Emerging Applications of Sparse Representations

Pierre Vandergheynst

Signal Processing Lab

Ecole Polytechnique Fédérale de Lausanne (EPFL)

Workshop Sparse Models and Machine Learning, INRIA, 15-16 Oct. 2012

Do we need a new paradigm ?

Do we need a new paradigm ?

2010: 980 exabytes of new digital information

Big Data

Do we need a new paradigm ?

2010: 980 exabytes of new digital information

Big Data

48 hrs video/minute on
Youtube (6 years/day)

Do we need a new paradigm ?

2010: 980 exabytes of new digital information

Big Data

48 hrs video/minute on
Youtube (6 years/day)

Hyperspectral imaging

Do we need a new paradigm ?

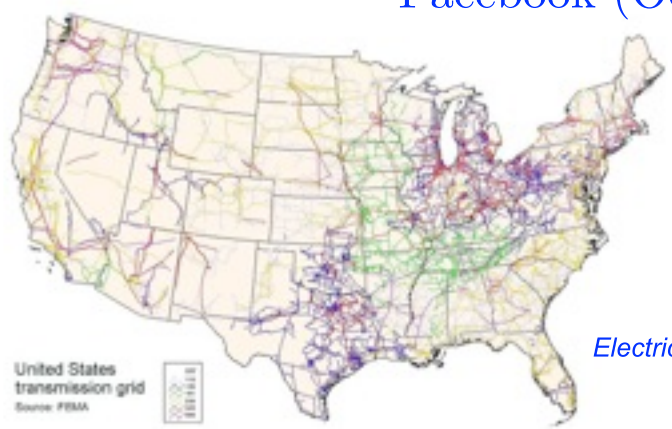
2010: 980 exabytes of new digital information

Big Data

48 hrs video/minute on
Youtube (6 years/day)

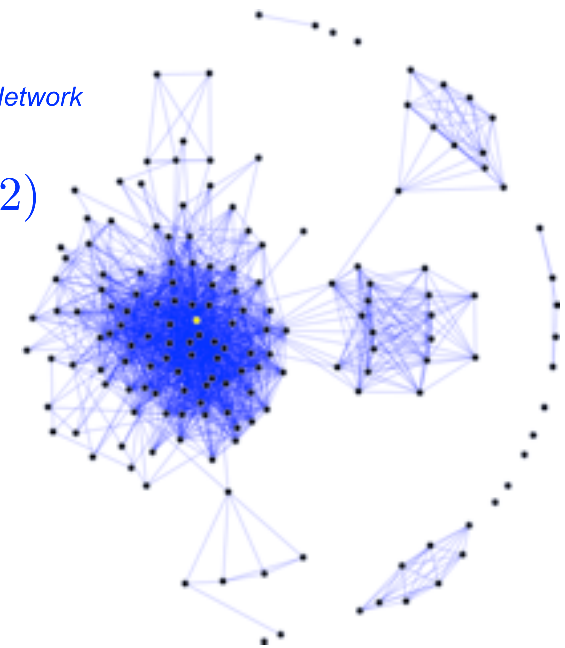
1 billion users on
Facebook (Oct. 2012)

Hyperspectral imaging



Electrical Network

Social Network



Do we need a new paradigm ?

2010: 980 exabytes of new digital information

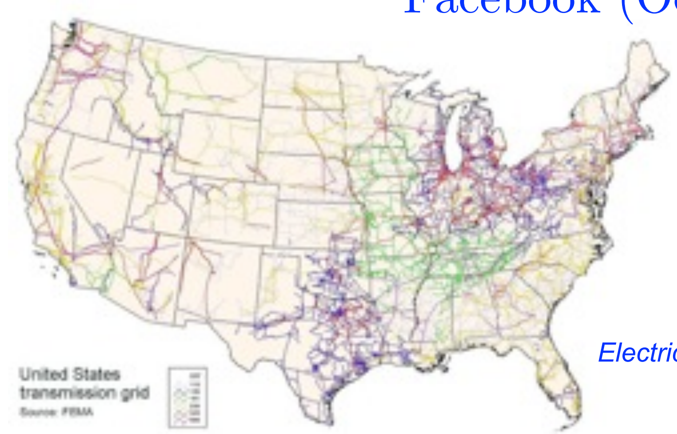
Big Data

Ubiquitous sensing

48 hrs video/minute on
Youtube (6 years/day)

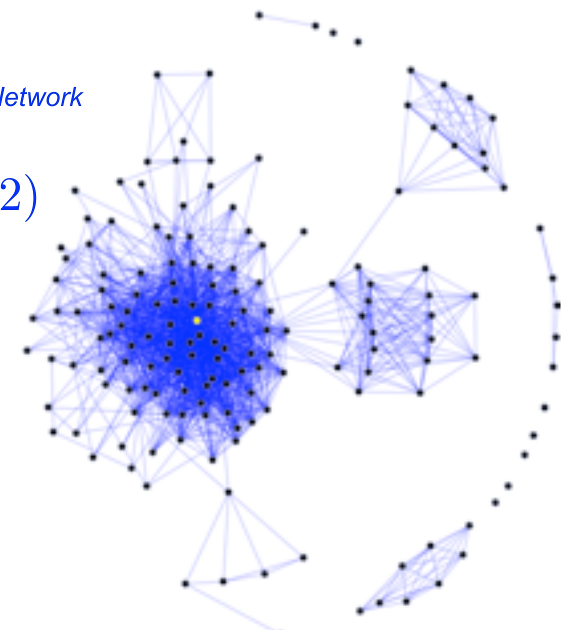
Hyperspectral imaging

1 billion users on
Facebook (Oct. 2012)

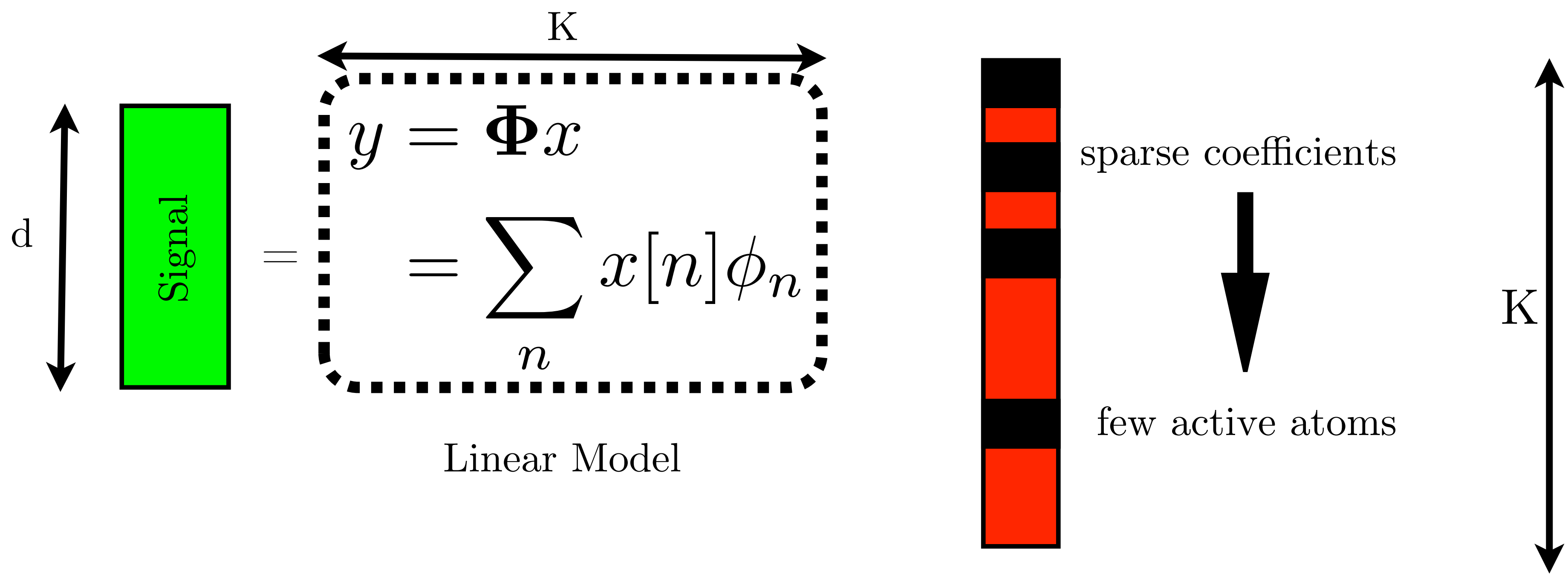


Electrical Network

Social Network

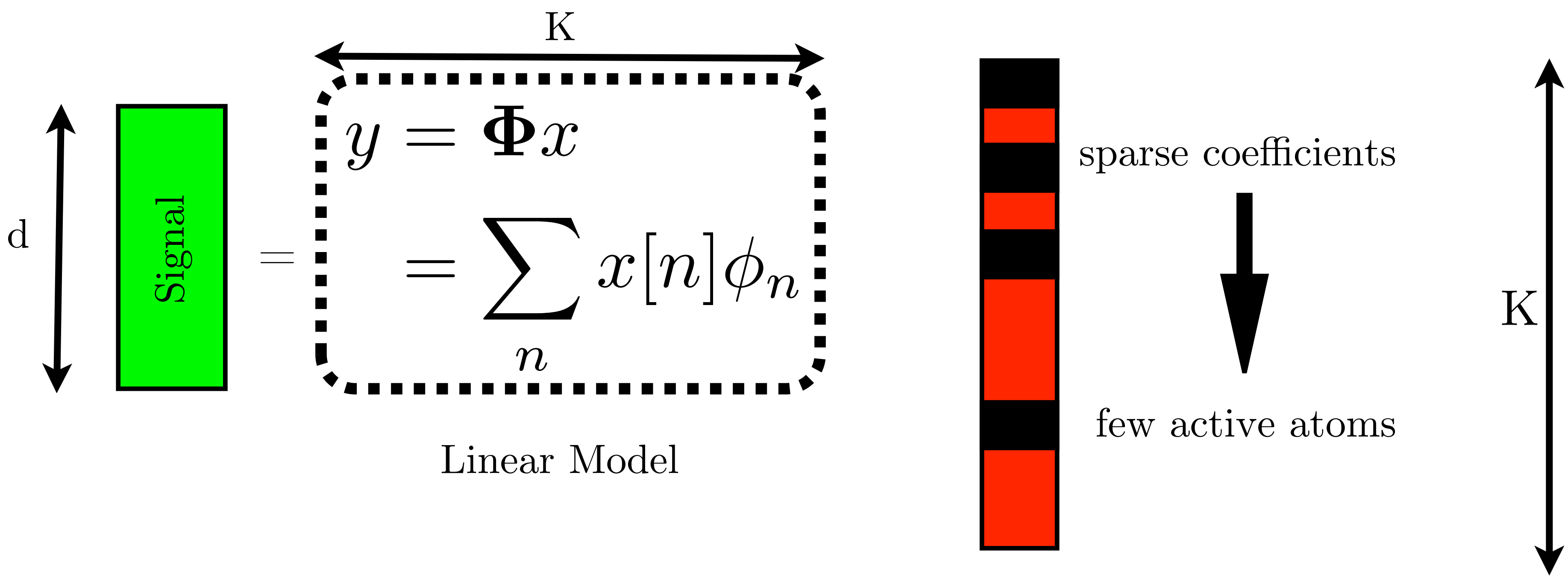


Some notations



$$y = \mathbf{\Phi} x = \sum_n x[n]\phi_n$$

Some notations



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

$$= \sum_n x[n]\phi_n$$

Linear Model

$\mathbf{\Phi} \in \mathbb{R}^{d \times K}$

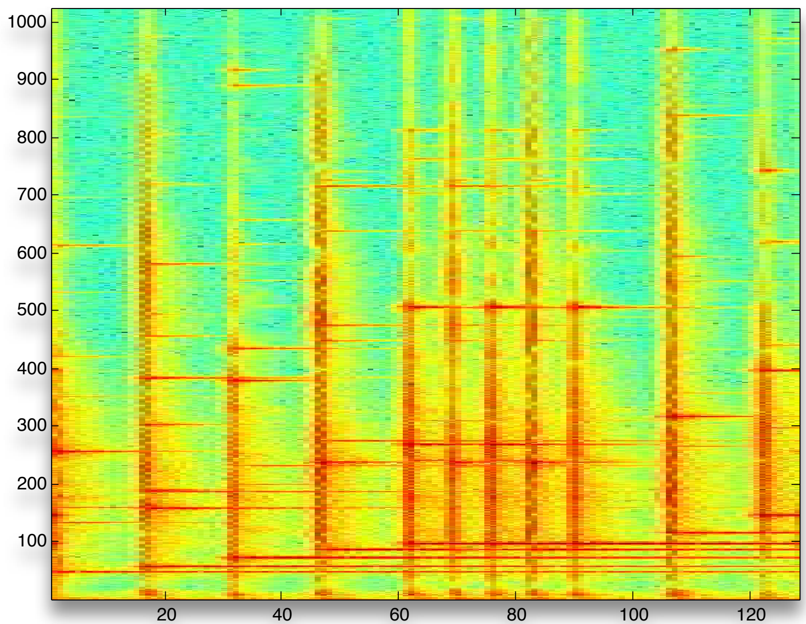
- A convenient ortho basis
- Redundant dictionary, fixed by signal properties (ex: union of wavelets, curvelets, gabor ...) or data driven
- Linear measurement system (compressed sensing)

Sparse Signal Models

$$y = \mathbf{\Phi} x + n$$

Efficient model: coefficient vector is very sparse OR contains few big entries

Example: MDCT/Gabor for audio signals



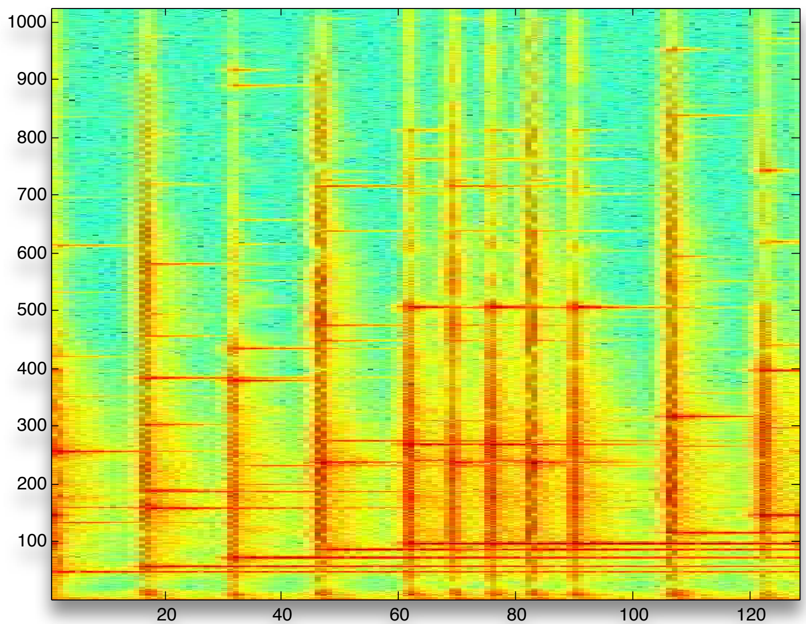
Challenge: Find "good" coefficients for the model

