $\ell^1$ Solvers

- $\ell^p$-min via reweighting [Candes-Wakin-Boyd 2004]

$$\min \sum \log(|x_i| + \epsilon) \text{ subj to } \mathbf{y} = A\mathbf{x}$$



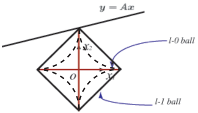

- Linearization!

$$\begin{aligned}\mathbf{x}^{(l+1)} &= \arg\min g(\mathbf{x}^{(l)}) + \nabla g(\mathbf{x}^{(l)}) \cdot (\mathbf{x} - \mathbf{x}^{(l)}) + \text{ h.o.t.} \quad \text{subj to } \mathbf{y} = A\mathbf{x} \\ &\approx \arg\min \nabla g(\mathbf{x}^{(l)}) \cdot \mathbf{x} \text{ subj to } \mathbf{y} = A\mathbf{x}\end{aligned}$$

# Other Nonlinear $\ell^1$ Solvers

- $\ell^p$-min via reweighting [Candes-Wakin-Boyd 2004]

$$\min \sum \log(|x_i| + \epsilon) \text{ subj to } \mathbf{y} = A\mathbf{x}$$




- Linearization!

$$\begin{aligned} \mathbf{x}^{(l+1)} &= \arg\min g(\mathbf{x}^{(l)}) + \nabla g(\mathbf{x}^{(l)}) \cdot (\mathbf{x} - \mathbf{x}^{(l)}) + \text{ h.o.t. } \text{ subj to } \mathbf{y} = A\mathbf{x} \\ &\approx \arg\min \nabla g(\mathbf{x}^{(l)}) \cdot \mathbf{x} \text{ subj to } \mathbf{y} = A\mathbf{x} \end{aligned}$$

- Reweighted $\ell^1$-min

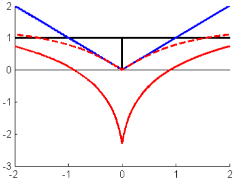
1. Set a reweighting matrix $W^{(0)} = \mathcal{I}$, $l = 0$.
2. Apply **any** $\ell^1$-min solution

$$\mathbf{x}^{(l+1)} = \arg\min \|W^{(l)}\mathbf{x}\|_1 \text{ subj to } \mathbf{y} = A\mathbf{x}.$$

3. Update the weight: $w_{ii}^{(l+1)} = \frac{1}{|x_i^{(l+1)}| + \epsilon}$.
4. Terminate on convergence.

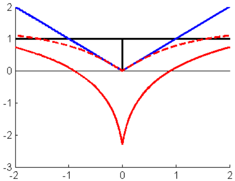
# Another Reweighting Function: atan

- Another example: $g(\mathbf{x}) = \sum_i \text{atan}(|x_i|)$



# Another Reweighting Function: atan

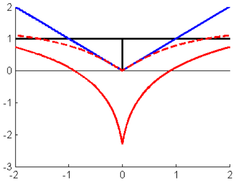
- Another example: $g(\mathbf{x}) = \sum_i \text{atan}(|x_i|)$




$$\boxed{w_{ii} = \frac{1}{1+x_i^2}}$$

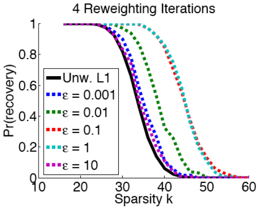
# Another Reweighting Function: atan

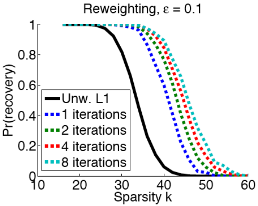
- Another example: $g(\mathbf{x}) = \sum_i \operatorname{atan}(|x_i|)$




$$\boxed{w_{ii} = \frac{1}{1+x_i^2}}$$

- Simulation [Candes 2004]: 256-D signal randomly projected onto 100-D



# Distributed Object Recognition in Smart Camera Networks

Outlines:

1. How to enforce nonnegativity in SIFT histograms?

# Distributed Object Recognition in Smart Camera Networks

Outlines:

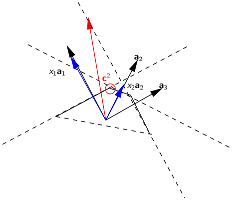
1. How to enforce nonnegativity in SIFT histograms?
2. How to enforce joint sparsity across multiple camera views?

