Short-course
Compressive Sensing of Videos

Venue
CVPR 2012, Providence, RI, USA
June 16, 2012

Organizers:
Richard G. Baraniuk
Mohit Gupta
Aswin C. Sankaranarayanan
Ashok Veeraraghavan
Part 2: Compressive sensing
Motivation, theory, recovery

• Linear inverse problems

• Sensing visual signals

• Compressive sensing
  – Theory
  – Hallmark
  – Recovery algorithms

• Model-based compressive sensing
  – Models specific to visual signals
Linear inverse problems
Linear inverse problems

• Many classic problems in computer can be posed as linear inverse problems

• Notation
  – **Signal** of interest \( x \in \mathbb{R}^N \)
  – **Observations** \( y \in \mathbb{R}^M \)
  – Measurement model \( y = \Phi x + e \)

• Problem definition: given \( y \), recover \( x \)
Linear inverse problems

\[ y = \Phi x + e \]

\[ x \in \mathbb{R}^N \]
\[ y \in \mathbb{R}^M \]

- Problem definition: given \( y \), recover \( x \)

- **Scenario 1**
  \[ M \geq N \]
  \[ \hat{x} = \Phi^{-1} y \]

- We can invert the system of equations

- Focus more on **robustness** to noise via signal priors
Linear inverse problems

\[ y = \Phi x + e \]

\[ x \in \mathbb{R}^N \]
\[ y \in \mathbb{R}^M \]

- Problem definition: given \( y \), recover \( x \)

- Scenario 2

- Measurement matrix has a \((N-M)\) dimensional null-space

- Solution is no longer unique

- Many interesting vision problem fall under this scenario

- Key quantity of concern: Under-sampling ratio \( M/N \)
Image super-resolution

Low resolution input/observation

128x128 pixels
Image super-resolution

\[ y = \Phi x + e \]

\[ y(1,1) = (x_{1,1} + x_{1,2} + x_{2,1} + x_{2,2}) / 4 \]
Image super-resolution

$$y = \Phi x + e$$

$$y_{(1,1)} = (x_{1,1} + x_{1,2} + \ldots + x_{4,3} + x_{4,4}) / 16$$
Image super-resolution

\[ y = \Phi x + e \]

Super-resolution factor \( D \)

Under-sampling factor \( M/N = 1/D^2 \)

**General rule:**
The smaller the under-sampling, the more the unknowns and hence, the harder the super-resolution problem
Many other vision problems...

- Affine rank minimization, Matrix completion, deconvolution, Robust PCA

- Image synthesis
  - Infilling, denoising, etc.

- Light transport
  - Reflectance fields, BRDFs, Direct global separation, Light transport matrices

- Sensing
Sensing visual signals
High-dimensional visual signals

- Reflection
- Refraction
- Fog: Volumetric scattering
- Human skin: Sub-surface scattering
- Electron microscopy Tomography
The plenoptic function

Collection of all variations of light in a scene

Different slices reveal different scene properties
The plenoptic function

- **space** (3D)
- **time** (1D)
- **spectrum** (1D)
- **angle** (2D)

High-speed cameras

Lytro light-field camera

Hyper-spectral imaging

Hyper-spectral imaging
Sensing the plenoptic function

• **High-dimensional**
  – 1000 samples/dim == $10^{21}$ dimensional signal
  – Greater than all the *storage* in the world

• Traditional theories of sensing fail us!
Resolution trade-off

- Key enabling factor: **Spatial resolution is cheap!**

- Commercial cameras have 10s of megapixels

- One idea is the we trade-off spatial resolution for resolution in some other axis
Spatio-angular tradeoff

[Ng, 2005]
Spatio-angular tradeoff

[Levoy et al. 2006]
Spatio-temporal tradeoff

Stagger pixel-shutter within each exposure

[Bub et al., 2010]
Rearrange to get high temporal resolution video at lower spatial-resolution

[Bub et al., 2010]
Resolution trade-off

• Very powerful and **simple idea**

• **Drawbacks**
  – Does not extend to **non-visible** spectrum
    • 1 Megapixel SWIR camera costs 50-100k
  – Linear and global tradeoffs
  – With today’s technology, cannot obtain more than 10x for video without sacrificing spatial resolution completely
Compressive sensing
Sense by Sampling

\[ x \rightarrow \text{sample} \rightarrow N \]
Sense by *Sampling*

\[ x \rightarrow \text{sample} \rightarrow N \]

too much data!
Sense then *Compress*

\[ x \rightarrow \text{sample} \rightarrow N \rightarrow \text{compress} \rightarrow K \rightarrow \text{decompress} \rightarrow N \rightarrow \hat{x} \]