Sparse Signal Models

$$y = \mathbf{\Phi} x + n$$

Efficient model: coefficient vector is very sparse OR contains few big entries

Example: MDCT/Gabor for audio signals



Challenge: Find "good" coefficients for the model

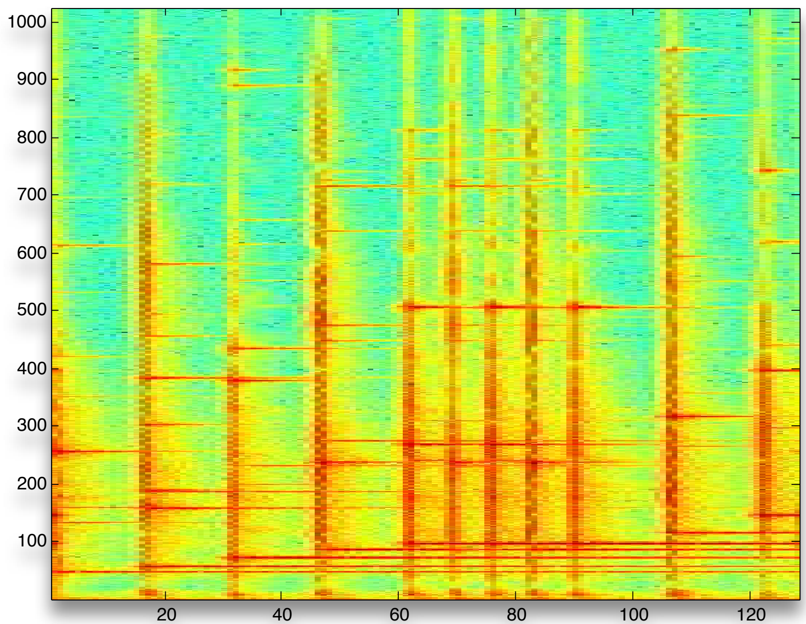
Sparse Signal Models

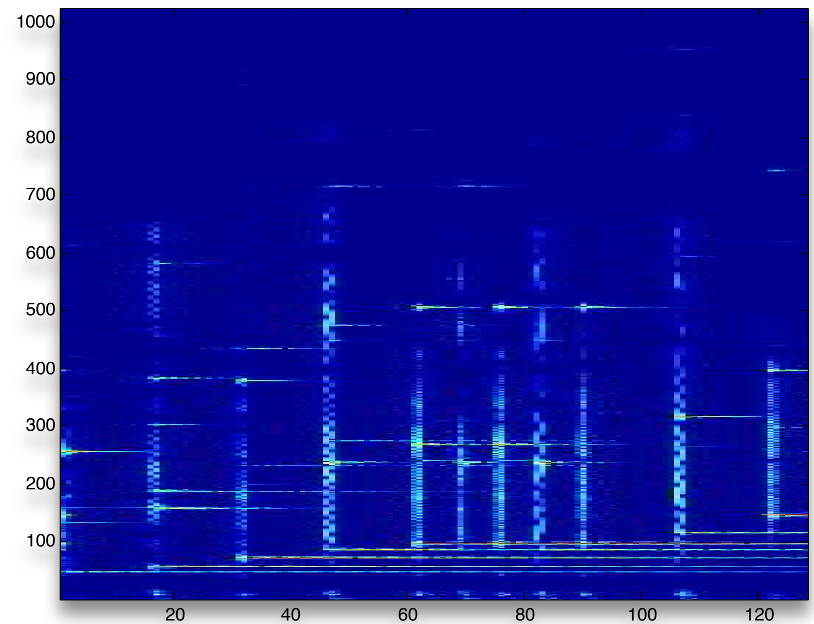
$$y = \mathbf{\Phi} x + n$$

Efficient model: coefficient vector is very sparse
OR contains few big entries

Example: MDCT/Gabor for audio signals

10 % coefficients





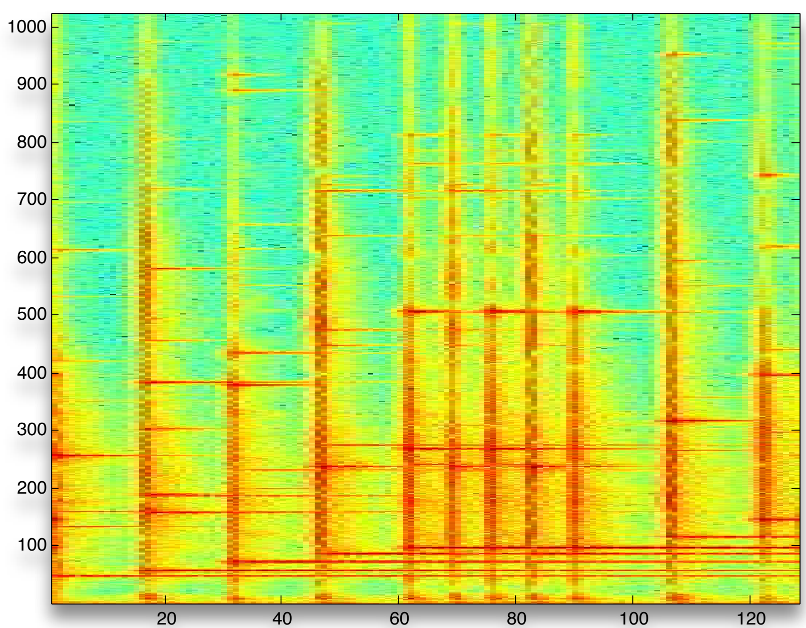
Challenge: Find "good" coefficients for the model

Sparse Signal Models

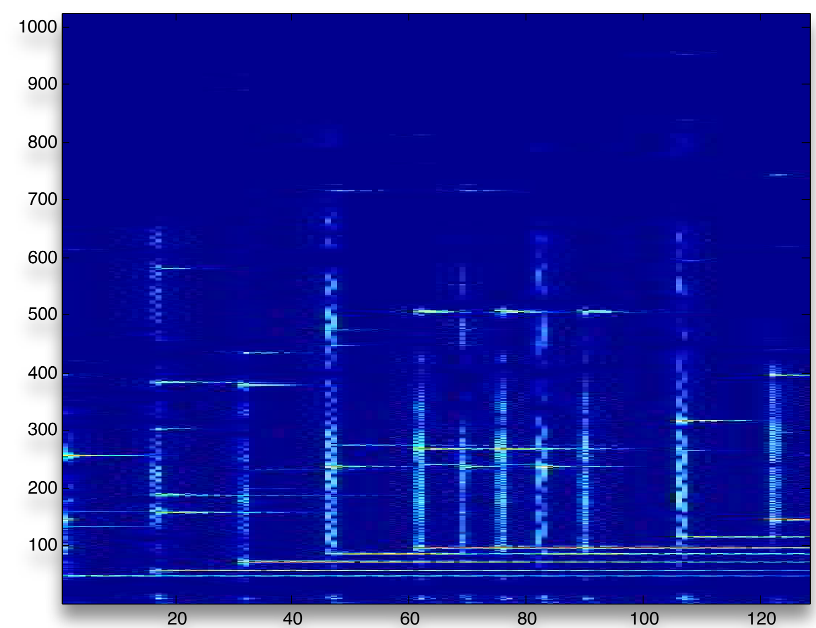
$$y = \mathbf{\Phi} x + n$$

Efficient model: coefficient vector is very sparse
OR contains few big entries

Example: MDCT/Gabor for audio signals



10 % coefficients



Challenge: Find "good" coefficients for the model

Sparse Signal Models - solving

Recover x by promoting sparsity:

Sparse Signal Models - solving

Recover x by promoting sparsity: $\begin{cases} \|x\|_0 \\ \|x\|_1 \end{cases}$

cfr. R. Gribonval

Sparse Signal Models - solving

Recover x by promoting sparsity:

$\|x\|_0$

cfr. R. Gribonval

$\|x\|_1$

$\|x\|_{\mathrm{Struct}}$

cfr. F. Bach

Sparse Signal Models - solving

Recover x by promoting sparsity:

- $||x||_0$
- $||x||_1$ cfr. R. Gribonval
- $||x||_{\text{Struct}}$ cfr. F. Bach

Ex: $$x^* = \arg\min_x \frac{1}{2}||y - \mathbf{\Phi}x||^2 + \mu||x||_?$$

If the model is an ortho basis it is easy to solve

In general models are embodied by dictionaries

Model coefficients can be recovered by families of algorithms, often solving an optimization problem that promotes sparsity:
iterative shrinkage, greedy algorithms etc ...

Sparsity Constrained Inverse Problems

Sparsity constrained recovery and inverse problems:

$$x^* = \arg\min_x \frac{1}{2}\|s - \mathbf{\Phi}x\|^2 + \mu\|x\|_1$$

Sparsity Constrained Inverse Problems

Sparsity constrained recovery and inverse problems:

$$x^* = \arg\min_x \frac{1}{2}\|s - \mathbf{\Phi}x\|^2 + \mu\|x\|_1$$

$$x^* = \arg\min_x \frac{1}{2}\|y - \mathbf{U}\mathbf{\Phi}x\|^2 + \mu\|x\|_1 \Longrightarrow s^* = \mathbf{\Phi}x^*$$

observed signal **degrading operator**

$\tilde{y} = \mathbf{U}s$

Sparsity Constrained Inverse Problems

Sparsity constrained recovery and inverse problems:

$$x^* = \arg\min_x \frac{1}{2}\|s - \mathbf{\Phi}x\|^2 + \mu\|x\|_1$$

$$x^* = \arg\min_x \frac{1}{2}\|y - \mathbf{U\Phi}x\|^2 + \mu\|x\|_1 \Longrightarrow s^* = \mathbf{\Phi}x^*$$

observed signal $\tilde{u} = \mathbf{U}s$

degrading operator

Sparsity Constrained Inverse Problems

Sparsity constrained recovery and inverse problems:

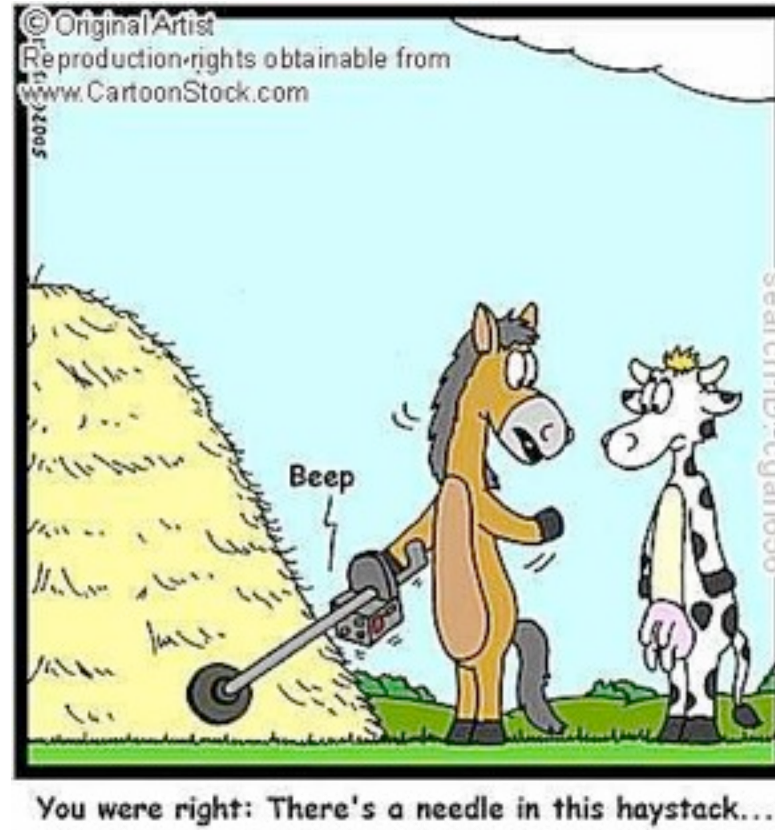
$$x^* = \arg\min_x \frac{1}{2}\|s - \mathbf{\Phi}x\|^2 + \mu\|x\|_1$$

$$x^* = \arg\min_x \frac{1}{2}\|\underbrace{y}_{\text{observed signal}} - \underbrace{\mathbf{U}}_{\text{degrading operator}}\mathbf{\Phi}x\|^2 + \mu\|x\|_1 \Longrightarrow s^* = \mathbf{\Phi}x^*$$

$\tilde{u} = \mathbf{U}s$

New perspective: Sparsity as a regularizer (ex: TV - sparsity of gradients)

How to find a needle in a haystack ?



See tutorials of Rémi Gribonval and Francis Bach on how to find a needle in a haystack

Take Home Messages So Far

- Many signals are sparse on some basis or dictionary
  - *zoology of fixed "optimal" bases*
  - *bases/dictionary learning*
  - *data driven representation (link with Machine Learning)*
- Sparsity offers a lot of flexibility
  - *dimensionality reduction*
  - *compression*
- Algorithms to handle sparsity (provably correct)
  - *greedy, convex relaxation ...*
- Applications !
  - *in particular "compressive sensing"*

Sad Realization and Hopeful Wish

Sparse recovery techniques are great for processing data but ...

... you acquire the whole signal, i.e dimension N and then ...

you trash most of it because you know it is sparse on some good basis !

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

Sad Realization and Hopeful Wish

Sparse recovery techniques are great for processing data but ...

... you acquire the whole signal, i.e dimension N and then ...

you trash most of it because you know it is sparse on some good basis !