# Enforcing Nonnegativity

- One advantage of PFP is that enforcing nonnegativity is trivial:
  1. **Algebraically:** Do not add antipodal vertexes

$$\mathbf{y} = [A; \boxed{\text{-A}}]\tilde{\mathbf{x}}$$

  2. **Geometrically:** Pursuit on positive faces



# Sparse Innovation Model

- Definition (SIM):

$$\begin{aligned} \mathbf{x}_1 &= \tilde{\mathbf{x}} + \mathbf{z}_1, \\ &\vdots \\ \mathbf{x}_L &= \tilde{\mathbf{x}} + \mathbf{z}_L. \end{aligned}$$

$\tilde{\mathbf{x}}$ is called the **joint sparse** component, and $\mathbf{z}_i$ is called an **innovation**.

# Sparse Innovation Model

- Definition (SIM):

$$\begin{aligned} \mathbf{x}_1 &= \tilde{\mathbf{x}} + \mathbf{z}_1, \\ &\vdots \\ \mathbf{x}_L &= \tilde{\mathbf{x}} + \mathbf{z}_L. \end{aligned}$$

  $\tilde{\mathbf{x}}$ is called the **joint sparse** component, and $\mathbf{z}_i$ is called an **innovation**.

- Joint recovery of SIM

$$\begin{aligned} \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_L \end{bmatrix} &= \begin{bmatrix} A_1 & A_1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \\ A_L & 0 & \cdots & 0 & A_L \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}} \\ \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_L \end{bmatrix} \\ \Leftrightarrow \quad \mathbf{y}' &= A'\mathbf{x}' \in \mathbb{R}^{dL}. \end{aligned}$$

# Sparse Innovation Model

- Definition (SIM):

$$\begin{aligned} \mathbf{x}_1 &= \tilde{\mathbf{x}} + \mathbf{z}_1, \\ &\vdots \\ \mathbf{x}_L &= \tilde{\mathbf{x}} + \mathbf{z}_L. \end{aligned}$$

  $\tilde{\mathbf{x}}$ is called the **joint sparse** component, and $\mathbf{z}_i$ is called an **innovation**.

- Joint recovery of SIM

$$\begin{aligned} \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_L \end{bmatrix} &= \begin{bmatrix} A_1 & A_1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \\ A_L & 0 & \cdots & 0 & A_L \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}} \\ \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_L \end{bmatrix} \\ \Leftrightarrow \quad \mathbf{y}' &= A'\mathbf{x}' \in \mathbb{R}^{dL}. \end{aligned}$$

1. New histogram vector is **nonnegative** and **sparse**.

# Sparse Innovation Model

- Definition (SIM):

$$\begin{aligned} \mathbf{x}_1 &= \tilde{\mathbf{x}} + \mathbf{z}_1, \\ &\vdots \\ \mathbf{x}_L &= \tilde{\mathbf{x}} + \mathbf{z}_L. \end{aligned}$$

  $\tilde{\mathbf{x}}$ is called the **joint sparse** component, and $\mathbf{z}_i$ is called an **innovation**.