JPEG
JPEG2000

...
Sparsity

\( N \) pixels

\( K \ll N \)

large wavelet coefficients

(bleu = 0)
Sparsity

$N$ pixels

$K \ll N$
large wavelet coefficients
(blue = 0)

$N$ wideband
signal samples

$K \ll N$
large Gabor (TF) coefficients

$N$ pixels

$K \ll N$
large wavelet coefficients
(blue = 0)

$N$ wideband
signal samples

$K \ll N$
large Gabor (TF) coefficients
Concise Signal Structure

- **Sparse** signal: only $K$ out of $N$ coordinates nonzero
Concise Signal Structure

- **Sparse** signal: only $K$ out of $N$ coordinates nonzero
  - model: union of $K$-dimensional subspaces aligned w/ coordinate axes
Concise Signal Structure

- **Sparse** signal: only $K$ out of $N$ coordinates nonzero
  - model: union of $K$-dimensional subspaces

- **Compressible** signal: sorted coordinates decay rapidly with power-law
Concise Signal Structure

- **Sparse** signal: only $K$ out of $N$ coordinates nonzero
  - model: union of $K$-dimensional subspaces

- **Compressible** signal: sorted coordinates decay rapidly with power-law
  - model: $\ell_p$ ball: $\|x\|_p^p = \sum_i |x_i|^p \leq 1, \ p \leq 1$
What’s Wrong with this Picture?

- Why go to all the work to acquire $N$ samples only to discard all but $K$ pieces of data?
What’s Wrong with this Picture?

**linear** processing

**linear** signal model (bandlimited subspace)

**nonlinear** processing

**nonlinear** signal model (union of subspaces)
Compressive Sensing

- Directly acquire “compressed” data via dimensionality reduction
- Replace samples by more general “measurements”

\[ K \approx M \ll N \]
Sampling

- Signal $x$ is $K$-sparse in basis/dictionary $\Psi$
  - WLOG assume sparse in space domain $\Psi = I$
• Signal $x$ is $K$-sparse in basis/dictionary $\Psi$  
  – WLOG assume sparse in space domain $\Psi = I$

• Sampling

\[
\begin{align*}
N \times 1 \quad &\text{measurements} \\
\begin{array}{c}
\text{y} \\
\Phi = I \\
\text{x}
\end{array} \\
\begin{array}{c}
N \times 1 \\
\text{sparse} \\
\text{signal}
\end{array}
\end{align*}
\]

\[K \quad \text{nonzero entries}\]
Compressive Sampling

• When data is sparse/compressible, can directly acquire a **condensed representation** with no/little information loss through linear **dimensionality reduction**

\[ y = \Phi x \]

\[ M \times 1 \quad \text{measurements} \]
\[ \Phi \]
\[ M \times N \]
\[ x \]
\[ N \times 1 \quad \text{sparse signal} \]
\[ K \quad \text{nonzero entries} \]

\[ K < M \ll N \]
How Can It Work?

- Projection $\Phi$ **not full rank**...

$M < N$

... and so **loses information** in general

- Ex: Infinitely many $x$’s map to the same $y$ (null space)
How Can It Work?

• Projection $\Phi$ not full rank...

$$M < N$$

... and so loses information in general

• But we are only interested in **sparse** vectors
How Can It Work?

• Projection $\Phi$ not full rank...

\[ M < N \]

... and so loses information in general

• But we are only interested in \textit{sparse} vectors

• $\Phi$ is effectively $M\times K$
How Can It Work?

- Projection $\Phi$ not full rank...

$M < N$

... and so loses information in general

- But we are only interested in \textit{sparse} vectors

- \textbf{Design} $\Phi$ so that each of its $M \times K$ submatrices are full rank (ideally close to orthobasis)
  
  - Restricted Isometry Property (RIP)
Restricted Isometry Property (RIP)

- Preserve the structure of sparse/compressible signals
Restricted Isometry Property (RIP)

- “Stable embedding”
- RIP of order $2K$ implies: for all $K$-sparse $x_1$ and $x_2$

\[
(1 - \delta_{2K}) \leq \frac{\|\Phi x_1 - \Phi x_2\|_2^2}{\|x_1 - x_2\|_2^2} \leq (1 + \delta_{2K})
\]
RIP = Stable Embedding

- An information preserving projection $\Phi$ preserves the geometry of the set of sparse signals.