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

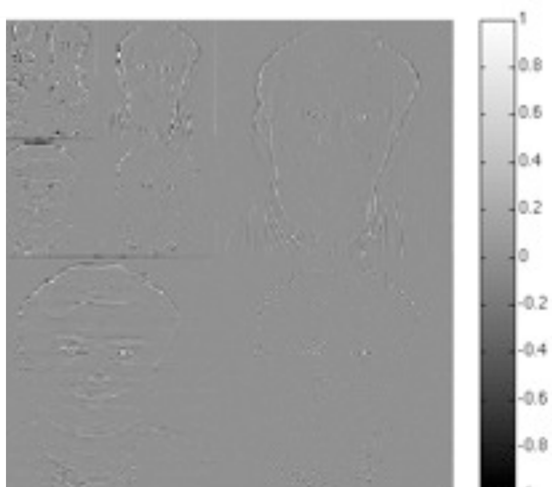
Sad Realization and Hopeful Wish

Sparse recovery techniques are great for processing data but ...

... you acquire the whole signal, i.e dimension N and then ...

you trash most of it because you know it is sparse on some good basis !

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



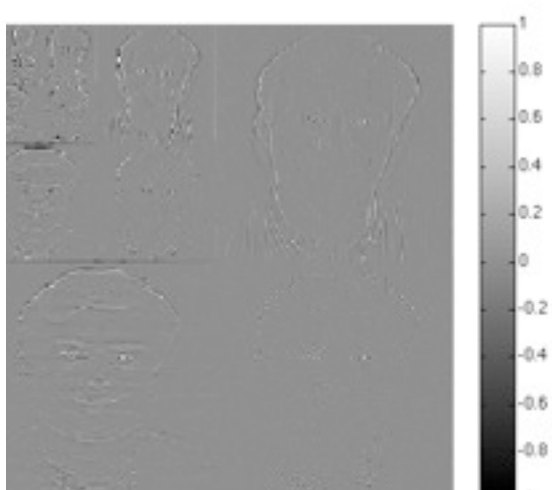
Sad Realization and Hopeful Wish

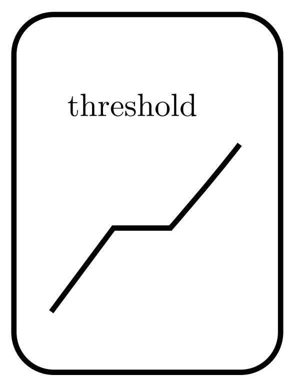
Sparse recovery techniques are great for processing data but ...

... you acquire the whole signal, i.e dimension N and then ...

you trash most of it because you know it is sparse on some good basis !

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





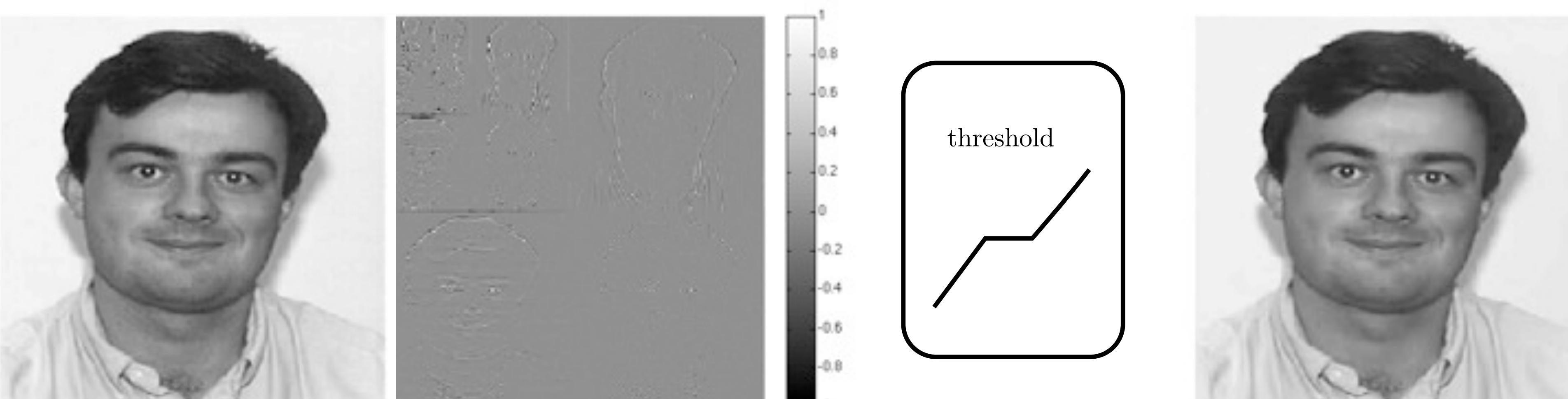
Sad Realization and Hopeful Wish

Sparse recovery techniques are great for processing data but ...

... you acquire the whole signal, i.e dimension N and then ...

you trash most of it because you know it is sparse on some good basis !

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

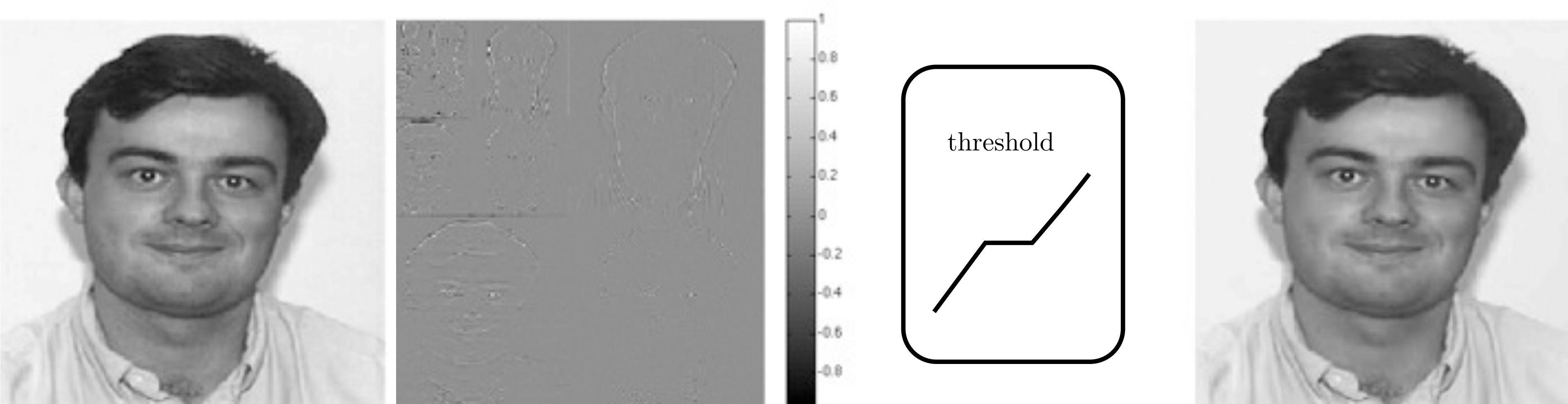


Sad Realization and Hopeful Wish

Sparse recovery techniques are great for processing data but ...

... you acquire the whole signal, i.e dimension N and then ...

Can we estimate the sparse components from few measurements ?



Sparse Recovery and Compressive Sensing

We measure an unknown signal y : $s = \mathbf{A}y$ $\mathbf{A} \in \mathbb{R}^{M \times N}$ $M \ll N$

But we know it comes from a model

$$\arg\min_x \|s - \mathbf{A}\mathbf{\Phi}x\|_2^2 \quad \longrightarrow$$

Try to fit the model to the observations

Sparse Recovery and Compressive Sensing

We measure an unknown signal y : $s = \mathbf{A}y$ $\mathbf{A} \in \mathbb{R}^{M \times N}$ $M \ll N$

But we know it comes from a model

$$\arg\min_x \|s - \mathbf{A\Phi}x\|_2^2 \rightarrow$$

Try to fit the model to the observations

The fit maybe hard to find BUT we know the model is sparse

$$\arg\min_x \|s - \mathbf{A\Phi}x\|_2^2 + \lambda\|x\|_1$$

$$\arg\min_{x \in \mathbb{R}^N} \|x\|_1 \text{ subject to } \|y - \mathbf{A\Phi}x\|_2^2 \leq \epsilon$$

Sparsity constrained inverse problem

Bring Home Key Concepts

- Sparsity / Compressibility
  - large dimension but few degrees of freedom
- Linear (non adaptive !) measurements
  - $M = \mathcal{O}(K \log N/K)$
- Incoherence / Randomness
  - each measurement counts !
  - universality, robustness, scalability
- Recovery
  - provably correct algoS to solve inverse problem

See 2nd talk of R. Gribonval

[Candès, Romberg, Tao, Donoho]

Applications

well, applications @EPFL really...

Spread-Spectrum Compressive Sensing

A common sensing model: projections onto an ONB

$y = \mathbf{\Psi} x$ Generative sparse model

$s = \mathbf{\Phi}_{\Omega}^{*} x$ Sensing model: randomly select few projections onto an ONB

Spread-Spectrum Compressive Sensing

A common sensing model: projections onto an ONB

$y = \mathbf{\Psi} x$ Generative sparse model

$s = \mathbf{\Phi}_{\Omega}^{*} x$ Sensing model: randomly select few projections onto an ONB

Recovery performance driven by the incoherence between the two bases:

$$\mu = \max_{i,j} \left| \langle \phi_i, \psi_j \rangle \right|$$

Spread-Spectrum Compressive Sensing

Spread-Spectrum Compressive Sensing

Sparsity/Sensing in ONBs: CS results driven by mutual coherence

Introduce modulation aimed at decoherence

Spread-Spectrum Compressive Sensing

Sparsity/Sensing in ONBs: CS results driven by mutual coherence

Introduce modulation aimed at decoherence

Theoretical Analysis of Recovery (joint with R. Gribonval):

$$\mathsf{A}_\Omega = \Phi_\Omega^* \mathsf{C} \Psi \in \mathbb{C}^{m \times N} \qquad \beta(\Phi, \Psi) = \max_{1 \leqslant i,j \leqslant N} \sqrt{\sum_{k=1}^{N} \left| \phi_{ki}^* \psi_{kj} \right|^2}$$

modulation

$$\mu \leqslant \beta(\Phi, \Psi) \sqrt{2 \log (2N^2/\epsilon)} \quad \text{with probability at least } 1 - \epsilon$$

$$m \geqslant C' N \beta^2 (\Phi, \Psi)\, s \log^8 (N)$$

Spread-Spectrum Compressive Sensing

Sparsity/Sensing in ONBs: CS results driven by mutual coherence

Introduce modulation aimed at decoherence

Theoretical Analysis of Recovery (joint with R. Gribonval):

$$\mathsf{A}_\Omega = \Phi_\Omega^* \mathsf{C} \Psi \in \mathbb{C}^{m \times N} \qquad \beta(\Phi, \Psi) = \max_{1 \leqslant i,j \leqslant N} \sqrt{\sum_{k=1}^{N} \left|\phi_{ki}^* \psi_{kj}\right|^2}$$

modulation

$$\mu \leqslant \beta(\Phi, \Psi) \sqrt{2 \log(2N^2/\epsilon)} \quad \text{with probability at least } 1 - \epsilon$$

$$m \geqslant C' N \beta^2(\Phi, \Psi)\, s \log^8(N)$$

Universal Sensing Basis (Fourier, Hadamard, Noiselet ...)

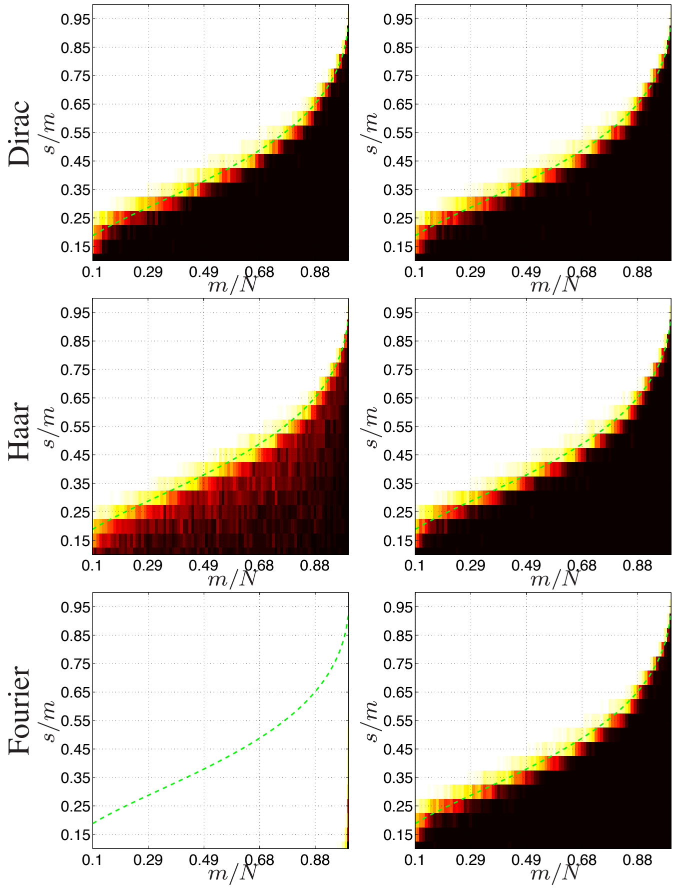
$$|\phi_{ki}| = N^{-1/2} \Longrightarrow \beta(\Phi, \Psi) = N^{-1/2} \text{ and } \mu \simeq N^{-1/2}$$

Spread-Spectrum Compressive Sensing

Universal sensing bases give optimal sampling, independently of sparsity basis !