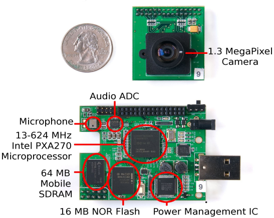
- Joint recovery of SIM

$$\begin{aligned} \begin{bmatrix} \mathbf{y}_1 \\ \vdots \\ \mathbf{y}_L \end{bmatrix} &= \begin{bmatrix} A_1 & A_1 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \\ A_L & 0 & \cdots & 0 & A_L \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}} \\ \mathbf{z}_1 \\ \vdots \\ \mathbf{z}_L \end{bmatrix} \\ \Leftrightarrow \quad \mathbf{y}' &= A'\mathbf{x}' \in \mathbb{R}^{dL}. \end{aligned}$$

1. New histogram vector is **nonnegative** and **sparse**.
2. Joint sparsity $\tilde{\mathbf{x}}$ is automatically determined by $\ell^1$-min: No prior training, no assumption about fixing cameras.

# CITRIC: Wireless Smart Camera Platform

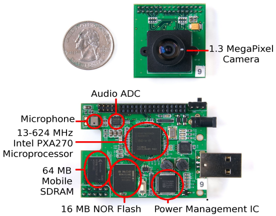
- CITRIC platform




- Available library functions
  1. Full support **Intel IPP Library** and **OpenCV**.
  2. **JPEG compression**: 10 fps.
  3. **Edge detector**: 3 fps.
  4. **Background Subtraction**: 5 fps.
  5. **SIFT detector**: 10 sec per frame.

# CITRIC: Wireless Smart Camera Platform

- CITRIC platform




- Available library functions
  1. Full support **Intel IPP Library** and **OpenCV**.
  2. **JPEG compression**: 10 fps.
  3. **Edge detector**: 3 fps.
  4. **Background Subtraction**: 5 fps.
  5. **SIFT detector**: 10 sec per frame.

- Academic users:

# Demo: Topological Recovery of a Camera Network

# Experiment I: Simulation

- Comparison between **orthogonal matching pursuit**, **polytope faces pursuit**, and **sparse innovation model**:

Table: Simulation of solving 1000-D sparse histograms with $d = 200$, $k = 60$, and $L = 3$.

| Sparsity | (60,0) | (40,20) | (30,30) |
|---|---|---|---|
| $\ell^0_{OMP}$ | 56.14 | 56.14 | 56.14 |
| $\ell^2_{OMP}$ | 1.76 | 1.76 | 1.76 |
| $\ell^0_{PFP}$ | 3.48 | 3.48 | 3.48 |
| $\ell^2_{MMV}$ | 1.84 | 3.10 | 3.67 |
| $\ell^0_{SIM}$ | **1.85** | **1.65** | **1.95** |
| $\ell^2_{SIM}$ | **0.02** | **0.02** | **0.02** |

# Experiment II: COIL-100 object database

- **Database:** 100 objects, each provides 72 images captured with 5 degree difference.

# Experiment II: COIL-100 object database

- **Database:** 100 objects, each provides 72 images captured with 5 degree difference.


- **Setup:**
  - Dense sampling of overlapping $8 \times 8$ grids. Standard SIFT descriptor.
  - 4-level hierarchical $k$-means ($k = 10$): Leaf-node histogram is 1000-D.
  - For each object class, randomly select 10 image for training. Classifier via linear SVM.

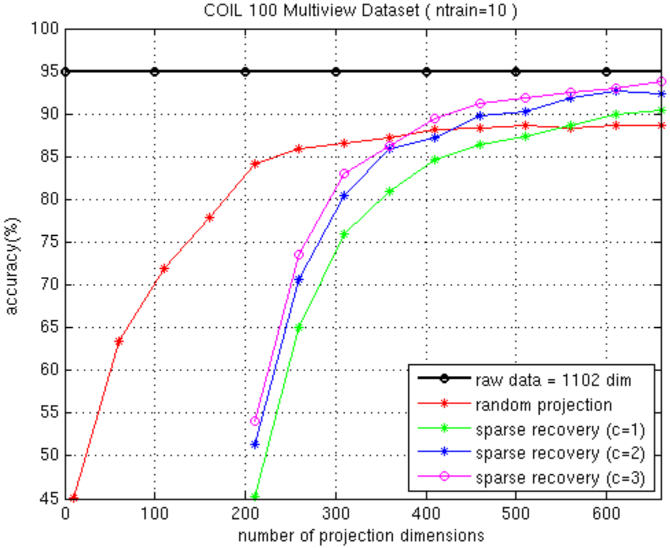



Figure: Per-view recognition accuracy on the COIL-100 database via linear SVMs.