- RIP ensures that

$$\| x_1 - x_2 \|_2 \approx \| \Phi x_1 - \Phi x_2 \|_2$$
How Can It Work?

- Projection $\Phi$ not full rank...

  $M < N$

  ... and so loses information in general

- **Design** $\Phi$ so that each of its $M \times K$ submatrices are full rank (RIP)

- Unfortunately, a combinatorial, **NP-complete design problem**
Insight from the 70’s [Kashin, Gluskin]

- Draw $\Phi$ at **random**
  - iid Gaussian
  - iid Bernoulli $\pm 1$
  ...

- Then $\Phi$ has the RIP with high probability provided

$$M = O(K \log(N/K)) \ll N$$
Randomized Sensing

- Measurements = random linear combinations of the entries of the signal

\[ y = \Phi x \]

- No information loss for sparse vectors whp

\[ M \times 1 \text{ measurements} = \begin{bmatrix} M \times N \end{bmatrix} \]

\[ N \times 1 \text{ sparse signal} \]

\[ M = O(K \log(N/K)) \]
CS Signal Recovery

- **Goal**: Recover signal $x$ from measurements $y$

- **Problem**: Random projection $\Phi$ not full rank (ill-posed inverse problem)

- **Solution**: Exploit the sparse/compressible geometry of acquired signal $x$
CS Signal Recovery

• Random projection $\Phi$ not full rank

• Recovery problem:
given $y = \Phi x$
find $x$

• Null space

• Search in null space for the “best” $x$
according to some criterion
  – ex: least squares

$(N-M)$-dim hyperplane at random angle
\[ \ell_2 \text{ Signal Recovery} \]

- **Recovery:**
  
  \[
  \text{given } y = \Phi x \]
  
  \[
  \text{find } x \text{ (sparse)}
  \]

- **Optimization:**

  \[
  \hat{x} = \arg \min_{y=\Phi x} \|x\|_2
  \]

- **Closed-form solution:**

  \[
  \hat{x} = (\Phi^T \Phi)^{-1} \Phi^T y
  \]
**L₂ Signal Recovery**

**Recovery:**
given \( y = \Phi x \)
find \( x \) (sparse)

**Optimization:**
\[
\hat{x} = \arg \min_{y=\Phi x} \|x\|_2
\]

**Closed-form solution:**
\[
\hat{x} = (\Phi^T \Phi)^{-1} \Phi^T y
\]

**Wrong answer!**
\( l_2 \) Signal Recovery

- **Recovery:**
  (ill-posed inverse problem)

- **Optimization:**

- **Closed-form solution:**

- **Wrong answer!**

\[
given \quad y = \Phi x \\
find \quad x \text{ (sparse)}
\]

\[
\hat{x} = \arg \min_{y = \Phi x} ||x||_2
\]

\[
\hat{x} = (\Phi^T \Phi)^{-1} \Phi^T y
\]
\( \ell_0 \) Signal Recovery

- **Recovery:**
  (ill-posed inverse problem)
  
  \[
  \text{given} \quad y = \Phi x \\
  \text{find} \quad \hat{x} \quad \text{(sparse)}
  \]

- **Optimization:**

  \[
  \hat{x} = \arg \min_{y=\Phi x} \|x\|_0
  \]

  “find \textbf{sparsest} vector in translated nullspace”

\[ \mathbb{R}^N \]
\( l_0 \) Signal Recovery

- **Recovery:**
  (ill-posed inverse problem)
  
  \[
  \text{given } \ y = \Phi x \\
  \text{find } \ x \text{ (sparse)}
  \]

- **Optimization:**
  \[
  \hat{x} = \arg \min_{y=\Phi x} \| x \|_0
  \]
  “find sparsest vector in translated nullspace”

- **Correct!**

- **But NP-Complete alg**
\( \ell_1 \) Signal Recovery

- Recovery: given \( y = \Phi x \) find \( x \) (sparse)
- Optimization: \( \hat{x} = \operatorname{arg\,min}_{y = \Phi x} \|x\|_1 \)
- **Convexify** the \( \ell_0 \) optimization

- Candes
- Romberg
- Tao
- Donoho
$\ell_1$ Signal Recovery

- **Recovery:**
  (ill-posed inverse problem)
  given $y = \Phi x$
  find $x$ (sparse)