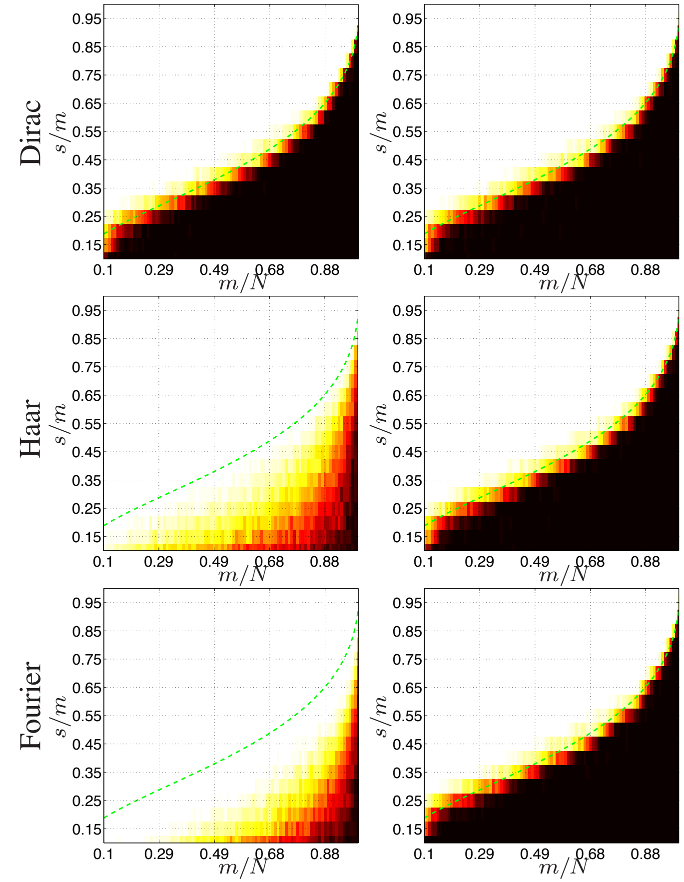
Sensing = Fourier

Without pre-modulation With pre-modulation



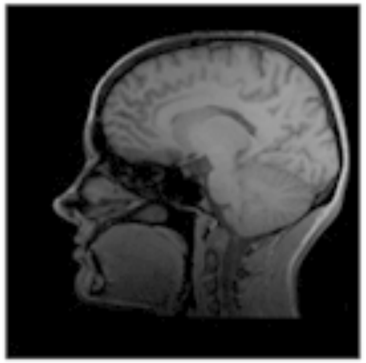
Sensing = Hadamard

Without pre-modulation With pre-modulation

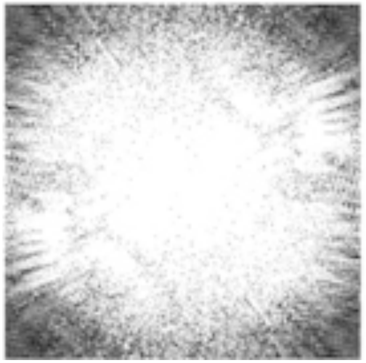


Compressed Sensing in MRI

Problem: accelerate MRI acquisition



$\rho(\mathbf{x})$



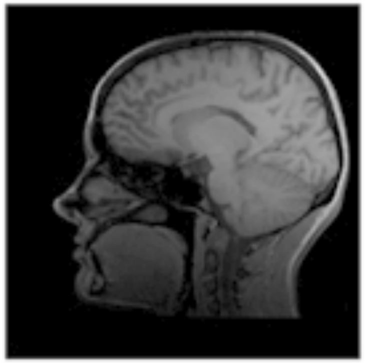
$$\nu\left(\boldsymbol{k}\right) = \int_{\mathbb{R}^2} \rho\left(\boldsymbol{x}\right) \mathrm{e}^{-2\mathrm{i}\pi \boldsymbol{k}\cdot\boldsymbol{x}} \, \mathrm{d}^2\boldsymbol{x}.$$

Sensing Model: wide-band modulation

$$y(t) = p(t) \cdot s(t) \Leftrightarrow \hat{y}(\omega) = \hat{p}(\omega) \star \hat{s}(\omega)$$

Compressed Sensing in MRI

Problem: accelerate MRI acquisition



$\rho(\mathbf{x})$

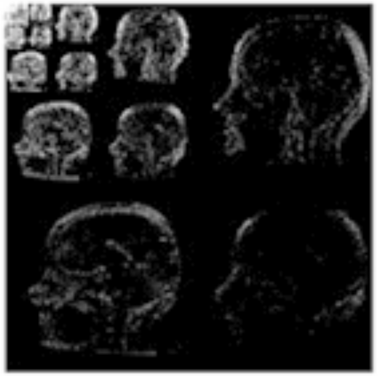
$$\nu(\boldsymbol{k}) = \int_{\mathbb{R}^2} \rho(\boldsymbol{x})\, \mathrm{e}^{-2\mathrm{i}\pi \boldsymbol{k}\cdot\boldsymbol{x}}\, \mathrm{d}^2\boldsymbol{x}.$$

Sensing Model: wide-band modulation

$$y(t) = p(t) \cdot s(t) \Leftrightarrow \hat{y}(\omega) = \hat{p}(\omega) \star \hat{s}(\omega)$$

Compressed Sensing in MRI

Problem: accelerate MRI acquisition



$\rho(\mathbf{x})$

$$\nu(\boldsymbol{k}) = \int_{\mathbb{R}^2} \rho(\boldsymbol{x})\, \mathrm{e}^{-2\mathrm{i}\pi \boldsymbol{k}\cdot\boldsymbol{x}}\, \mathrm{d}^2\boldsymbol{x}.$$

Sensing Model: wide-band modulation

$$y(t) = p(t)\cdot s(t) \Leftrightarrow \hat{y}(\omega) = \hat{p}(\omega) \star \hat{s}(\omega)$$

Spread Spectrum in MRI

CS has already been applied to MR [Lustig]

Here: Explore potential of spread-spectrum "conditioning"

$$\nu(\mathbf{k}) = \int_{\mathbb{R}^2} \rho(\mathbf{x}) \boxed{e^{i\pi w|\mathbf{x}|^2}} e^{-2i\pi \mathbf{k}\cdot\mathbf{x}} d^2\mathbf{x}$$

Phase Scrambling

- well-known in MRI (high Dynamic, reduce aliasing)
- obtained through dedicated coils or RF pulses

Measurement model:

$$\nu \equiv \phi^{(w)} \rho + \mathbf{n} \text{ with } \phi^{(w)} \equiv \mathbf{MFC}^{(w)}$$

M/2 complex

N-d real signal

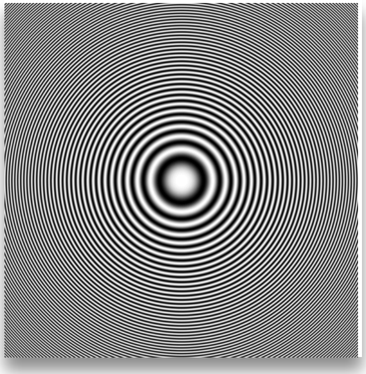
Sub-sample

Fourier

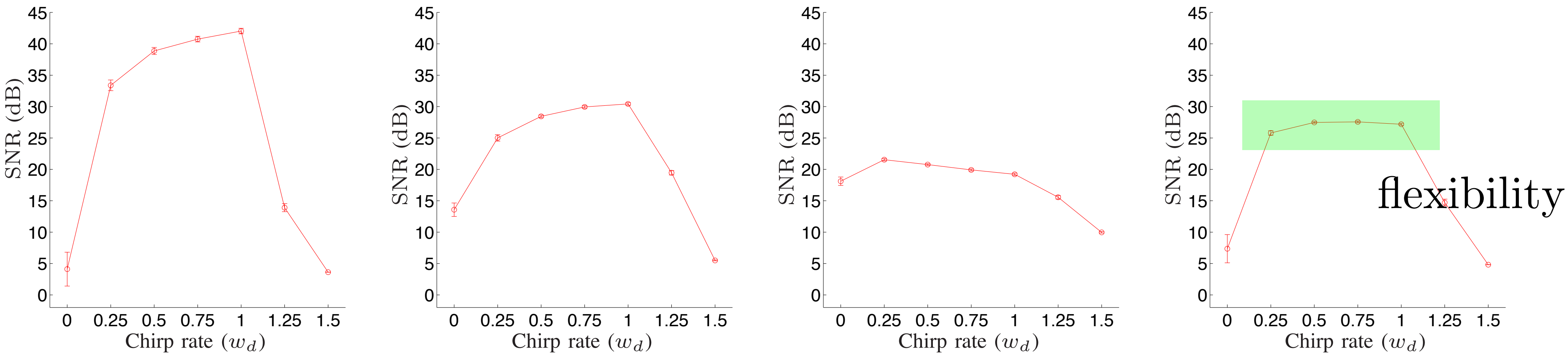
Diagonal chirp matrix

Simulations - leakage

Shepp-Logan phantom and chirp



Input SNR = 30dB, coverage 20%, sparsity basis = wavelets, 30 simulations



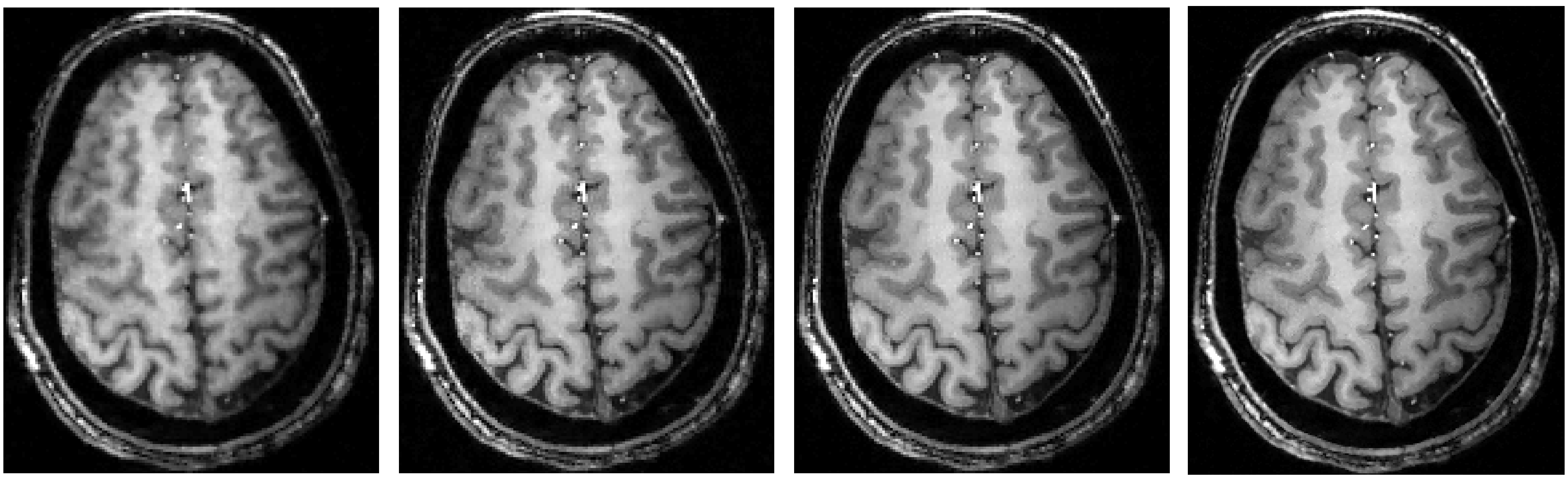
s=5 s=3 s=1 overall

Application: Fast MR Imaging

Real data acquisition, 7T MRI@EPFL

chirp pre-modulation implemented with a dedicated shim coil

Acceleration x 4	Acceleration x 2	Acceleration x 1	Enhanced resolution



Application: ULP bio-sensing

Application: ULP bio-sensing

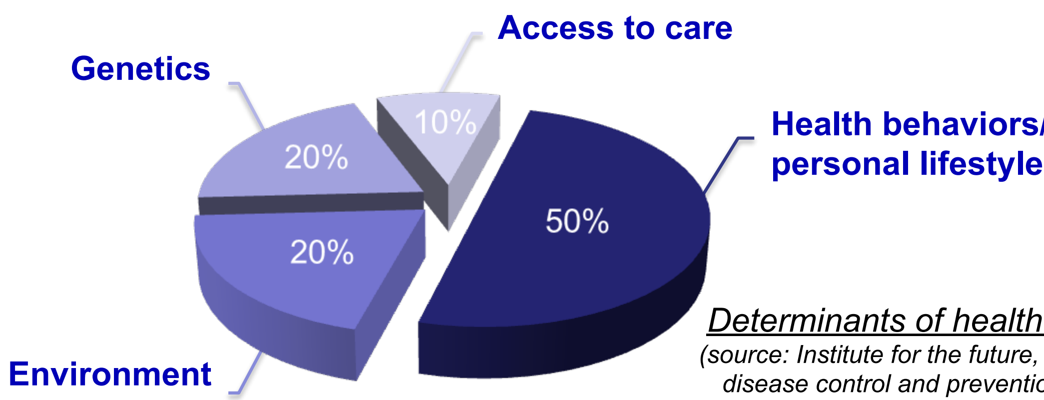
Societal motivation:

demography and life style conspire toward a global health care crisis (lifestyle-induced diseases, NCDs)

Application: ULP bio-sensing

Societal motivation:

demography and life style conspire toward a global health care crisis (lifestyle-induced diseases, NCDs)

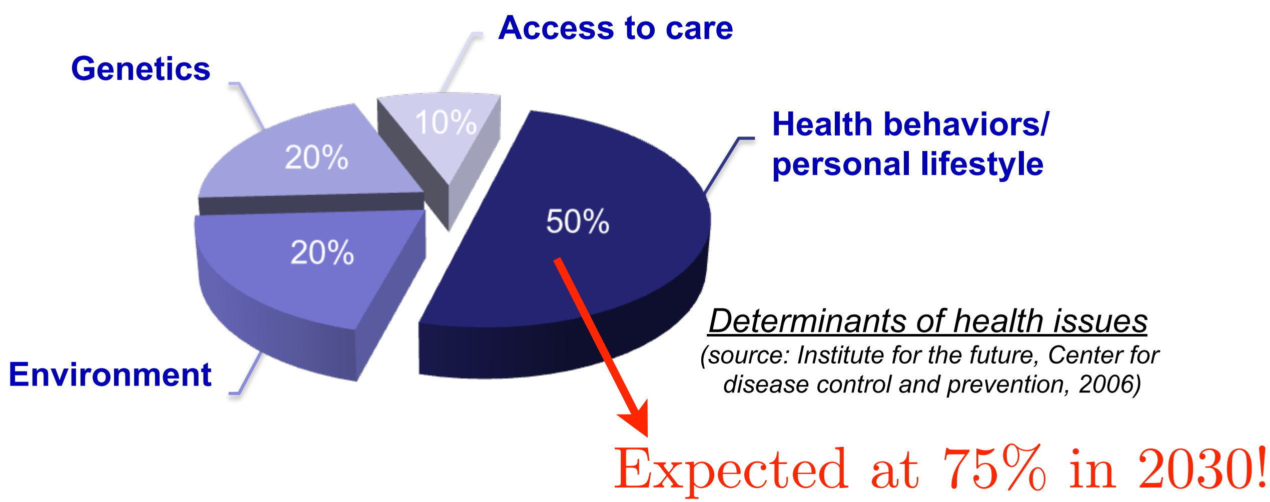


Determinants of health issues
(source: Institute for the future, Center for disease control and prevention, 2006)

