COMPRESSIVE SENSING

COMPRESSION

Why Compression?

Sensed data often includes redundant data: “data redundancy”.

Compression attempts to eliminate this redundancy.

In sensor networks, compression of data can potentially prolong network life time and maximize effective bandwidth utilization.

What is Redundancy?

- If one representation of the information content, M, takes X bytes and another takes Y bytes(Y< X), then X is a redundant representation relative to Y wrt M.
- Other forms of Redundancy

  If the representation capture content that is not useful, then removing such content will not affect the quality of the content.

  - For example, capturing audio frequencies outside the human hearing range can be avoided without any harm to the audio’s quality.

Is there a representation with an optimal size Z that cannot be improved upon?

This question is tackled by information theory.

REDUNDANCY IN THE CONTEXT OF WSN

Spatial redundancy:

1. a sensor reporting essentially the same measurements that has little spatial variability.

2. two sensors or more reporting measurements in areas where their measurement coverages overlap.

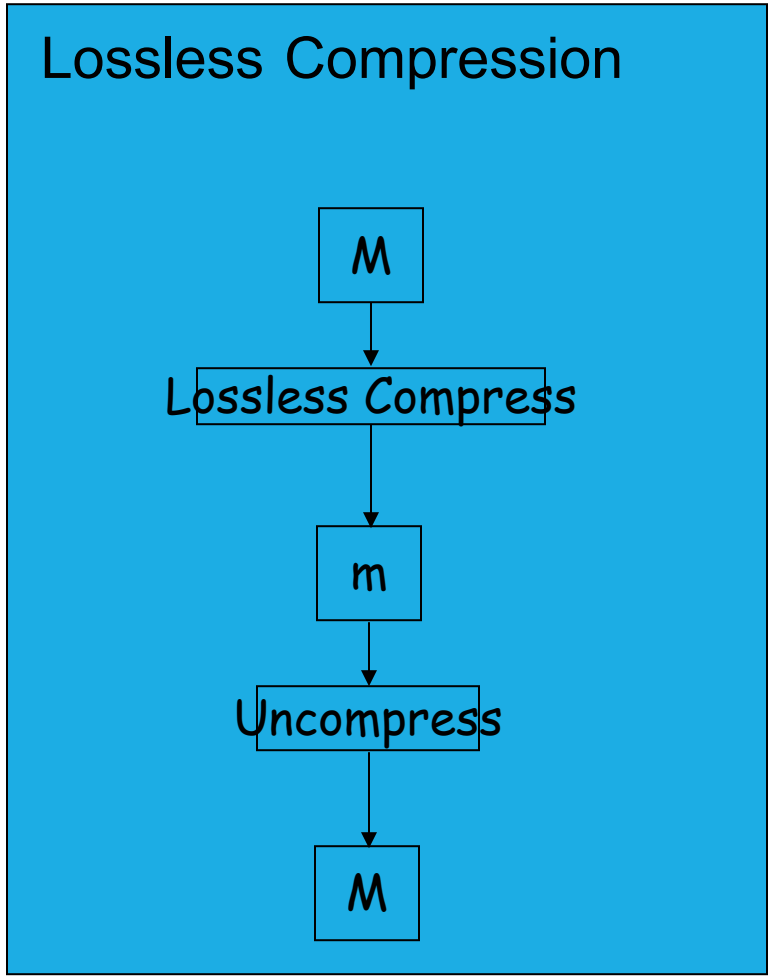
Coding redundancy,

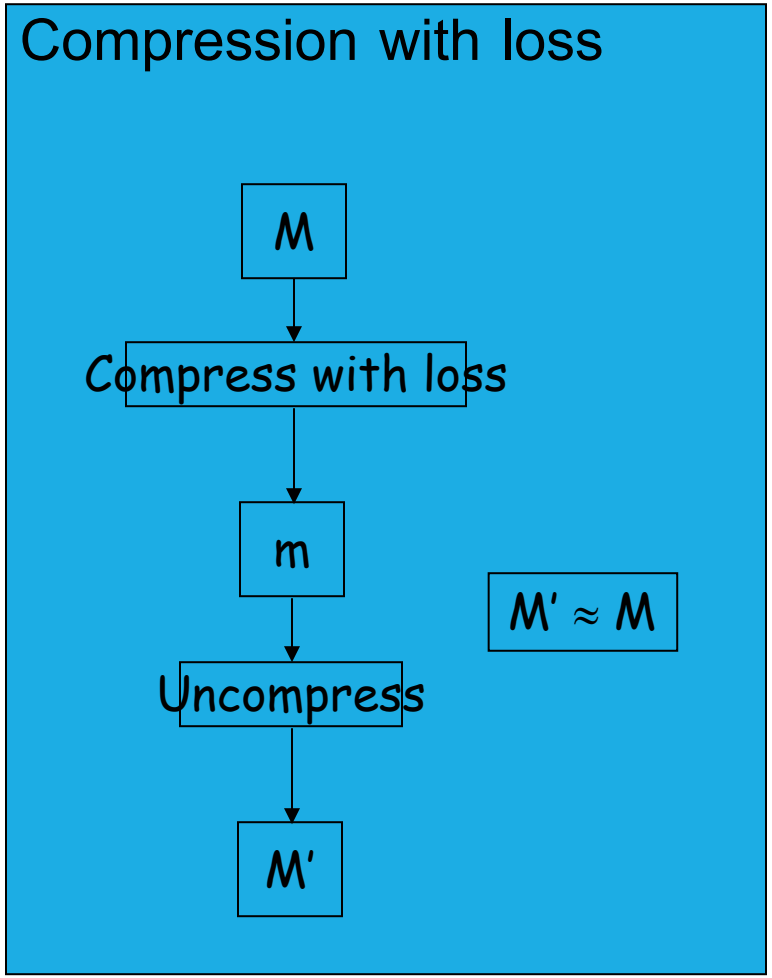
1. Sensor measurements are encoded in less than optimal coding. That is the same information contained in the measurements can be represented with less bits.

Temporal Redundancy:

1. When the measurements reported by the sensors remain the same or change in an insignificant manner.

LOSSLESS COMPRESSION VS LOSSY COMPRESSION





INFORMATION THEORY

According to Shannon, the **entropy**@ of an information source S is defined as:

$$H(S) = \sum_i (p_i \log_2 (1/p_i))$$

- $\log_2 (1/p_i)$ indicates the amount of information contained in *symbol* S_i, i.e., the number of bits needed to code *symbol* S_i.
- For example, in an image with uniform distribution of gray-level intensity, i.e. $p_i = 1/256$, with the number of bits needed to code each gray level being 8 bits. The entropy of the image is 8.
- Q: What is the entropy of a source with M symbols where each symbol is equally likely?
  - *Entropy,* $H(S) = \log_2 M$
- Q: How about an image in which half of the pixels are white and half are black?
  - *Entropy,* $H(S) = 1$

INFORMATION THEORY

Discussion:

- Entropy is a measure of how much information is encoded in a message. Higher the entropy, higher the information content.
  - We could also say entropy is a measure of uncertainty in a message. Information and Uncertainty are equivalent concepts.
- The units (in coding theory) of entropy are *bits per symbol*.
  - It is determined by the base of the logarithm:

    2: binary (bit);

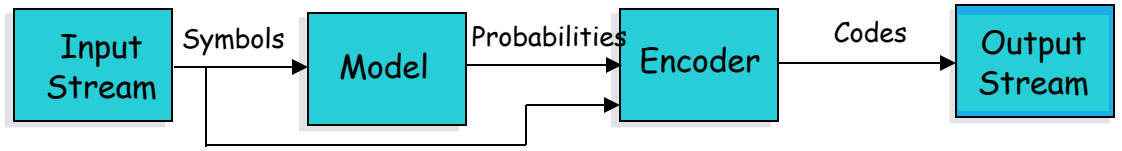
    10: decimal (digit).
- Entropy gives the actual number of bits of information contained in a message source.
  - **Example:** If the probability of the character 'e' appearing in this slide is $1/16$, then the information content of this character is 4 bits. So, the character string "eeeee" has a total content of 20 bits (contrast this to using an 8-bit ASCII coding that could result in 40 bits to represent "eeeee").