# Distributed Object Recognition in Band-Limited Smart Camera Networks

1. To harness the smart camera capacity, the system is separated in two components: **distributed feature extraction** and **centralized recognition**.

# Distributed Object Recognition in Band-Limited Smart Camera Networks

1. To harness the smart camera capacity, the system is separated in two components: **distributed feature extraction** and **centralized recognition**.
2. **Gaussian random projection** as universal dimensionality reduction function: J-L lemma.

# Distributed Object Recognition in Band-Limited Smart Camera Networks

1. To harness the smart camera capacity, the system is separated in two components: **distributed feature extraction** and **centralized recognition**.
2. **Gaussian random projection** as universal dimensionality reduction function: J-L lemma.
3. **Polytope faces pursuit** exploits two properties of general SIFT histograms:
    - Sparsity.
    - Nonnegativity.

# Distributed Object Recognition in Band-Limited Smart Camera Networks

1. To harness the smart camera capacity, the system is separated in two components: **distributed feature extraction** and **centralized recognition**.
2. **Gaussian random projection** as universal dimensionality reduction function: J-L lemma.
3. **Polytope faces pursuit** exploits two properties of general SIFT histograms:
   - Sparsity.
   - Nonnegativity.
4. **Sparse innovation model** exploits joint sparsity of multiple-view object histograms.

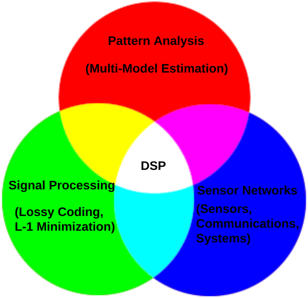
# Distributed Object Recognition in Band-Limited Smart Camera Networks

1. To harness the smart camera capacity, the system is separated in two components: **distributed feature extraction** and **centralized recognition**.
2. **Gaussian random projection** as universal dimensionality reduction function: J-L lemma.
3. **Polytope faces pursuit** exploits two properties of general SIFT histograms:
   - Sparsity.
   - Nonnegativity.
4. **Sparse innovation model** exploits joint sparsity of multiple-view object histograms.
5. Complete system implemented on Berkeley CITRIC sensors.

References


- *Distributed Compression and Fusion of Nonnegative Sparse Signals for Multiple-View Object Recognition*. Information Fusion, 2009.
- *Multiple-View Object Recognition in Band-Limited Distributed Camera Networks*. ICDSC, 2009.

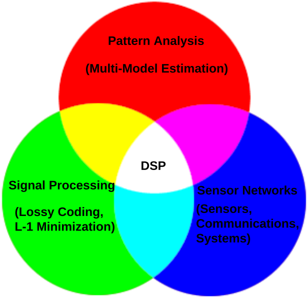
# Conclusion




## Take-Home Message

1. Compressive sensing converts **curse of dimensionality** to **blessing of dimensionality**.

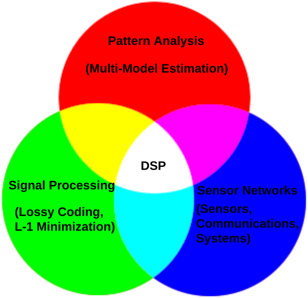
# Conclusion




## Take-Home Message

1. Compressive sensing converts **curse of dimensionality** to **blessing of dimensionality**.
2. Particularly suitable to **distributed sensing and perception** rich in HD data from distributed, resource-constrained sources.

# Conclusion




## Take-Home Message

1. Compressive sensing converts **curse of dimensionality** to **blessing of dimensionality**.
2. Particularly suitable to **distributed sensing and perception** rich in HD data from distributed, resource-constrained sources.
3. Potential impact spans beyond **computer vision** in **health care**, **security**, and **consumer electronics**.