Application: ULP bio-sensing

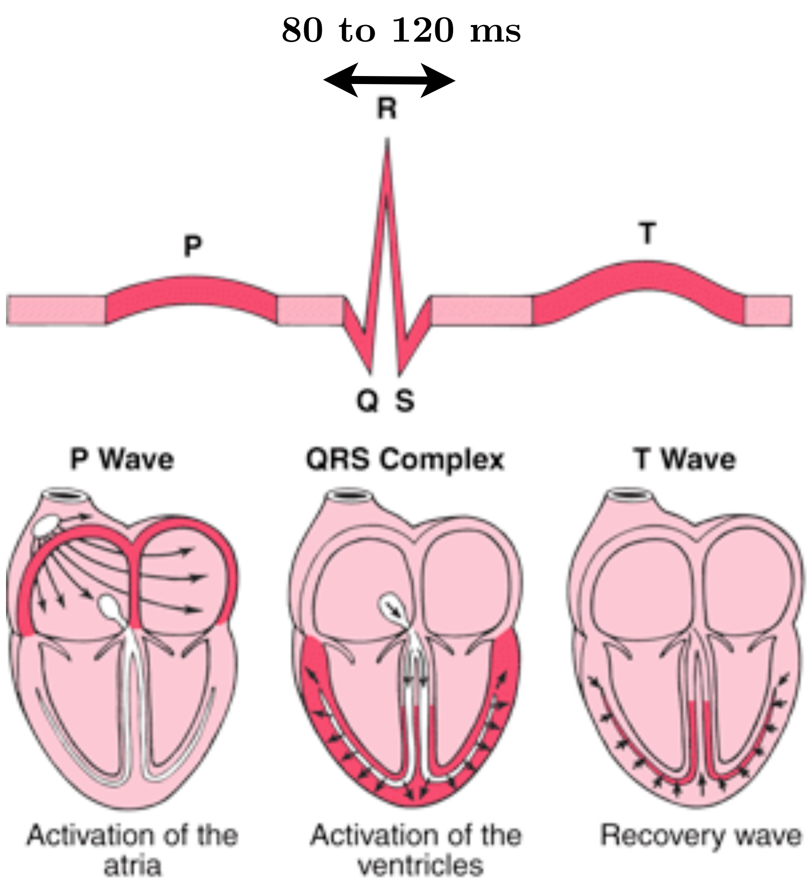
Societal motivation:

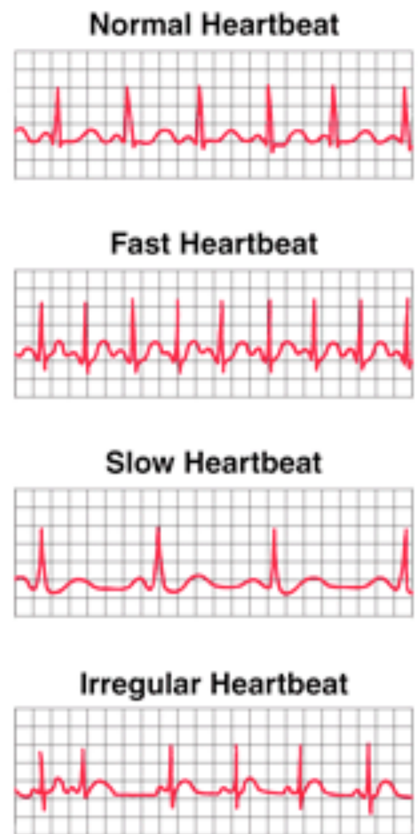
demography and life style conspire toward a global health care crisis (lifestyle-induced diseases, NCDs)



Determinants of health issues
(source: Institute for the future, Center for disease control and prevention, 2006)

Typical modality: ECG





Typical components, could be learned

Often, wavelets are used

Sampling rate few hundred Hz

Low-power ECG ambulatory system

Problem: sense and transmit ECG (possibly multi-lead) from a low power body-area network

Compression ? Surely if we transmit less, we will waste less power in communication.

Sure, but if we compress more we will waste energy using a complex encoder !

Can CS offer an interesting trade-off ?

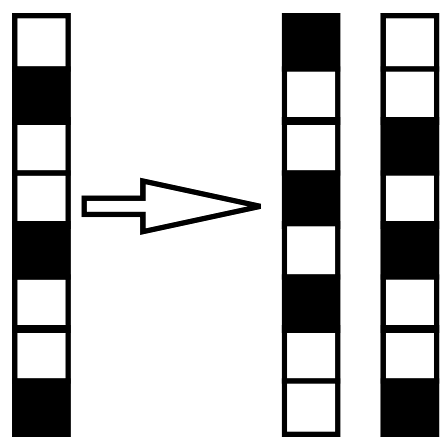
Can everything be real-time ?

- State-of-the-art
  - Wavelet transform, followed by thresholding, quantization and entropy coding
  - Pros: excellent compression results, signals nicely sparse (at least ventricular part)
  - Cons: Full wavelet transform must be implemented on the sensing node
- Can a light compressive sensing encoder achieve a good trade-off compression/power consumption ?

What is a good sensing matrix for low-power sensing ?

Surely not gaussian ! (dense, complex to apply to signal and even complex to generate ...)

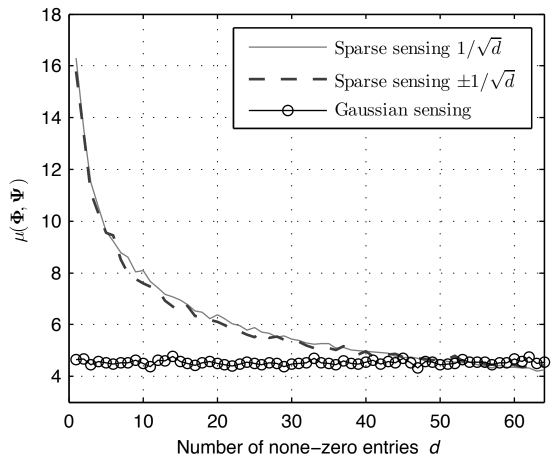
Sparse matrices, binary entries (ex: expander graphs)



generate binary vector with d non-zero elements

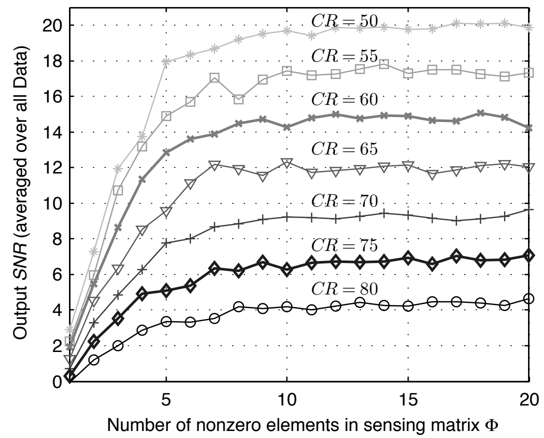
generate full matrix by random permutations

Performance indexes for sensing mechanism

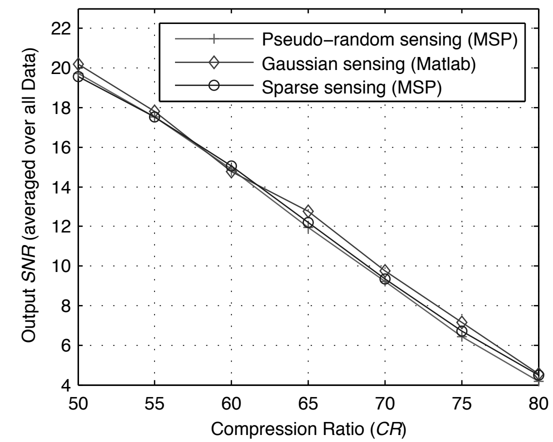


Mutual coherence $\mu(\mathbf{\Phi}, \mathbf{\Psi})$ *vs.* d

coherence quickly approaches "optimal case"



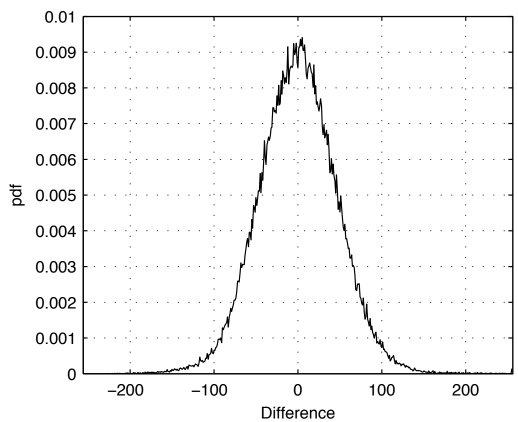
SNR saturates after d=12 non-zero elements



Hardly any difference between proposed sensing and gaussian sensing

2 s of signal are sensed in 82 ms

Coding: simple predictive scheme



difference between successive sensing vectors
Compression Ration: 20%

Gaussian RD theory

9 bits quantizer

Huffman coding

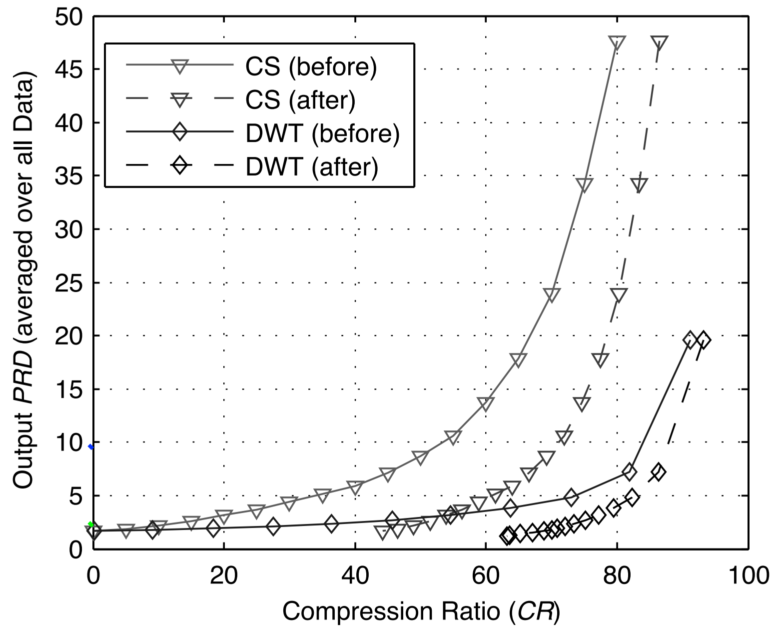
1.5 kB codebook stored on platform

Total memory footprint of CS implementation:

6.5 kB of RAM for computations

7.5 kB of Flash

Comparisons - Quality vs Compression



good

very good

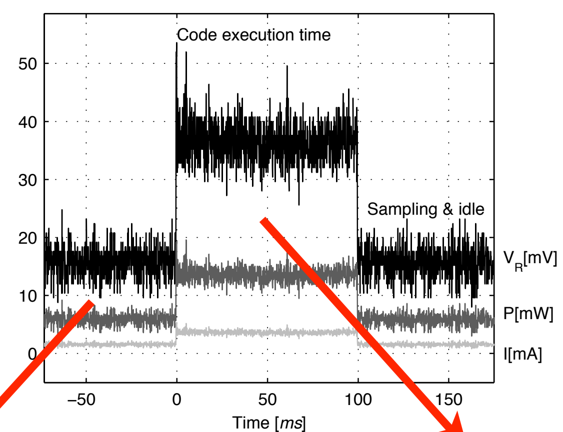
In terms of pure compression performance, an optimized DWT encoder is clearly (and obviously) better than the non-adaptive CS scheme

Comparisons - Power consumption

Consumption *measured* on the platform in real-life use

Code execution time: 95ms for CS, 580ms for DWT

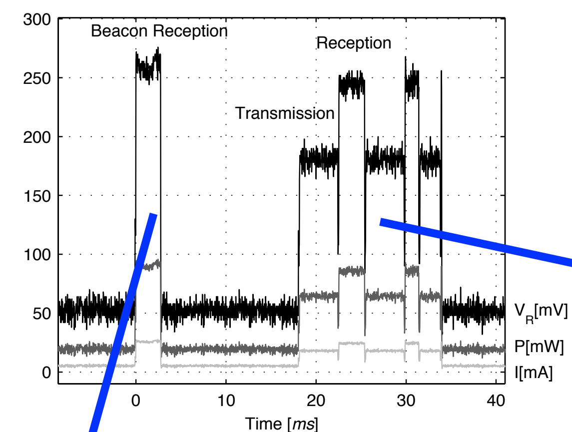
Node consumption



MSP430 idle and sampling till buffer full

CS code running

Radio consumption



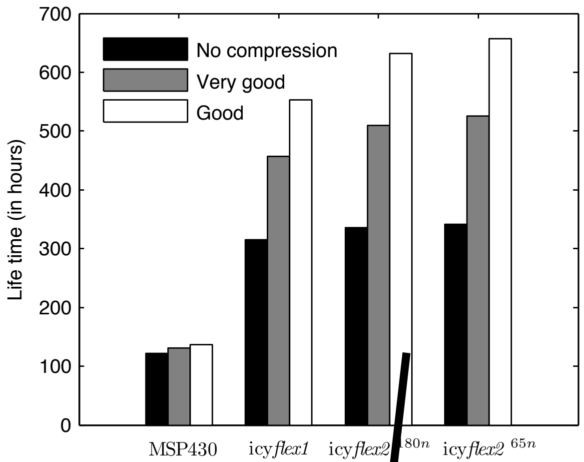
periodic “ping”

transmission & reception

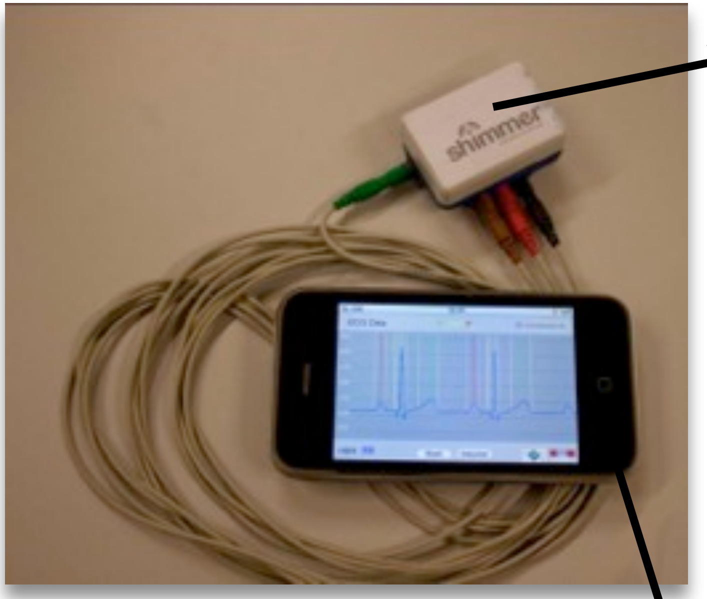
Note: contrary to what is usually assumed in literature, optimized antennas are really low power !