DATA COMPRESSION = MODELING + CODING

Data Compression consists of taking a stream of symbols and transforming them into codes.

- The model is a collection of data and rules used to process input symbols and determine their probabilities.
- A coder uses a model (probabilities) to spit out codes when its given input symbols

Let's take Huffman coding to demonstrate the distinction:



- The output of the Huffman encoder is determined by the Model (probabilities). *Higher the probability shorter the code.*
  - Model A could determine raw probabilities of each symbol occurring anywhere in the input stream. (p_i = *# of occurrences of* S_i */ Total number of Symbols*)
  - Model B could determine prob. based on the last 10 symbols in the i/p stream. (*continuously re-computes the probabilities*)

THE SHANNON-FANO ENCODING ALGORITHM

1. Calculate the frequencies of the list of symbols (organize as a list).
2. Sort the list in (decreasing) order of frequencies.
3. Divide list into two halfs, with the total freq. *Counts of each half being as close as possible to the other*.
4. The upper half is assigned a code of 0 and lower a code of 1.
5. Recursively apply steps 3 and 4 to each of the halves, until each symbol has become a corresponding code leaf on a tree.

Example

Symbol	A	B	C	D	E
Count	15	7	6	6	5
	0	0	1	1	1
	0	1	0	1	1
				0	1

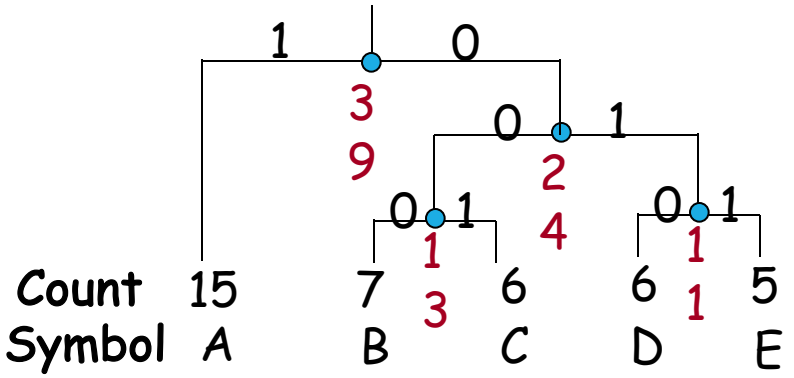
Symbol	Count	Info. $-\log_2(p_i)$	Code	Subtotal # of Bits
A	15	1.38	00	30
B	7	2.48	01	14
C	6	2.70	10	12
D	6	2.70	110	18
E	5	2.96	111	15
		85.25		89

It takes a total of 89 bits to encode 85.25 bits of information (Pretty good huh!)

THE HUFFMAN ALGORITHM

1. Initialization: Put all nodes in an OPEN list *L*, keep it sorted at all times (e.g., ABCDE).
2. Repeat the following steps until the list *L* has only one node left:
   1. From *L* pick two nodes having the lowest frequencies, create a parent node of them.
   2. Assign the sum of the children's frequencies to the parent node and insert it into L.
   3. Assign code 0, 1 to the two branches of the tree, and delete the children from L.

Example



Symbol	Count	Info. $-\log_2(p_i)$	Code	Subtotal # of Bits
A	15	1.38	1	15
B	7	2.48	000	21
C	6	2.70	001	18
D	6	2.70	010	18
E	5	2.96	011	15
		85.25		87

HUFFMAN ALG.: DISCUSSION

Decoding for the above two algorithms is trivial as long as the coding table (the statistics) is sent before the data. There is an overhead for sending this, negligible if the data file is big.

Unique Prefix Property: no code is a prefix to any other code (all symbols are at the leaf nodes)

--> great for decoder, unambiguous; **unique Decipherability?**

If prior statistics are available and accurate, then Huffman coding is very good.

Number of bits (per symbol) needed for Huffman Coding is:

87 / 39 = 2.23

Number of bits (per symbol)needed for Shannon-Fano Coding is:

89 / 39 = 2.28

Compressive Sensing

Compressive sampling is a recent development in digital signal processing that offers the potential of high resolution capture of physical signals from relatively few measurements, typically well below the number expected from the requirements of the Shannon/Nyquist sampling theorem.

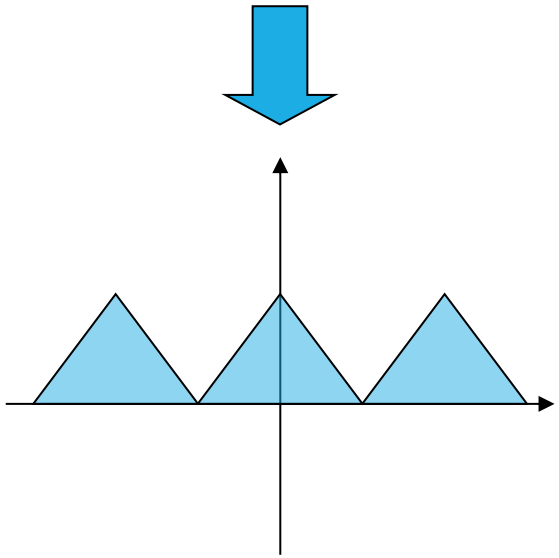
This technique combines two key ideas:

1. *sparse representation through an informed choice of linear basis for the class of signals under study; and*
2. *incoherent (eg. pseudorandom) measurements of the signal to extract the maximum amount of information from the signal using a minimum amount of measurements.*

Mathematical techniques required to implement compressive sampling include the development of novel types of linear bases (eg. wavelet, curvelet, etc.), L1 optimization to recover sparse representations, and design of optimal dual measurements.

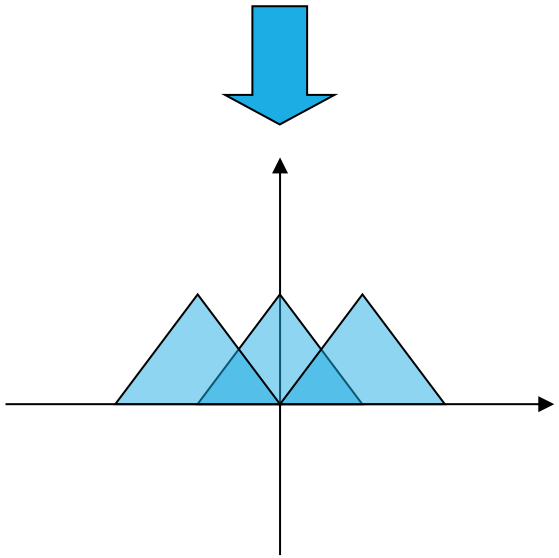
SAMPLING THEOREM BOUNDS THE NUMBER OF SAMPLES REQUIRED FOR FULL SIGNAL RECOVERY

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-jk\frac{2\pi}{N}n}$$



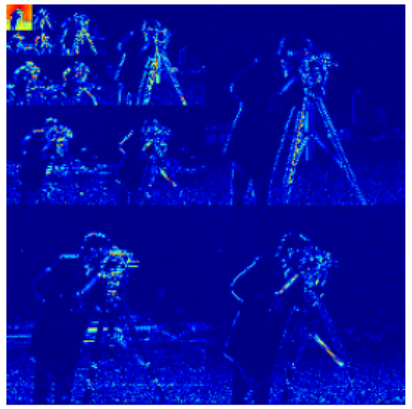
V.S.