- **Optimization:**
  $\hat{x} = \arg \min_{y=\Phi x} \|x\|_1$

- **Convexify** the $\ell_0$ optimization

- **Correct!**

- **Polynomial time** alg
  (linear programming)
Compressive Sensing

$\begin{align*}
M \times 1 & \quad \Phi \\
\text{random measurements} & \quad N \times 1 \\
M \times N & \\
K & \quad \text{nonzero entries}
\end{align*}$

• Signal recovery via $\ell_1$ optimization

$\hat{x} = \arg\min_{y=\Phi x} \|x\|_1$
Compressive Sensing

\[ y = \Phi x \]

- Signal **recovery** via iterative greedy algorithm
  - (orthogonal) matching pursuit \[\text{[Gilbert, Tropp]}\]
  - iterated thresholding
    \[\text{[Nowak, Figueiredo; Kingsbury, Reeves; Daubechies, Defrise, De Mol; Blumensath, Davies; …]}\]
  - CoSaMP \[\text{[Needell and Tropp]}\]
Greedy recovery algorithm #1

- Consider the following problem

\[
\min \|y - \Phi x\|_2 \\
\text{s.t} \\
\|x\|_0 \leq K
\]

- Suppose we wanted to minimize just the cost, then steepest gradient descent works as

\[
\hat{x}_k = \hat{x}_{k-1} + \eta\Phi^T (y - \Phi \hat{x}_{k-1})
\]

- But, the new estimate is no longer K-sparse

\[
\hat{x}_k = \text{thresh} \left( \hat{x}_{k-1} + \eta\Phi^T (y - \Phi \hat{x}_{k-1}), K \right)
\]
Iterated Hard Thresholding

goal: given $y = \Phi x$, recover a sparse $x$
initialize: $\hat{x}_0 = 0$, $r = y$, $i = 0$
iteration:

- $i \leftarrow i + 1$
- $b \leftarrow \hat{x}_{i-1} + \Phi^T r$ \hspace{1cm} \textbf{update signal estimate}$\$
- $\hat{x}_i \leftarrow \text{thresh}(b, K)$ \hspace{1cm} \textbf{prune signal estimate (best } K \text{-term approx)}$
- $r \leftarrow y - \Phi \hat{x}_i$ \hspace{1cm} \textbf{update residual}$\$

return: $\hat{x} \leftarrow \hat{x}_i$
Greedy recovery algorithm #2

• Consider the following problem

\[ y = \Phi x \]

\[ M \times N \]

\[ N \times 1 \]

sparse signal

• Can we recover the support?
Greedy recovery algorithm #2

• Consider the following problem

\[
\begin{align*}
y &= \Phi x \\
M \times N &\quad \text{sparse signal} \\
N \times 1 &\quad 1 \text{ sparse signal}
\end{align*}
\]

• If \( \Phi = [\phi_1, \phi_2, \ldots, \phi_N] \)

then \( \arg \max |\langle \phi_i, y \rangle| \) gives the support of \( x \)

• How to extend to \( K \)-sparse signals?
Greedy recovery algorithm #2

\[ y = \Phi x \]

\( N \times 1 \) sparse signal

\( M \times N \) sparse

residue:

\[ r = y - \Phi \hat{x}_{k-1} \]

find atom:

\[ k = \arg \max | \langle \phi_i, r \rangle | \]

Add atom to support:

\[ S = S \cup \{k\} \]

Signal estimate (Least squares over support)

\[ x_k = (\Phi_S)\dagger y \]
Orthogonal matching pursuit

goal:
given \( y = \Phi x \), recover a sparse \( x \)
columns of \( \Phi \) are unit-norm

initialize: \( \hat{x}_0 = 0, r = y, \Lambda = \{\}, i = 0 \)

iteration:
\( i = i + 1 \)
\( b = \Phi^T r \)
\( k = \arg \max \{|b(1)|, |b(2)|, \ldots, |b(N)|\} \)
\( \Lambda = \Lambda \cup k \)
\( (\hat{x}_i)|_{\Lambda} = (\Phi|_{\Lambda})^\dagger y, (\hat{x}_i)|_{\Lambda^c} = 0 \)
\( r = y - \Phi \hat{x}_i \)

Find atom with largest support
Update signal estimate
update residual
Specialized solvers

- **CoSAMP** [Needell and Tropp, 2009]

- **SPG\_l1** [Friedlander, van der Berg, 2008]