Comparisons - Power consumption

Final results: it is important to know your architecture VERY well



State-of-the-art low-power micro-controllers

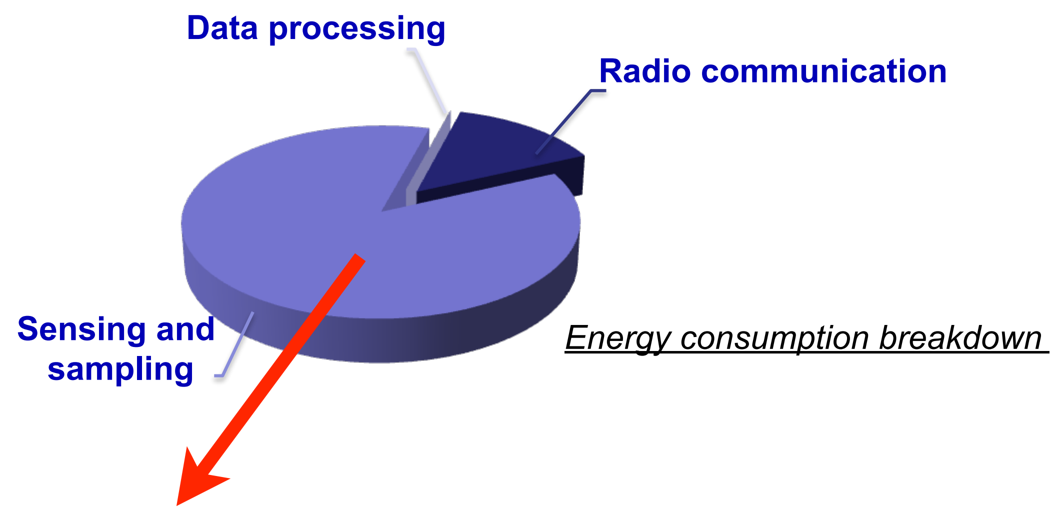


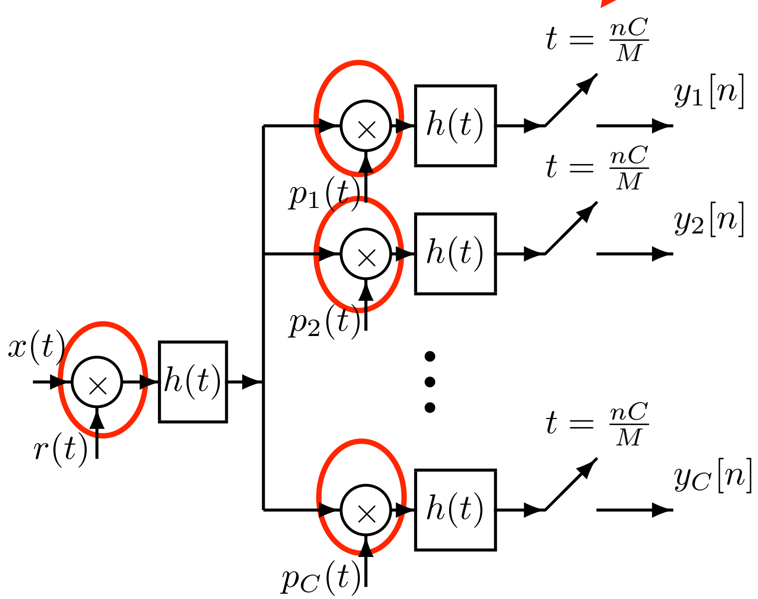
Light Compressive sensing encoder on embedded system

CS decoder (iterative soft thresholding) running real-time on an iphone

92% lifetime extension, 6 times better than MSP430

Going further: Full compressive DAC



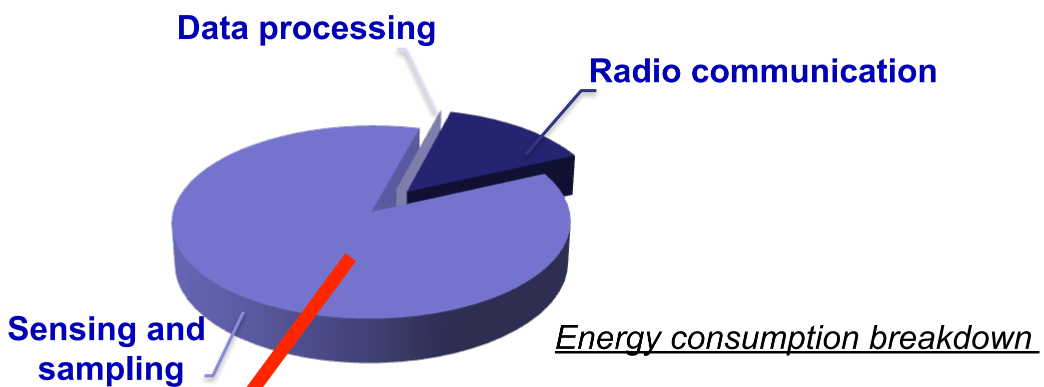


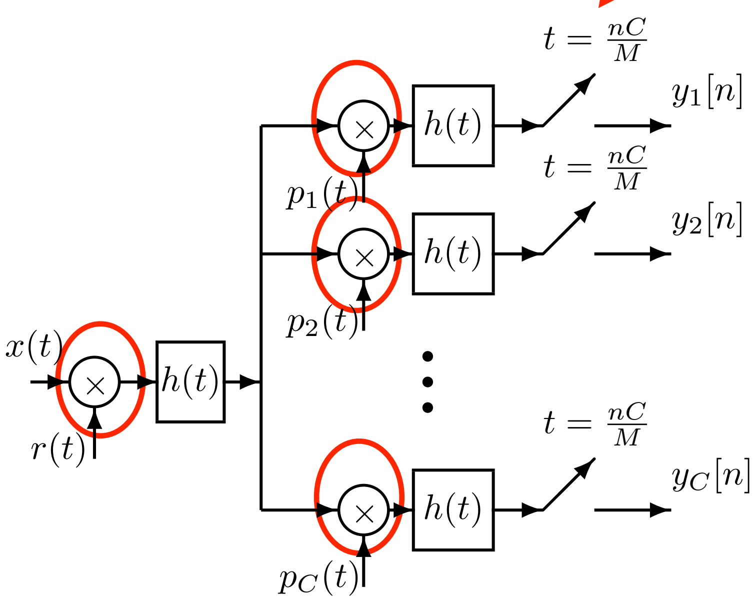
$$x(t) = \sum_{n=1}^{N} \alpha_n \psi_n(t), \qquad \text{for } t \in [0,1).$$

$$y[m] = \sum_{n=1}^{N} \alpha_n \int_{\frac{m}{M}}^{\frac{(m+1)}{M}} \psi_n(\tau) P_c(\tau) h(\frac{m}{M} - \tau) d\tau.$$

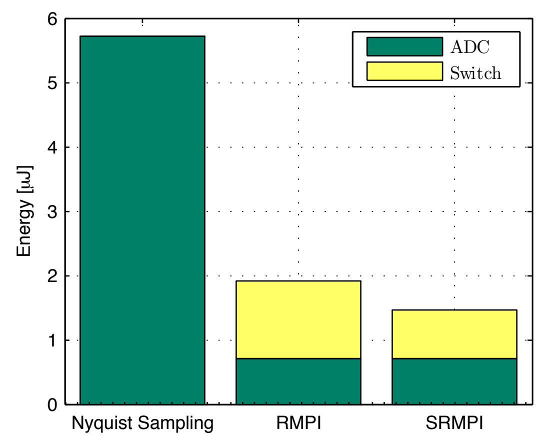
$$\min_{\tilde{\boldsymbol{\alpha}} \in \mathbb{R}^N} \ ||\tilde{\boldsymbol{\alpha}}||_1$$
$$||\boldsymbol{\Phi}\boldsymbol{\Psi}\tilde{\boldsymbol{\alpha}} - \mathbf{y}||_2 \leq \sigma$$

Going further: Full compressive DAC





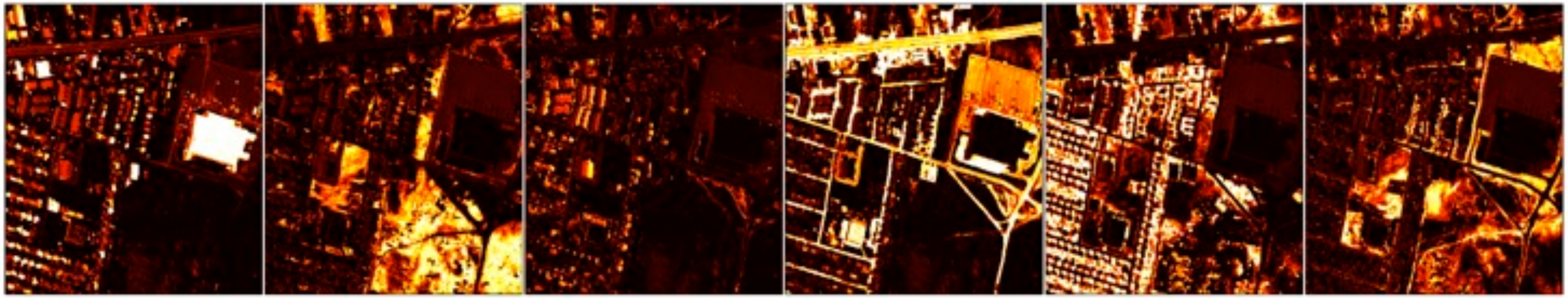
With off-the-shelf components and
a simple CS-based AFE:



- Big advantage of direct samples
  - they are easily interpreted
  - they can be processed
- Can we process signals in the compressed domain?
  - detection, recognition, classification
  - segmentation

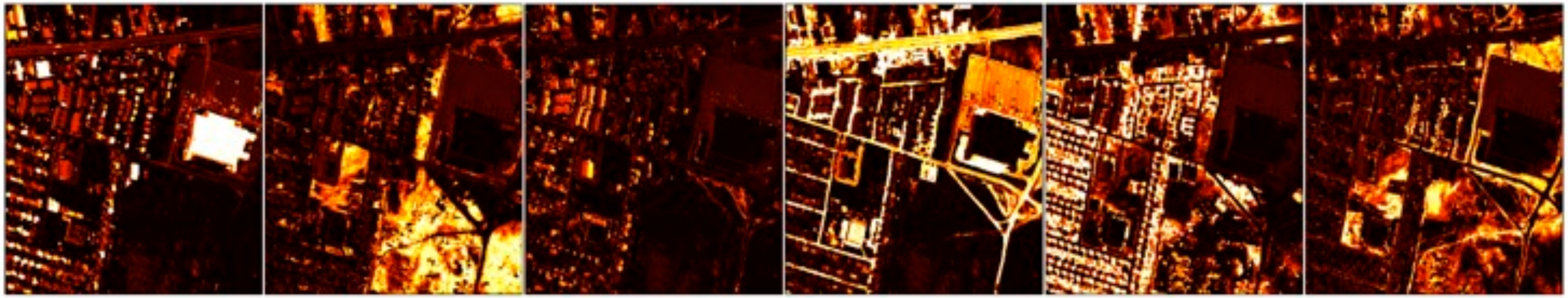
Hyper Spectral Imaging

S: Sources (element abundancies) $\mathbf{S} \in \mathbb{R}^{n_1 \times \rho}$

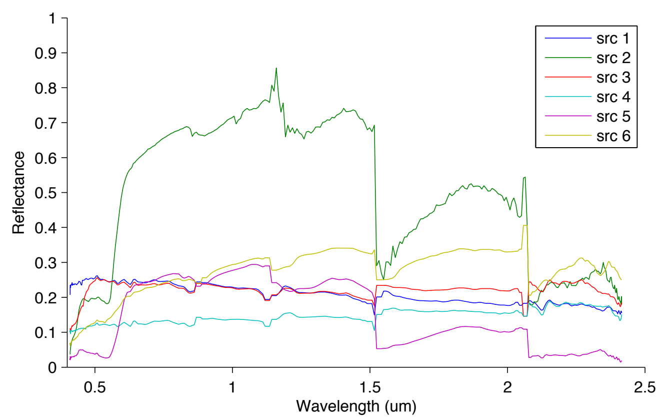


Hyper Spectral Imaging

$\mathbf{S}$: Sources (element abundancies) $\mathbf{S} \in \mathbb{R}^{n_1 \times \rho}$

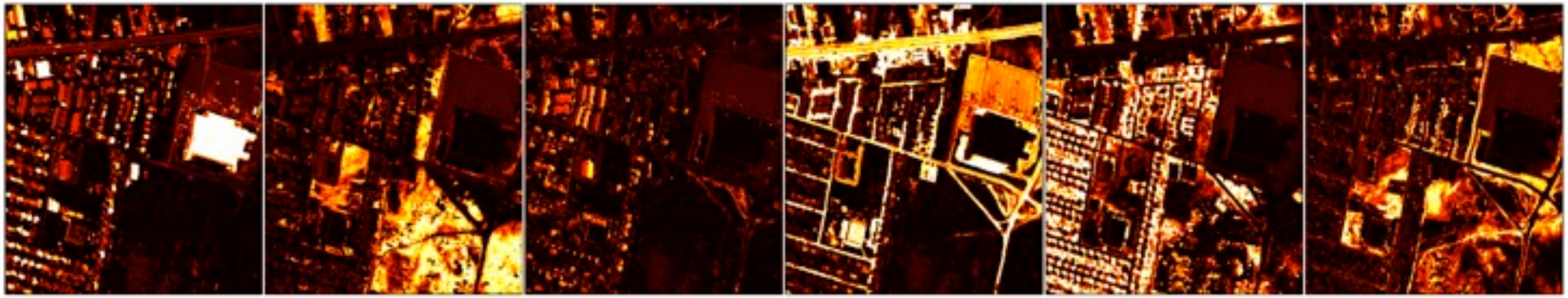


$\mathbf{A}$: Spectra (depending on modality) $\mathbf{A} \in \mathbb{R}^{n_2 \times \rho}$