$$X[k] = \sum_{n=0}^{N/2-1} x[2n] \cdot e^{-jk\frac{2\pi}{N}2n}$$



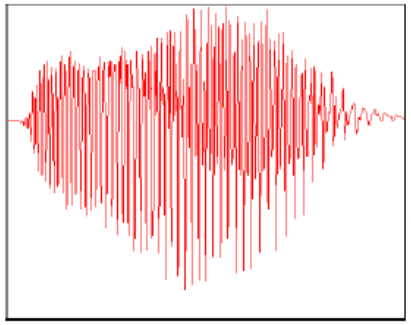
COMPRESSIBLE SIGNAL: "INHOMOGENEOUS" INFORMATION DISTRIBUTION

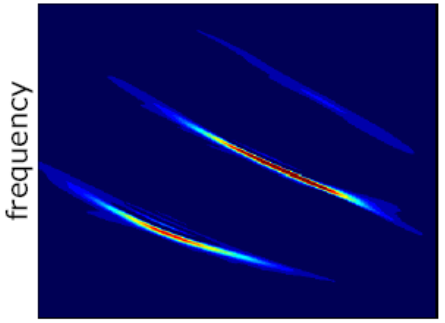
N
pixels



$K \ll N$
large
wavelet
coefficients

N
wideband
signal
samples





$K \ll N$
large
Gabor
coefficients

Tutorial on Compressive Sensing, R. Baraniuk et al. (Feb 2008)

WHAT IS COMPRESSED SENSING?

Underlying Assumption: Most signals are compressible in some representation (i.e., most coefficients are small relative to some basis)

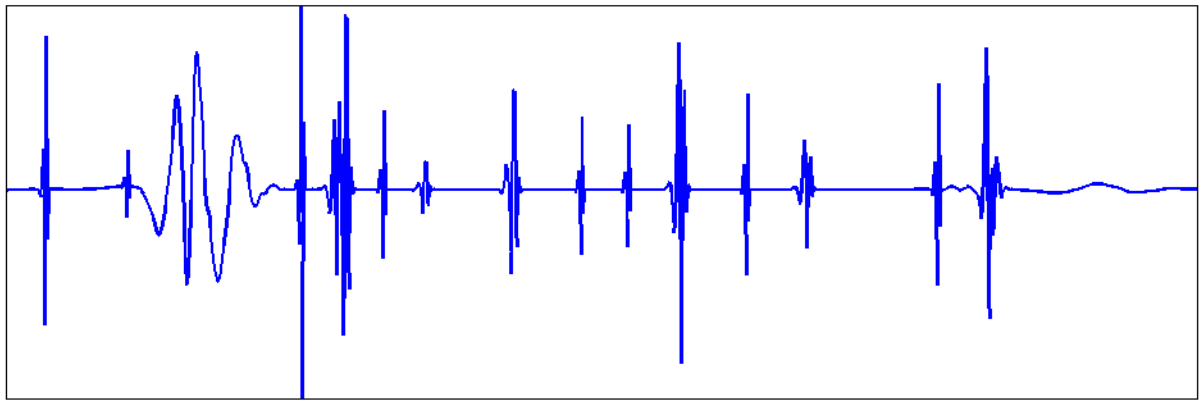
Compressed Sensing:

- "Measure" the signal via a random projection to yield a compact representation
- Reconstruct the signal from its compact representation

NYQUIST VS. COMPRESSED SENSING

Nyquist rate samples of wideband signal (sum of 20 wavelets)

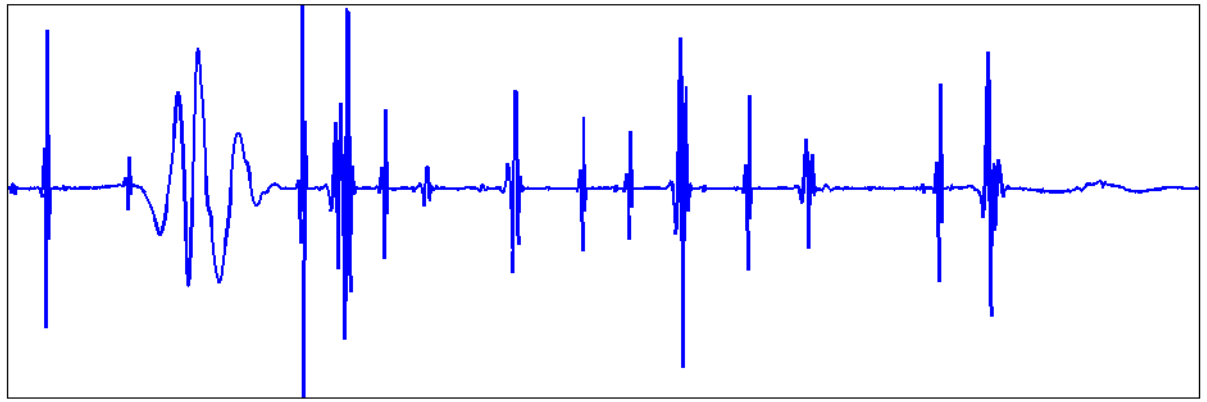
$N = 1024$ samples/second



Reconstruction from **compressed sensing**

$M = 150$ **random** measurements/second

MSE < 2% of signal energy



NYQUIST VS. COMPRESSED SENSING

Nyquist rate samples of image

$N = 65536$ pixels

Reconstruction from **compressed sensing**

$M = 20000$ projections

MSE < 3% of signal energy

COMPRESSED SENSING

- Under certain conditions, it is possible to fully recover a signal from sampling points **much fewer than that defined by Shannon's sampling theorem**

- Given **x** of length N, only M measurements (M<N) is required to fully recover **x** when **x** is K-sparse (K<M<N)

- However, three conditions named CS1-3 are to be satisfied for the above statement to be true

COMPRESSIBILITY CONDITIONS

Sparsity:

- The desired signal has a sparse representation in a known transform domain

Incoherence

- Undersampled sampling space must generate noise-like aliasing in that transform domain

Non-linear Reconstruction

- Requires a non-linear reconstruction to exploit sparsity while maintaining consistency with acquired data

SPARSITY

Number of significant(strictly speaking, nonzero)components is relatively small compared to signal length

- Ex: [**1** 0 **10** 0 0 0 0 0.....0 0]

Sparsity Representation:

- L_{p-Norm}:

$$\|x\|_p = \left(\sum_{i=1}^{N} |x_i|^p \right)^{\frac{1}{p}}$$

- L_0 norm counts the number of non−zero components of **x**
- Ex: if **x**=[1, 100000, 2, 0], then L_{0-Norm}=3

INCOHERENCE

Sampling must generate noise-like aliasing in image domain (more strictly, transform domain)

