Lecture 23
Compressed Sensing
RADIOS

• https://inst.eecs.berkeley.edu/~ee123/sp15/radio.html
Q: What is the rate you need to sample at?
A: At least Nyquist!
Q: What is the rate you need to sample at?
A: Maybe less than Nyquist....
Images are compressible

Standard approach: First collect, then compress
Medical images are compressible
Standard approach: First collect, then compress
Medical images are compressible

Standard approach: First collect, then compress
Medical images are compressible
Standard approach: First collect, then compress

*Courtesy, M. Uecker, J Frahm Max Planck
Example 1: Audio

Raw audio: 44.1Khz, 16bit, stereo = 1378 Kbit/sec
MP3: 44.1Khz, 16bit, stereo = 128 Kbit/sec
10.76 fold!
Example II: Images

Raw image ( RGB ): 24 bit/pixel

JPEG : 1280x960, normal = 1.09 bit/pixel

22 fold!
Example III: Videos

Raw Video: (480x360)p x 24b/p x 24fps + 44.1Khz x 16b x 2 = 98,578 Kb/s

MPEG4 : 1300 Kb/s

75 fold!
Almost all compression algorithms use transform coding:

- mp3: DCT
- JPEG: DCT
- JPEG2000: Wavelet
- MPEG: DCT & time-difference
Sparse Transform

Signal

Sparse Transform

Quantization

Entropy encoding

DCT

Sorted coefficients
What sparsifying transform would you use here?

Sparse Transform

Signal

Sparse Transform

Difference

Quantization

Entropy encoding

Signal
Sparsity & Compressibility
Sparsity and Noise

sparse

not sparse

*M. Lustig, EECS UC Berkeley

*image courtesy of J. Trzasko
Sparsity and Noise

sparse

not sparse

denoise/separate by threshold

*image courtesy of J. Trzasko
AHA
SURE!
ON THE COUNT
OF THREE

THIS SPECTRUM IS NOISY
CAN YOU GIVE
ME A HAND?

YEP, JUST
A THRESHOLD

LET'S GET OUT OF
HERE BEFORE SOMEONE
SEES US

THAT WAS EASY
IT'S LIGHT
TOO

SPARSITY MAKES IT EASY TO SEPARATE
SIGNAL FROM NOISE
Transform Sparsity

not sparse

Sparse Edges
Transform Sparsity and Denoising

not sparse

sparse

- wavelet transform

low-frequency

high frequency

denoised

Transform Sparsity and Denoising

not sparse

sparse

wavelet transform

low-frequency

high frequency

denoised

Transform Sparsity and Denoising

wavelet denoising

More Sparse Transforms

*Video courtesy of Juan Santos, Heart Vista
Sparsity and Compression

- Only need to store non-zeros
From Samples to Measurements

• Shannon-Nyquist sampling
  – Worst case for ANY bandlimited data

• Compressive sampling (CS)
  “Sparse signals statistics can be recovered from a small number of non-adaptive linear measurements”
  – Integrated sensing, compression and processing.
  – Based on concepts of incoherency between signal and measurements
Traditional Sensing

• $x \in \mathbb{R}^N$ is a signal
• Make $N$ linear measurements

$$y = \Phi x$$

Desktop scanner/ digital camera sensing
Traditional Sensing

- $x \in \mathbb{R}^N$ is a signal
- Make N linear measurements

$$y = \Phi x$$

MRI Fourier Imaging

sensing matrix
Traditional Sensing

- $x \in \mathbb{R}^N$ is a signal
- Make $N$ linear measurements

A “good” sensing matrix is orthogonal

$\Phi^* \Phi = I$
Compressed Sensing

(Candes, Romber, Tao 2006; Donoho 2006)

- $x \in \mathbb{R}^N$ is a K-sparse signal ($K \ll N$)

- Make $M$ ($K < M \ll N$) incoherent linear projections

A “good” compressed sensing matrix is incoherent, i.e., approximately orthogonal

Incoherency can preserve information
CS recovery

- Given $y = \Phi x$
  - find $x$

- But there’s hope, $x$ is sparse!

\[ y = \Phi x \]
Given $y = \Phi x$

find $x$

But there’s hope, $x$ is sparse!
CS recovery

• Given $y = \Phi x$
  find $x$

• But there’s hope, $x$ is sparse!

\[
\minimize ||x||_2 \\
\text{s.t. } y = \Phi x
\]

WRONG!
CS recovery

• Given \( y = \Phi x \)
  find \( x \)

\[ \text{minimize } \|x\|_0 \]
\[ \text{s.t. } y = \Phi x \]

HARD!
CS recovery

• Given $y = \Phi x$
  find $x$

• But there’s hope, $x$ is sparse!

$$\begin{align*}
\text{minimize } & \|x\|_1 \\
\text{s.t. } & y = \Phi x
\end{align*}$$

need $M \approx K \log(N) << N$

Solved by linear-programming
Geometric Interpretation

- Domain of sparse signals
- Minimum $\|x\|_1$
- Minimum $\|x\|_2$

Mathematical expression:

\[
\begin{bmatrix}
0 & 0 & 1 \\
3 & 0 & 0 \\
0 & 1 & 0
\end{bmatrix}
\begin{bmatrix}
a_1 \\
a_2 \\
a_3
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
x_3
\end{bmatrix}
= [y_1]
\]