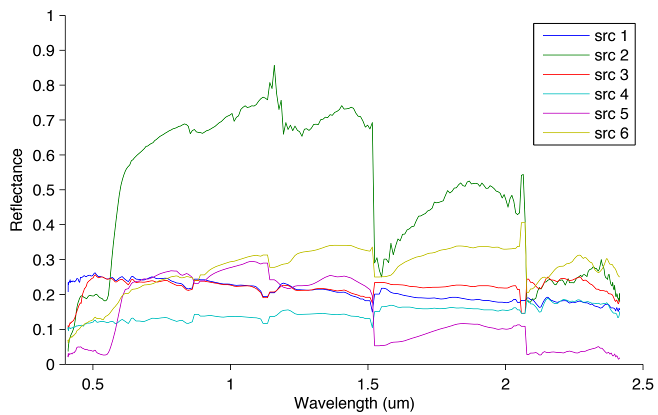


Hyper Spectral Imaging

$\mathbf{S}$: Sources (element abundancies) $\mathbf{S} \in \mathbb{R}^{n_1 \times \rho}$



$\mathbf{A}$: Spectra (depending on modality) $\mathbf{A} \in \mathbb{R}^{n_2 \times \rho}$



Each pixel is a weighted combination of source spectra: $y = \mathbf{S}\mathbf{A}^T$

Hyper Spectral Imaging

Very high dimensional data (thousands of channels)

Typical problem: Source Separation

Given the dictionary of spectra $\mathbf{A}$ and the data y

Recover the source abundances, factorizing $y = \mathbf{S}\mathbf{A}^T$

Hyper Spectral Imaging

Very high dimensional data (thousands of channels)

Typical problem: Source Separation

Given the dictionary of spectra $\mathbf{A}$ and the data y

Recover the source abundances, factorizing $y = \mathbf{S}\mathbf{A}^T$

Can this be achieved even when:

$y = \mathcal{A}(\mathbf{S}\mathbf{A}^T)$ Indirect/degraded observations

The dictionary of spectra $\mathbf{A}$ is unknown

Hyper Spectral Imaging

HSI Compressive Blind Source Separation (CS-BSS)

$$\arg\min_{\mathbf{S},\mathbf{A}} \quad \|y - \mathcal{A}(\mathbf{S}\mathbf{A}^T)\|_{\ell_2}^2$$

subject to

$$\sum_j^{\rho} \|\mathbf{S}_j\|_{TV} \leq \tau$$
$$\sum_{i=1}^{\rho} [\mathbf{S}]_{i,j} = 1$$
$$[\mathbf{S}]_{i,j} \geq 0$$

Source image constraints

$$\|\Psi_{1D}^T \mathbf{A}\|_{\ell_1} \leq \gamma$$
$$[\mathbf{A}]_{i,j} \geq 0.$$

Spectral signature constraints

- Bi-convex minimization

Algorithm: alternating convex minimization

1- Initialize A at random

2- Source recovery given A

3- Mixture recovery given S

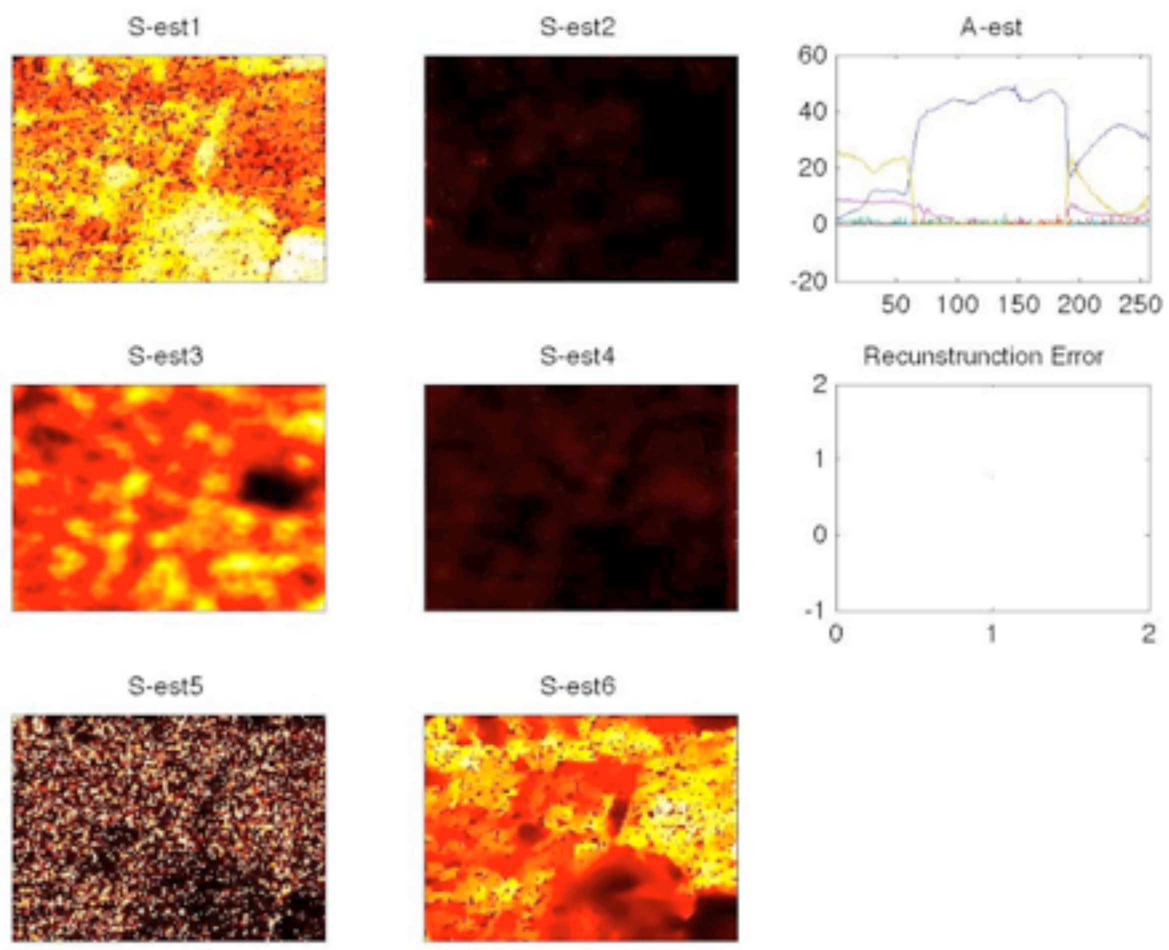
4- Repeat 2-3 until convergence

Goal: Compute source maps directly from compressed measurements

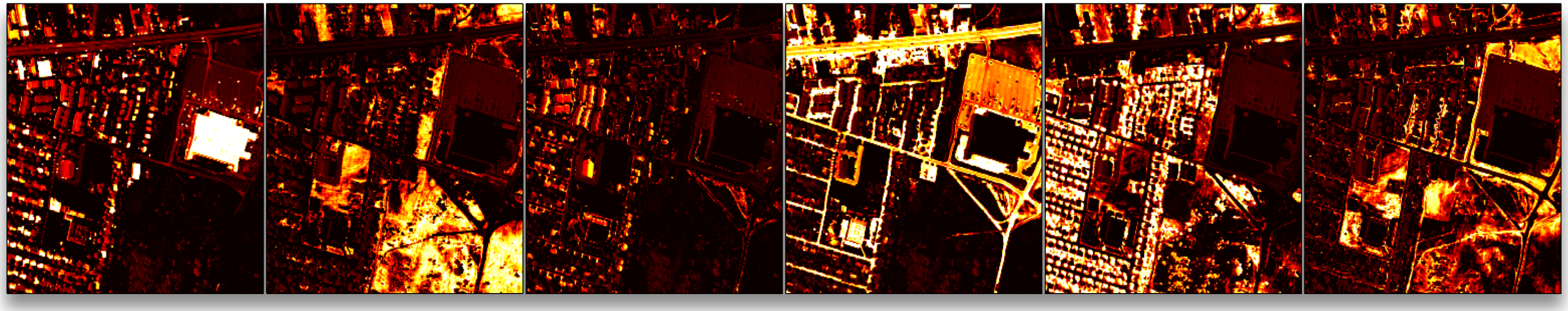
Separation or Segmentation in the compressed domain

Hyper-Spectral Imaging

Hyper-Spectral Imaging

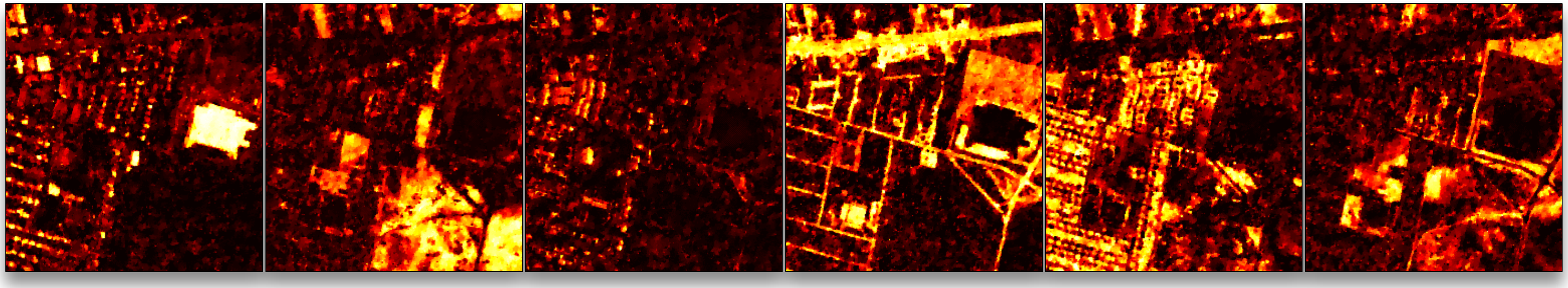


Hyper-Spectral Imaging



(a) Ground truth

From 3% of the original data:

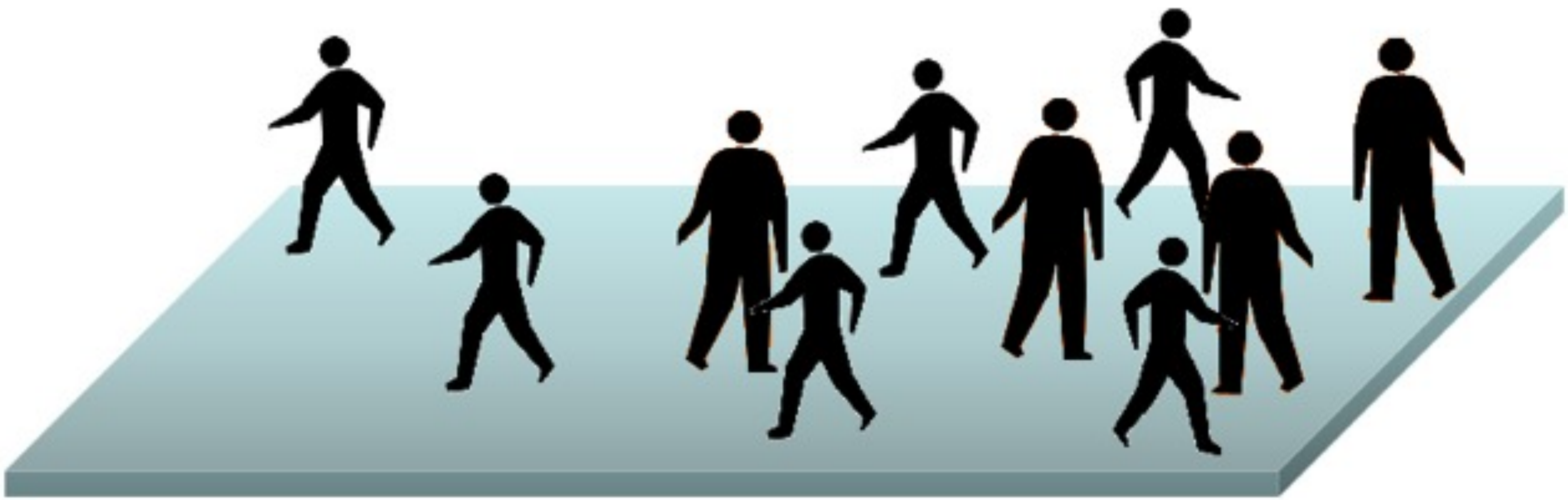


(e) SS-TV-decorr, source reconstruction SNR: 8.64 dB

Other applications: Mass spectrometry (MALDI), ...