Very loosely speaking, patterns of sampling must demonstrate enough randomness

NON-LINEAR RECONSTRUCTION

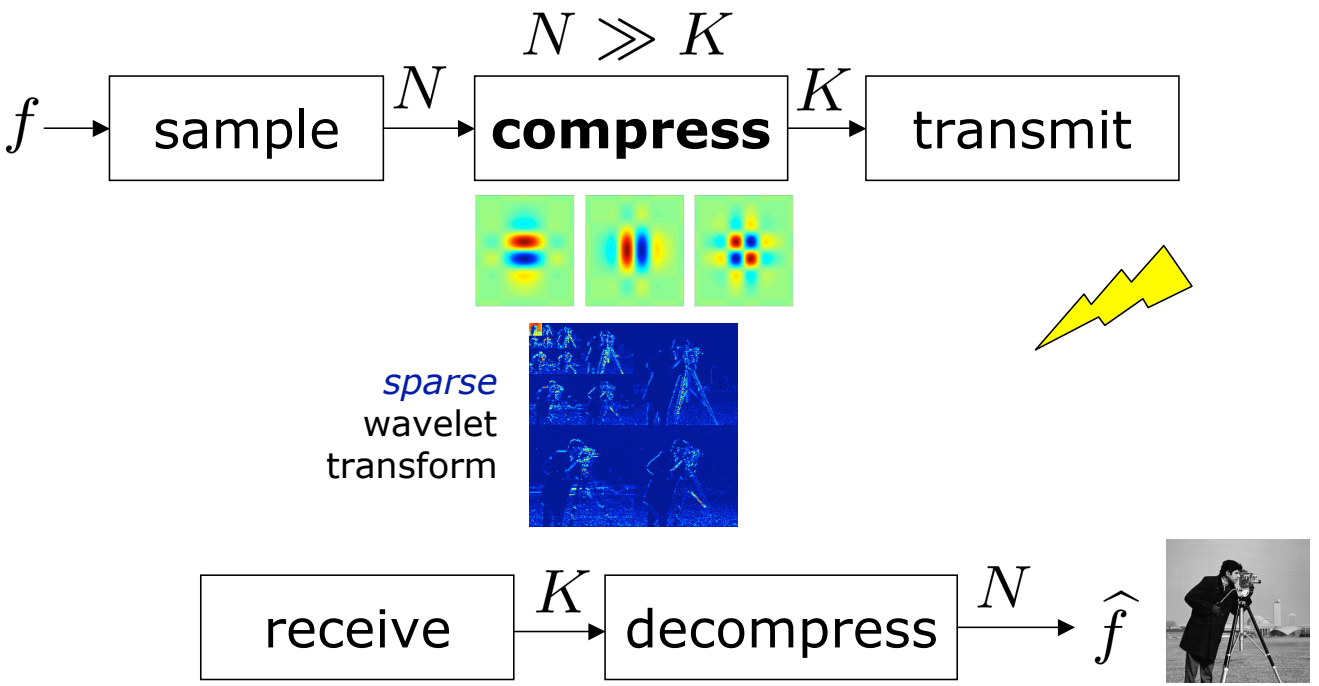
Lacks the linearity of FFT and iFFT

Does not have analytical solution as in STFT, Gabor Transform, WDF(Wigner distribution function),….etc

Involves optimizations (often iterative) satisfying certain boundary conditions

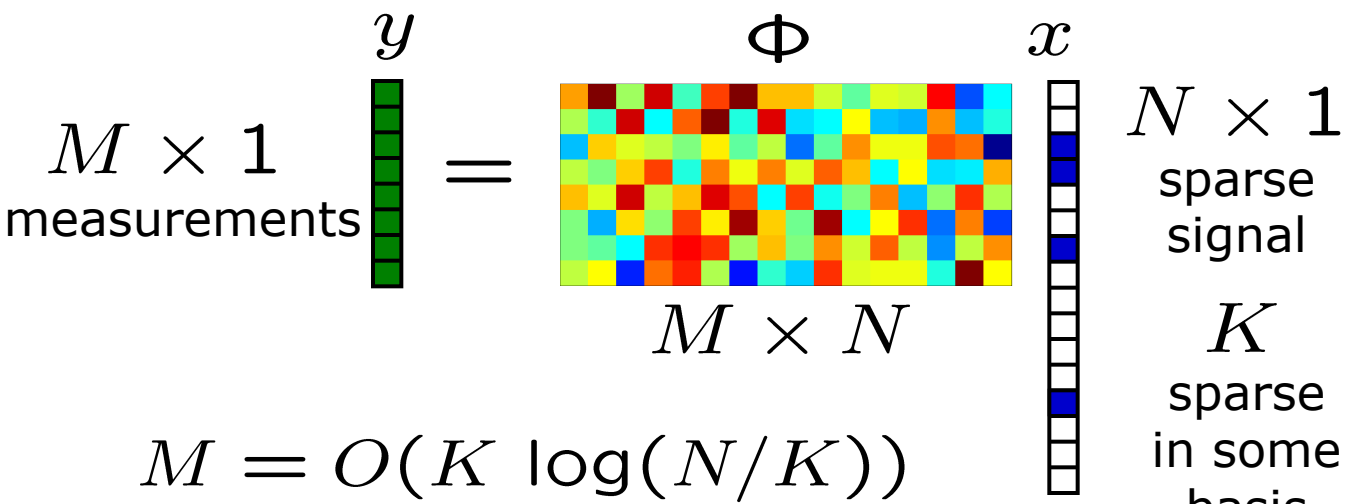
Sensing by *Sampling*

- Long-established paradigm for digital data acquisition
  - ***sample*** data (A-to-D converter, digital camera, …)
  - ***compress*** data (signal-dependent, nonlinear)
  - ***wasteful:*** can we obtained compressed version directly?



COMPRESSED SENSING

$$y = \Phi x$$



$$M = O(K \log(N/K))$$

$$K < M \ll N$$

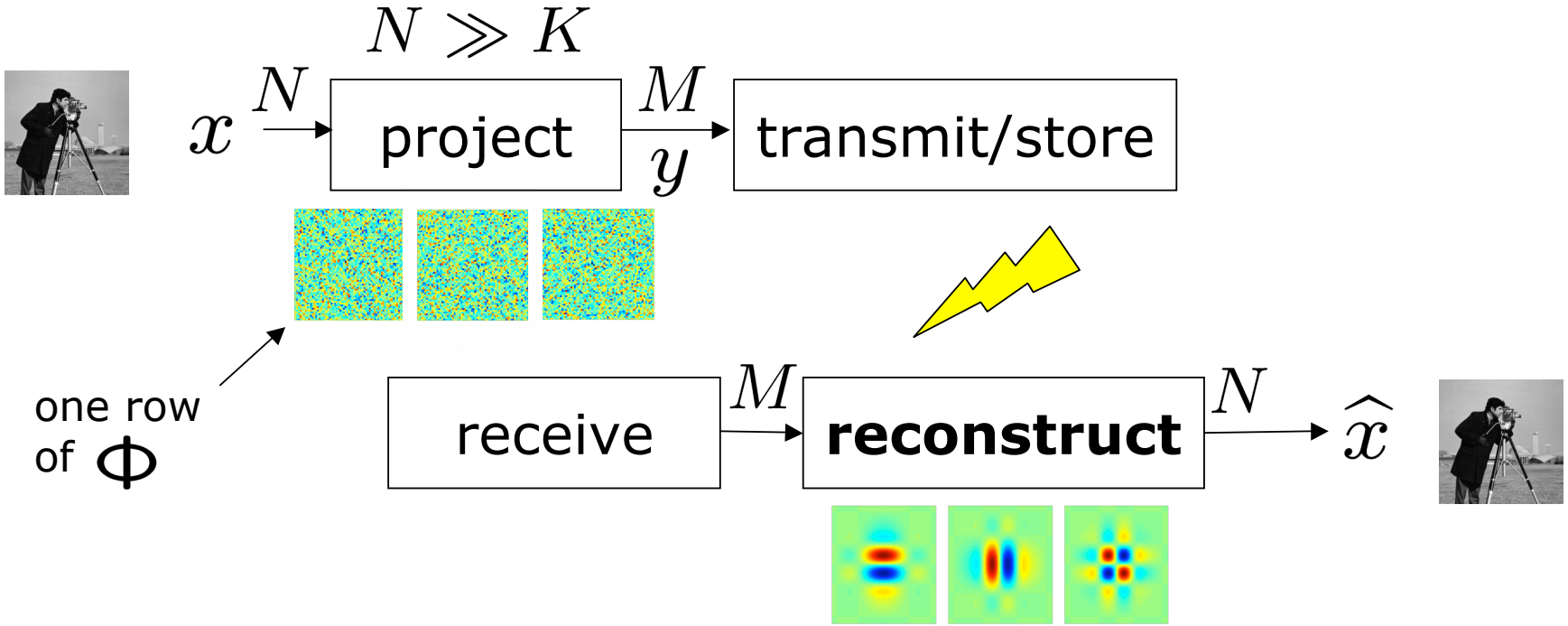
Forward Problem: Random projection is the key idea