- **FPC** [Hale, Yin, and Zhang, 2007]
  [http://www.caam.rice.edu/~optimization/L1/fpc/](http://www.caam.rice.edu/~optimization/L1/fpc/)

- **AMP** [Donoho, Montanari and Maleki, 2010]

many many others, see [dsp.rice.edu/cs](http://dsp.rice.edu/cs) and [https://sites.google.com/site/igorcarron2/cscodes](https://sites.google.com/site/igorcarron2/cscodes)
CS Hallmarks

- **Stable**
  - acquisition/recovery process is numerically stable

- **Asymmetrical** (most processing at decoder)
  - conventional: smart encoder, dumb decoder
  - CS: dumb encoder, smart decoder

- **Democratic**
  - each measurement carries the same amount of information
  - robust to measurement loss and quantization
  - “digital fountain” property

- Random measurements **encrypted**

- **Universal**
  - same random projections / hardware can be used for *any* sparse signal class (generic)
Universality

- Random measurements can be used for signals sparse in any basis

\[ x = \Psi \alpha \]
Universality

- Random measurements can be used for signals sparse in any basis

\[ y = \Phi x = \Phi \Psi \alpha \]
Universality

- Random measurements can be used for signals sparse in any basis

\[ y = \Phi x = \Phi \Psi \alpha = \Phi' \alpha \]
Summary: CS

- **Compressive sensing**
  - randomized dimensionality reduction
  - exploits signal **sparsity** information
  - integrates sensing, compression, processing

- Why it works: with high probability, random projections preserve information in signals with concise geometric structures
  - sparse signals
  - compressible signals
Summary: CS

- **Encoding:** \( y = \text{random linear combinations of the entries of } x \)
  \[ M \times 1 \text{ measurements} = \Phi \]
  \[ M \times N \]
  \[ M = O(K \log(N/K)) \]

- **Decoding:** Recover \( x \) from \( y \) via optimization
  \[ N \times 1 \text{ sparse signal} \]
  \[ K \text{ nonzero entries} \]
Image/Video specific signal models and recovery algorithms
Transform basis

- Recall Universality: Random measurements can be used for signals sparse in *any* basis

$$x \Psi \alpha$$

- DCT/FFT/Wavelets ...
  - Fast transforms; very useful in large scale problems
Dictionary learning

• For many signal classes (ex: videos, light-fields), there are **no** obvious sparsifying transform basis

• Can we *learn* a sparsifying transform instead?

• **GOAL:** Given training data \( x_1, x_2, \ldots, x_T \)

  learn a “dictionary” \( D \), such that

  \[
  x_i = D s_i
  \]

  \( s_i \) are **sparse**.
Dictionary learning

- GOAL: Given training data $x_1, x_2, \ldots, x_T$
  
  learn a "dictionary" $D$, such that
  
  $$x_i = D s_i$$
  
  $s_i$ are sparse.

$$\min_{D,S} \|X - DS\|_F$$

s.t.

$$\forall i, \|s_i\|_0 \leq K$$
Dictionary learning

\[
\min_{D,S} \|X - DS\|_F \quad \text{s.t.} \quad \forall i, \|s_i\|_0 \leq K
\]

- Non-convex constraint
- Bilinear in \(D\) and \(S\)
Dictionary learning

\[
\min_{D,S} \| X - DS \|_F + \lambda \sum_k \| s_k \|_1
\]

- Biconvex in \(D\) and \(S\)
  - Given \(D\), the optimization problem is convex in \(s_k\)
  - Given \(S\), the optimization problem is a least squares problem

- **K-SVD**: Solve using alternate minimization techniques
  - Start with \(D = \) wavelet or DCT bases
  - Additional pruning steps to control size of the dictionary

Aharon et al., TSP 2006
Dictionary learning

\[ \min_{D,S} \| X - DS \|_F + \lambda \sum_k \| s_k \|_1 \]

- **Pros**
  - Ability to handle arbitrary domains

- **Cons**
  - Learning dictionaries can be computationally intensive for high-dimensional problems; need for very large amount of data
  - Recovery algorithms may suffer due to lack of fast transforms
Models on image gradients

- Piecewise constant images
  - Sparse image gradients

- Natural image statistics
  - Heavy tailed distributions
Total variation prior

\[ TV(I) = \sum_{u,v} \| \nabla I(u, v) \|_2 = \sum_{u,v} \sqrt{I_x^2(u, v) + I_y^2(u, v)} \]