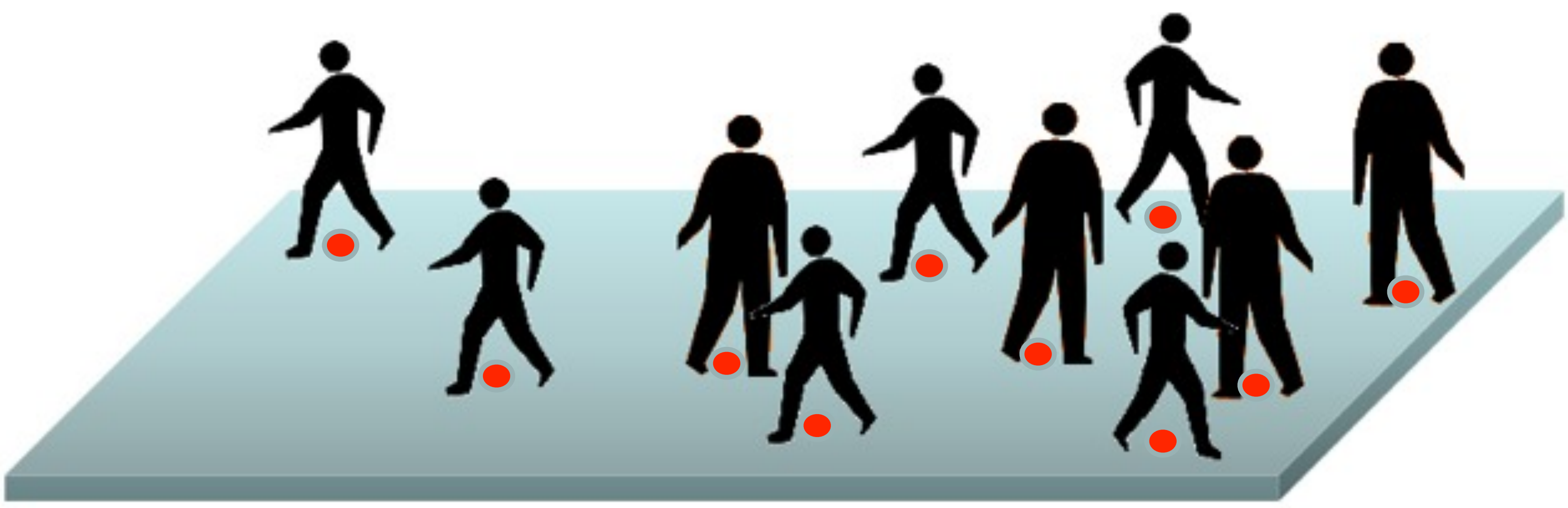
Detecting and Following People

Detecting and Following People



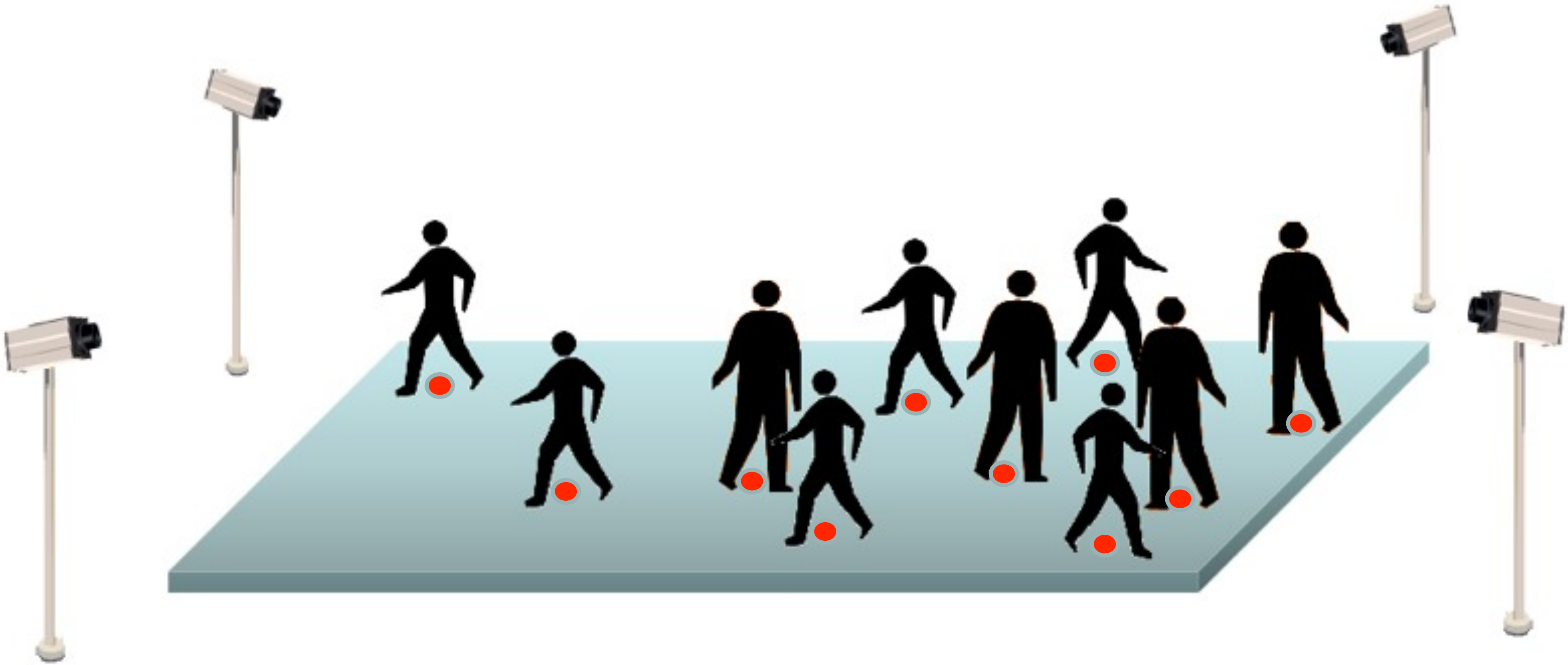
Detecting and Following People

- Locate group of people occluding each other



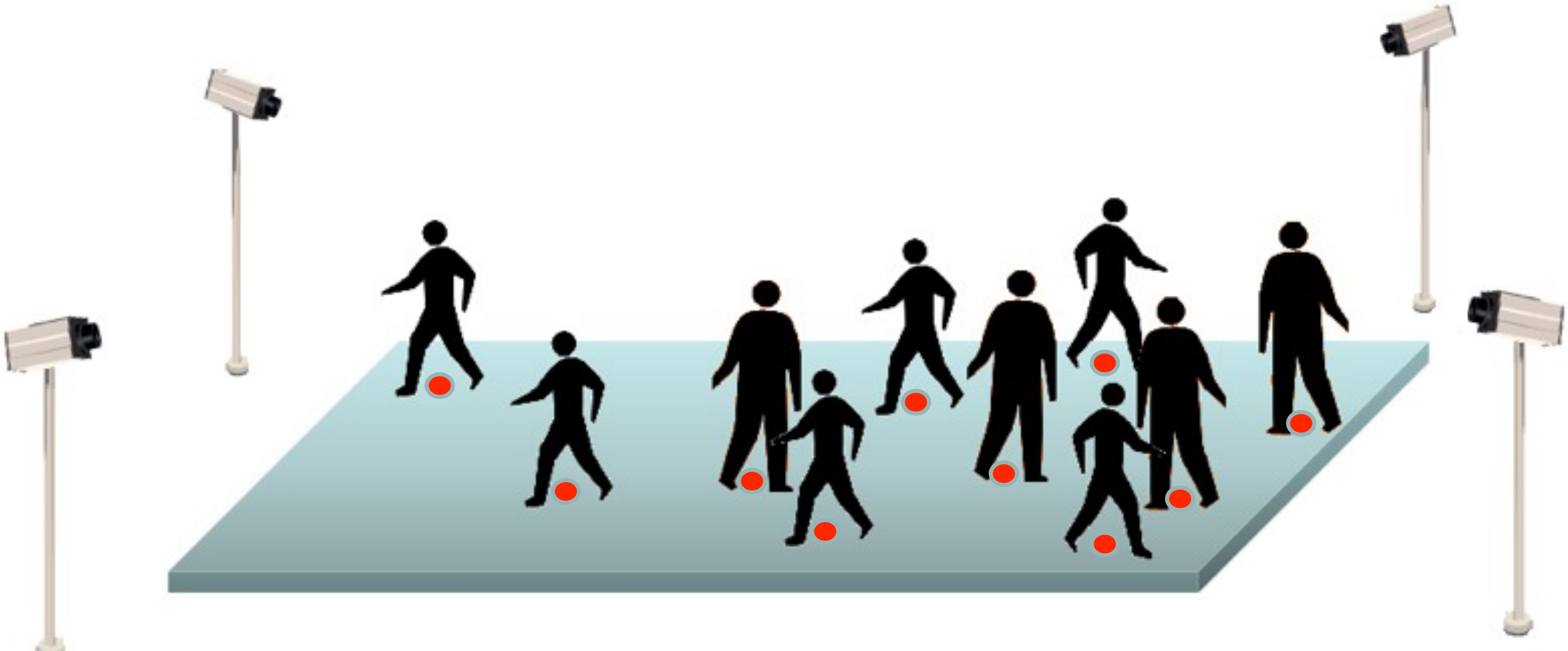
Detecting and Following People

- Locate group of people occluding each other
- With a network of cameras



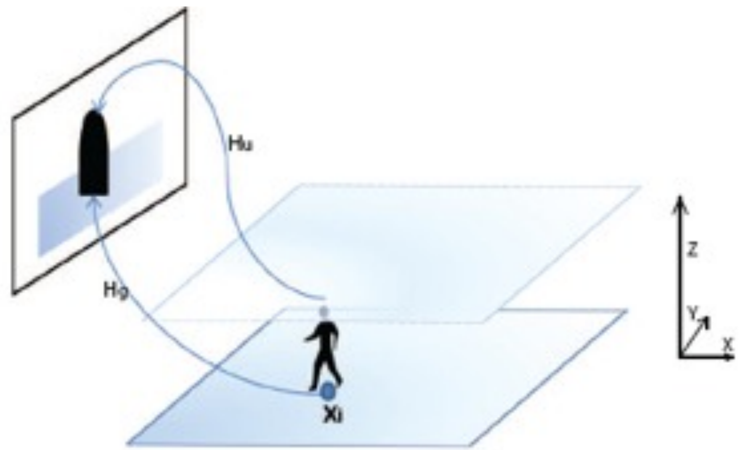
Detecting and Following People

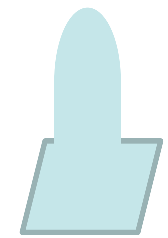
- Locate group of people occluding each other
- With a network of cameras
- Given extracted foreground silhouettes only



Problem Formulation

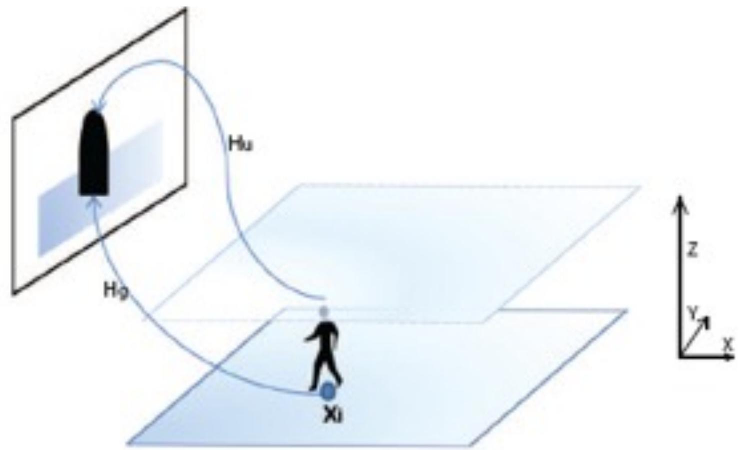
- Use a dictionary D associating to x a certain configuration of silhouettes in y

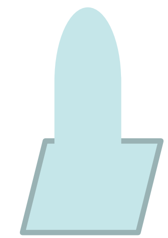




Problem Formulation

- Use a dictionary D associating to x a certain configuration of silhouettes in y



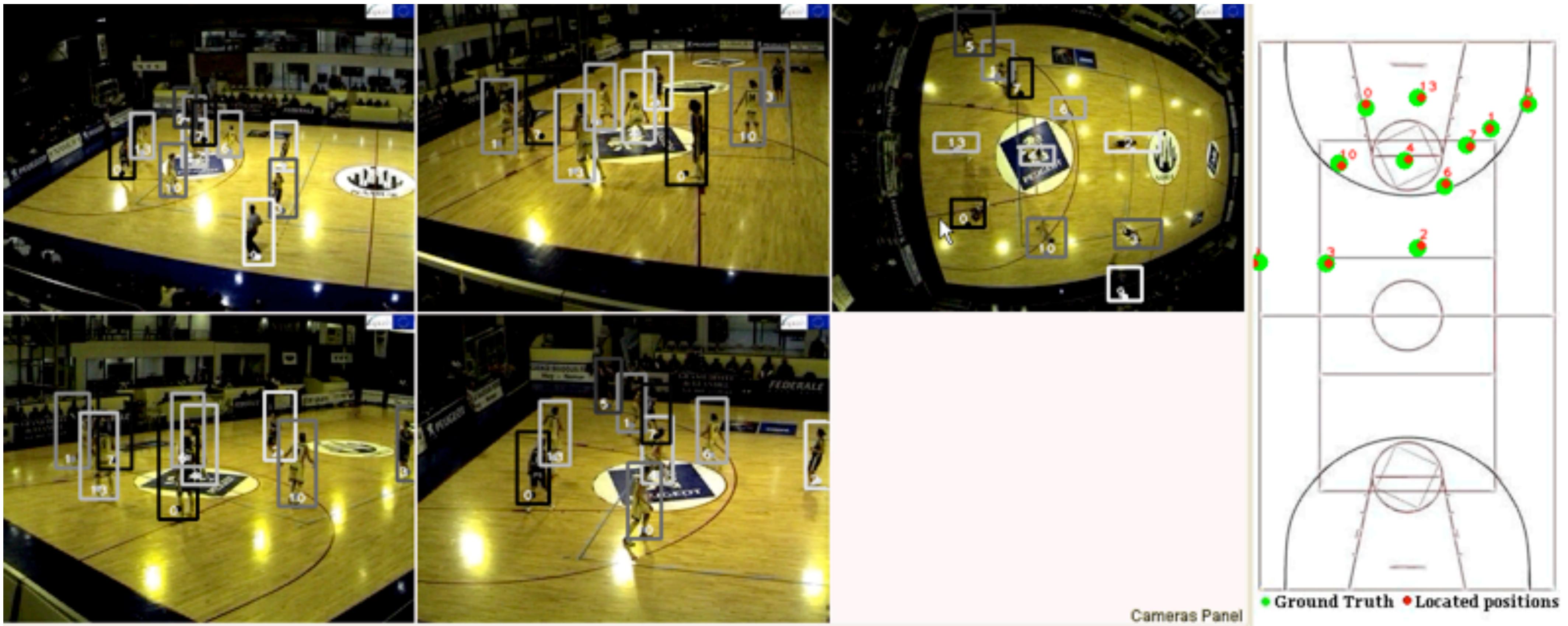


- We want:

$$\arg\min_{x \in \{0,1\}^N} \|x\|_0 \quad \text{s.t.} \quad \|y - Q(Dx)\|_2^2 < \varepsilon$$

y - $Q(Dx)$

Results



Alexandre Alahi, Mohammad Golbabaee

Outlook

- Significant challenges ahead in signal processing
  - Big Data
  - Ubiquitous but Cheap Sensing (i.e dirty signals)
- Need new models, algorithms
  - Hardware/Software co-design
  - New sensor designs
- Where do we go from here in terms of applications ?
  - Structured notions of sparsity for specific applications
  - More data driven approaches
  - Non-linearities ?

Application focus

- Big Data
  - Too big for sparsity ?
- What can we hope from Sparsity+Machine Learning?
  - Better algos ?
  - Better models ?
- What new application fields ?
  - Beyond restoration ?
- What's missing ?
  - Time/Space variant operators
  - Non-Linearities