Inverse Problem: Reconstruct x from y ; this is an ill–posed problem

[Candes-Romberg-Tao, Donoho, 2004]

Compressive Sensing

- Measure linear projections onto *incoherent* basis where data is *not sparse/compressible*



- Reconstruct via *nonlinear processing* (optimization) (using sparsity-inducing basis and Φ)

CS MATHEMATICALLY

Minimize $\|x\|_0$ *such that* $\|y'-y\|_2 < \varepsilon$ *, where*

x stands for reconstructed signal

y' stands for the estimated measurement

y stands for the initial measurement

ε serves as the boundary condition (usually noise level)

In other words, among all possible solutions of **x**, find one with the smallest L_{0-norm}(i.e. sparsest) whose estimated measurement **y'** remains consistent with the initial measurement **y** with deviation less than **ε**

REVISIT CS MATHEMATICALLY

Minimize $\|\Psi x\|_0$ *such that* $\|y'-y\|_2 < \varepsilon$ *, where*

x stands for reconstructed signal

Ψ stands for sparsifying transform

y' stands for the estimated measurement

y stands for the initial measurement

ε serves as the boundary condition (usually noise level)

Among all possible **sparsified** solutions, find one with the smallest L_{0-norm}(i.e. sparsest) whose estimated measurement **y'** remains consistent with initial measurement **y** with deviation less than **ε**

Most use L_{1-norm}, i.e. minimize $\|\Psi x\|_1$ instead

REPRESENTATION OF SPARSITY IS ESSENTIAL TO REQUIRED SAMPLE NUMBER

L_{0-norm} is ideal, yet intractable

- Needs only **M**=**K**+1 samples for **K**–sparse signals
-

$$\|x\|_p = \left(\sum_{i=1}^{N} |x_i|^p \right)^{\frac{1}{p}}$$

is an NP problem when p=0

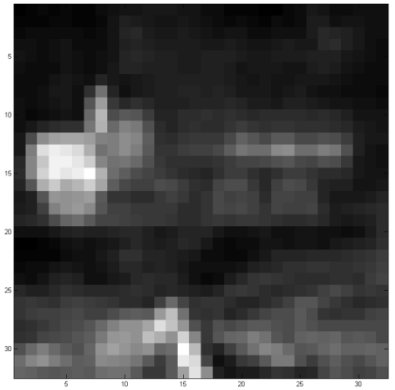
L_{2-norm}(i.e.) is well–known, yet inaccurate

- p=2 represents Least Mean Square

L_{1-norm} requires more samples than L_0, yet is most feasible in its tractability and accuracy

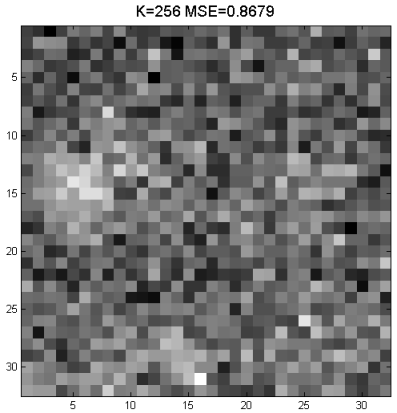
- Needs approximately **K log(N/K)** samples, yet no longer NP
- L_{1-norm} minimization is equivalent to a classical convex optimization problem with many well–established approaches

EXAMPLE USING IDEAL WEIGHTS



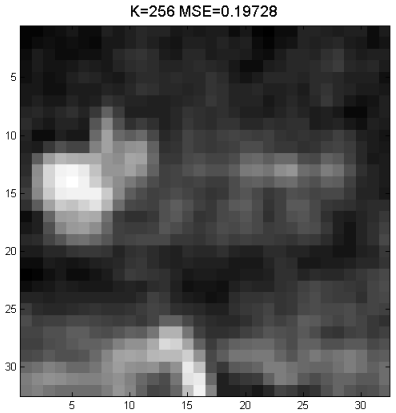
Original 32 x 32 image

Conventional L2 norm

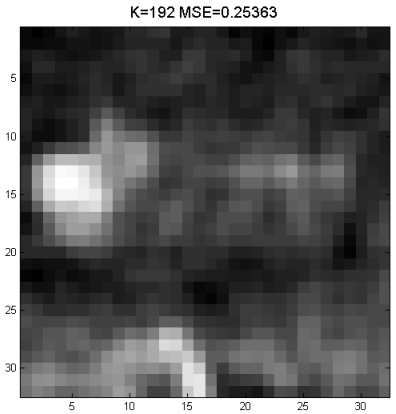


K=256
mse=0.86

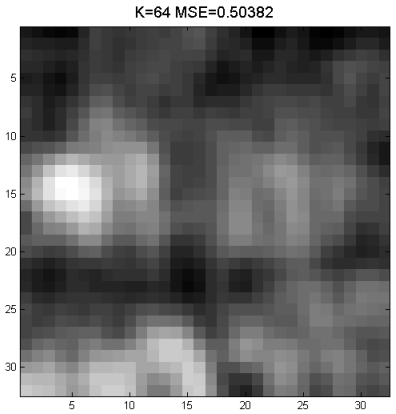
WEIGTED L2 norm



K=256
mse=0.19



K=192
mse=0.25



K=64
mse=0.5

Original image is 32 x 32 $(1024$ elements)