- TV norm
  - Sparse-gradient promoting norm

- Formulation of recovery problem

\[
\begin{align*}
\min & \quad TV(x) \\
\text{s.t} & \quad y = \Phi x
\end{align*}
\]
Total variation prior

- Optimization problem
  - **Convex**
  - Often, works “better” than transform basis methods

- Variants
  - 3D (video)
  - Anisotropic TV

- Code
  - TVAL3
  - Many many others (see dsp.rice/cs)

\[
\begin{align*}
\min & \ TV(x) \\
\text{s.t} & \ y = \Phi x
\end{align*}
\]
Beyond sparsity

Model-based CS
Beyond Sparse Models

• Sparse signal model captures _simplistic primary structure_

wavelets: natural images

Gabor atoms: chirps/tones

pixels: background subtracted images
Beyond Sparse Models

• Sparse signal model captures *simplistic primary structure*

• Modern compression/processing algorithms capture *richer secondary coefficient structure*

wavelets: natural images

Gabor atoms: chirps/tones

pixels: background subtracted images
Sparse Signals

- **$K$-sparse signals** comprise a particular set of $K$-dim subspaces
Structured-Sparse Signals

- A **K-sparse signal model** comprises a particular *(reduced)* set of *K*-dim subspaces  
  [Blumensath and Davies]

- Fewer subspaces  
  <> relaxed RIP  
  <> stable recovery using fewer measurements $M$
Wavelet Sparse

- Typical of wavelet transforms of natural signals and images (piecewise smooth)
Tree-Sparse

- **Model:** $K$-sparse coefficients + significant coefficients lie on a **rooted subtree**

- Typical of wavelet transforms of natural signals and images (piecewise smooth)
Wavelet Sparse

- **RIP:** stable embedding

\[ M = O(K \log(N/K)) \]
Tree-Sparse

- **Model:** $K$-sparse coefficients + significant coefficients lie on a rooted subtree

- **Tree-RIP:** stable embedding
  [Blumensath and Davies]

\begin{align*}
M &= O(K) < O(K \log(N/K))
\end{align*}
Tree-Sparse

• **Model:**  $K$-sparse coefficients
  + significant coefficients
  lie on a rooted subtree

• **Tree-RIP:** stable embedding
  [Blumensath and Davies]

• **Recovery:** inject tree-sparse approx into IHT/CoSaMP
Recall: Iterated Thresholding

goal: given $y = \Phi x$, recover a sparse $x$
initialize: $\hat{x}_0 = 0$, $r = y$, $i = 0$
iteration:

- $i \leftarrow i + 1$
- $b \leftarrow \hat{x}_{i-1} + \Phi^T r$ \textbf{update signal estimate}
- $\hat{x}_i \leftarrow \text{thresh}(b, K)$ \textbf{prune signal estimate}
  \hspace{1cm} (best $K$-term approx)
- $r \leftarrow y - \Phi \hat{x}_i$ \textbf{update residual}

return: $\hat{x} \leftarrow \hat{x}_i$
Iterated Model Thresholding

goal: given $y = \Phi x$, recover a sparse $x$
initialize: $\hat{x}_0 = 0$, $r = y$, $i = 0$
iteration:

- $i \leftarrow i + 1$
- $b \leftarrow \hat{x}_{i-1} + \Phi^T r$ \hspace{1cm} update signal estimate
- $\hat{x}_i \leftarrow M(b, K)$ \hspace{1cm} prune signal estimate
  (best $K$-term model approx)
- $r \leftarrow y - \Phi \hat{x}_i$ \hspace{1cm} update residual

return: $\hat{x} \leftarrow \hat{x}_i$
Tree-Sparse Signal Recovery

- Target signal
  - Tree-sparse CoSaMP (RMSE=0.037)
  - L1-minimization (RMSE=0.751)

Signal length
\(N=1024\)

Random measurements
\(M=80\)
Clustered Signals

- Probabilistic approach via **graphical model**

- Model **clustering of significant pixels** in space domain using Ising Markov Random Field

- Ising model approximation performed efficiently using **graph cuts**  
  \[\text{[Cevher, Duarte, Hegde, Baraniuk’08]}\]

![target](image1)
![Ising-model recovery](image2)
![CoSaMP recovery](image3)
![LP (FPC) recovery](image4)
Part 2: Compressive sensing
Motivation, theory, recovery

• Linear inverse problems

• Sensing visual signals

• Compressive sensing
  – Theory
  – Hallmark
  – Recovery algorithms

• Model-based compressive sensing
  – Models specific to visual signals