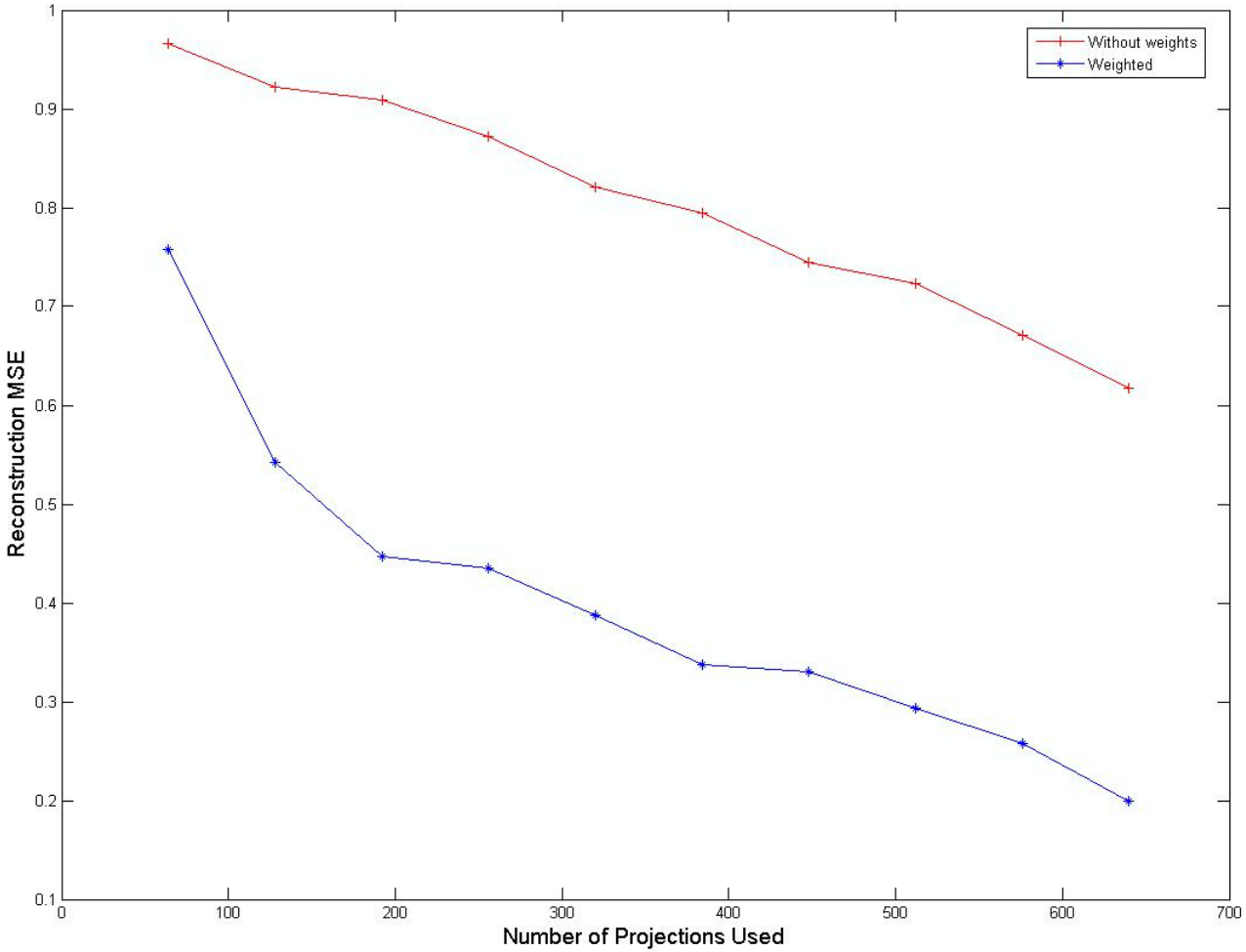
DCT is used as a basis set

- Any other basis set that allows compact representation can be used
- ideal coefficients are used as a “place-holder” for weights
  - In practice, these will be estimated representative images of the class of objects of interest, or statistically modeled.

Weighted L_{2-norm} produces recognizable results using $1/4^{th}$ the data $(256$ measurements)

- Conventional L_{2-norm} does not perform well

WEIGHTED VS. CONVENTIONAL APPROACH USING DCT BASIS



Comparison of conventional and weighted minimum L_{2-norm} reconstruction using the DCT basis functions. Weighting the reconstruction process makes a significant difference in the reconstruction error

MATLAB EXAMPLE

If we sample the signal (represented by the vector x_0) using a matrix A, we get the smaller vector b:

$Ax_0 = b$

b must be smaller than x, otherwise no compression of the signal has occurred.

This means that the matrix A must have fewer rows than columns, and the system is "underdetermined", having more unknowns than equations.

This usually means that the vector b is not unique, and if we try to solve the system for x_0 using A and b, we will get many different solutions.

However, in compressed sensing, we can guarantee that that we will be able to solve the system to recover the original signal.

This guarantee requires two things:

- first, the signal must be sparse,
- and second, we must use a special kind of measurement matrix. Most of these special matrices satisfy the "Restricted Isometry Property".

Creating an Example Signal

Let's create a sparse signal that we can measure and reconstruct.

We will create a "spike train" signal, which only has -1 and 1 magnitudes. Our signal will have k nonzero values out of n values.

```
n = 512; k = 20;
```

Create a random permutation of the integers 1 to n; the first k of these will be the indices of the signal's nonzero values.

```
p = randperm(n);
```

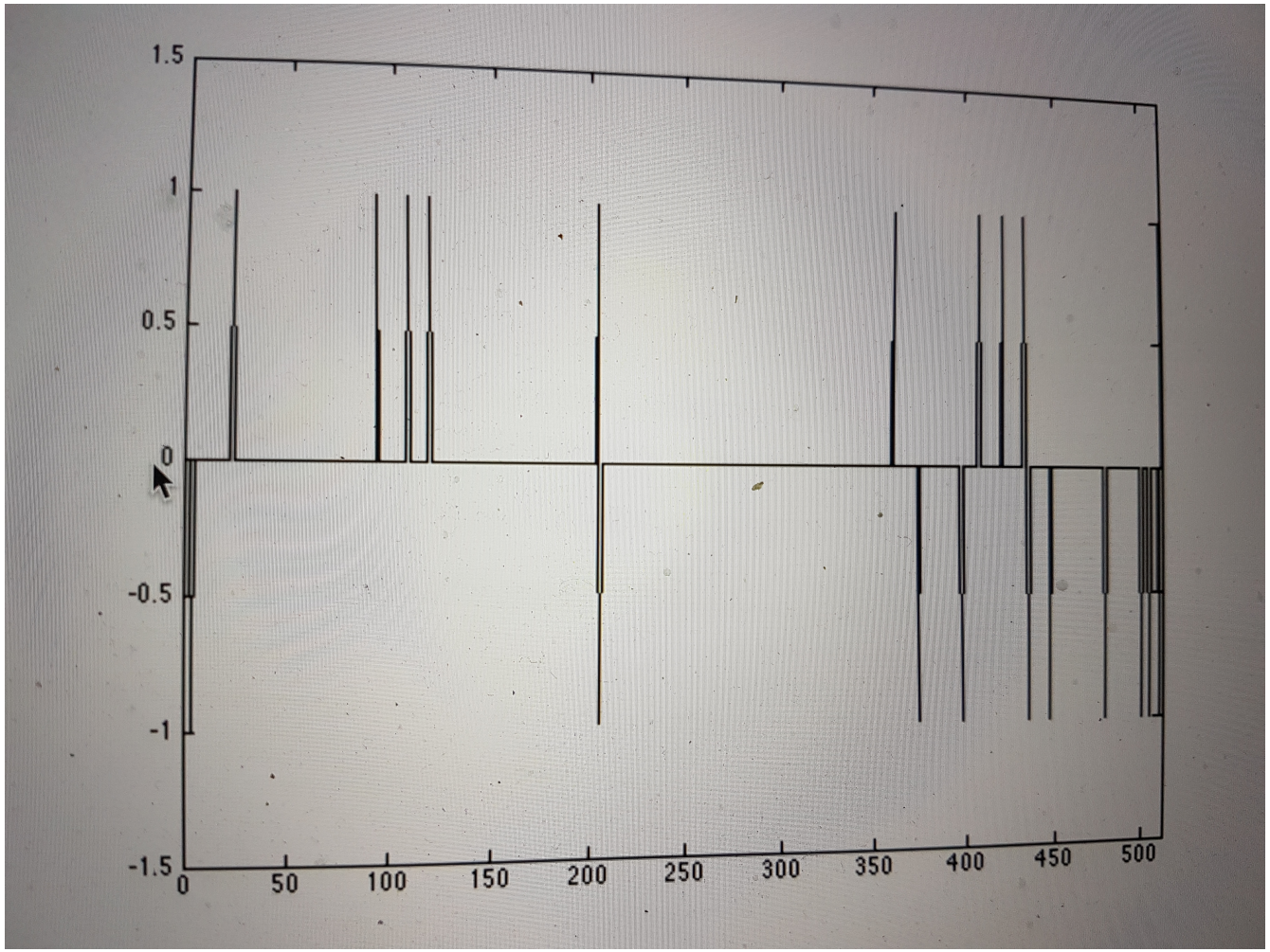
Initialize the signal x_0 as a column vector of n zeros.

Set the designated nonzero values to be randomly 1 or -1:

x0 = zeros(n,1); x0(p(1:k)) = sign(randn(k,1)); Plot the resulting signal:

figure(1); plot(1:n, x0); axis([0 512 -1.5 1.5]);

THE SIGNAL



```
m = 120;
A = opGaussian(m,n,2)
A = Spot operator: Gaussian(120,512)
rows: 120; complex: no; cols: 512
type: Gaussian
```

TAKING THE MEASUREMENTS

To take our measurements, we simply have to apply A to x_0.

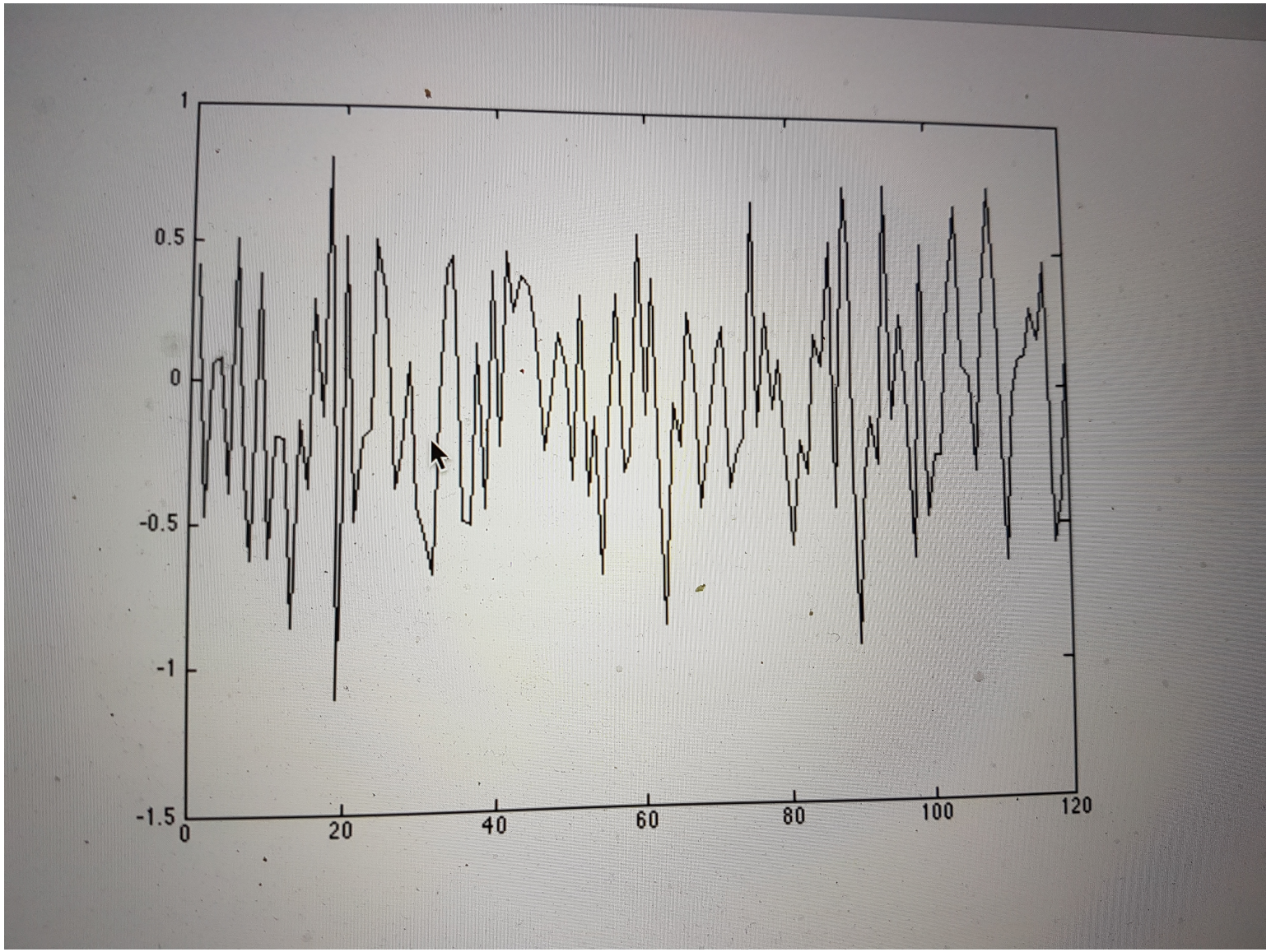
Our new vector b has m entries. We will also add some random noise:

b = A*x_0 + 0.005 * randn(m,1);

This is what our compressed data looks like:

figure(); plot(1:m, b)

SAMPLED MEASUREMENTS



We have stored our signal as the vector b; let's try to recover it using A.

We will use a solver called SPGL1, which solves the following basis pursuit denoising problem:

$$\text{Minimize } ||x||_{1}$$

$$\text{Subject to } ||Ax-b|| <= \sigma$$

The second half of this problem is our condition that x actually satisfies the equation Ax=b;
within some range represented by a
Minimizing the 1-norm of that meets this condition.
First we will set our "optimality tolerance" to 0.0001. This means that the solution that the solver finds is guaranteed to be within 0.01% of the optimal solution.

```
opts = spgSetParms('optTol', 1e-4, 'verbosity', 1);
```

```
opts = spgSetParms('optTol', 1e-4, 'verbosity', 1);
```

Next we simply pass A, b, $\sigma = 0.001$, and our parameters to the SPGL1 solver:

```
[x,r,g,info] = spg_bpdn(A,b,1e-3,opts);
```

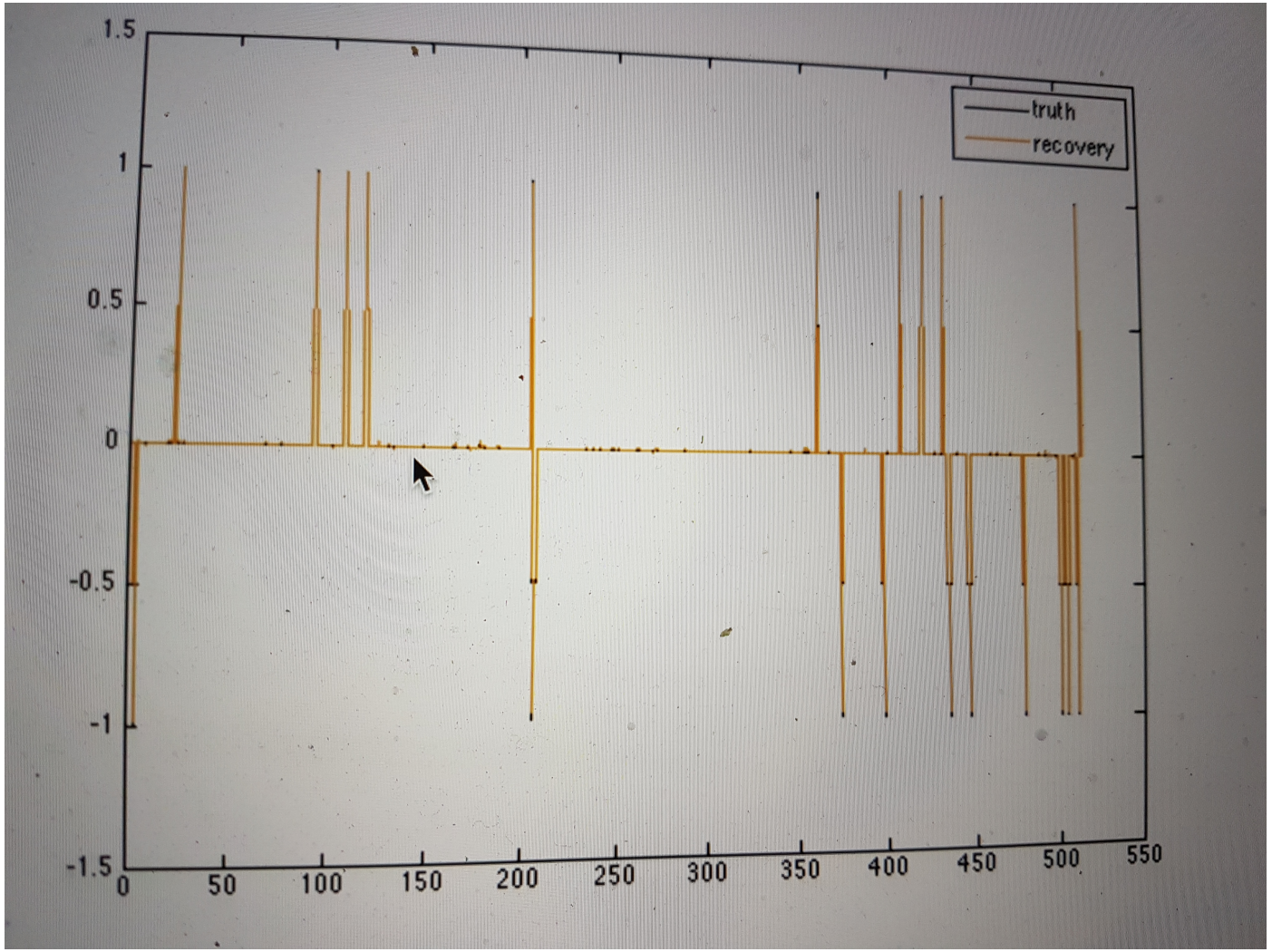
```
================================================================================
SPGL1  v. 1017  (Mon, 16 Jun 2008)
================================================================================
No. rows              :      120     No. columns           :      512
Initial tau           : 0.00e+00     Two-norm of b         : 4.51e+00
Optimality tol        : 1.00e-04     Target objective      : 1.00e-03
Basis pursuit tol     : 1.00e-06     Maximum iterations    :     1200

 Iter     Objective   Relative Gap  Rel Error      gNorm    stepG   nnzX   nnzG        tau
    0  4.5099723e+00  0.0000000e+00  1.00e+00  1.538e+00      0.0      0      0  1.3220579e+01
   17  1.3973498e+00  8.5342625e-02  9.99e-01  2.915e-01     -0.3     32      1  1.9914305e+01
  102  4.6195711e-02  8.4784772e-03  4.52e-02  7.806e-03     -0.3     58     58  2.0181762e+01
  165  8.4319194e-03  3.7388083e-04  7.43e-03  1.076e-03      0.0     98     98  2.0240011e+01
  363  1.2680178e-03  2.3306436e-04  2.68e-04  1.634e-04     -0.6    108    108  2.0242091e+01
  364  1.0396718e-03  1.5000575e-03  3.97e-05  1.774e-04      0.0    108    108

EXIT -- Found a root

Products with A     :      601     Total time    (secs) :      2.1
Products with A'    :      366     Project time  (secs) :      0.1
Newton iterations   :        5     Mat-vec time  (secs) :      1.2
Line search its     :      236     Subspace iterations  :        0
```

the solver stopped once the relative error became less than 0.0001.

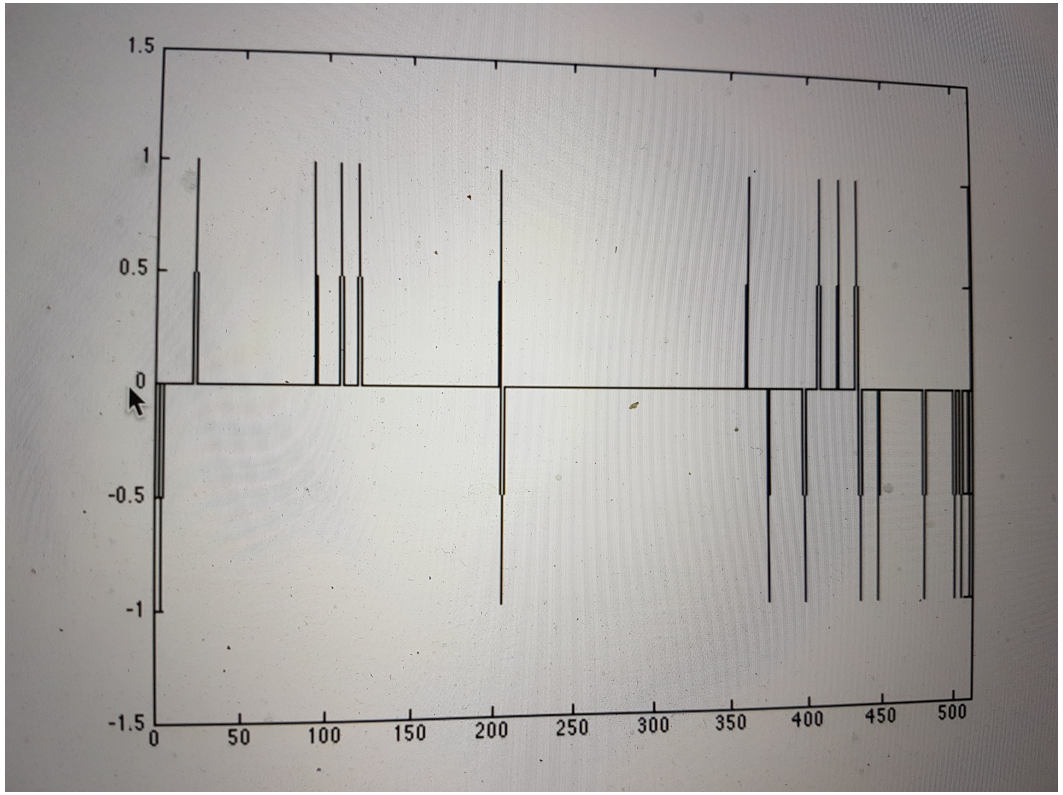
Plot the reconstruction, x, with the original signal, x0:

figure(2); plot(1:n, x0, 1:n, x); axis([0 550 −1.5 1.5]); legend('truth', 'recovery');

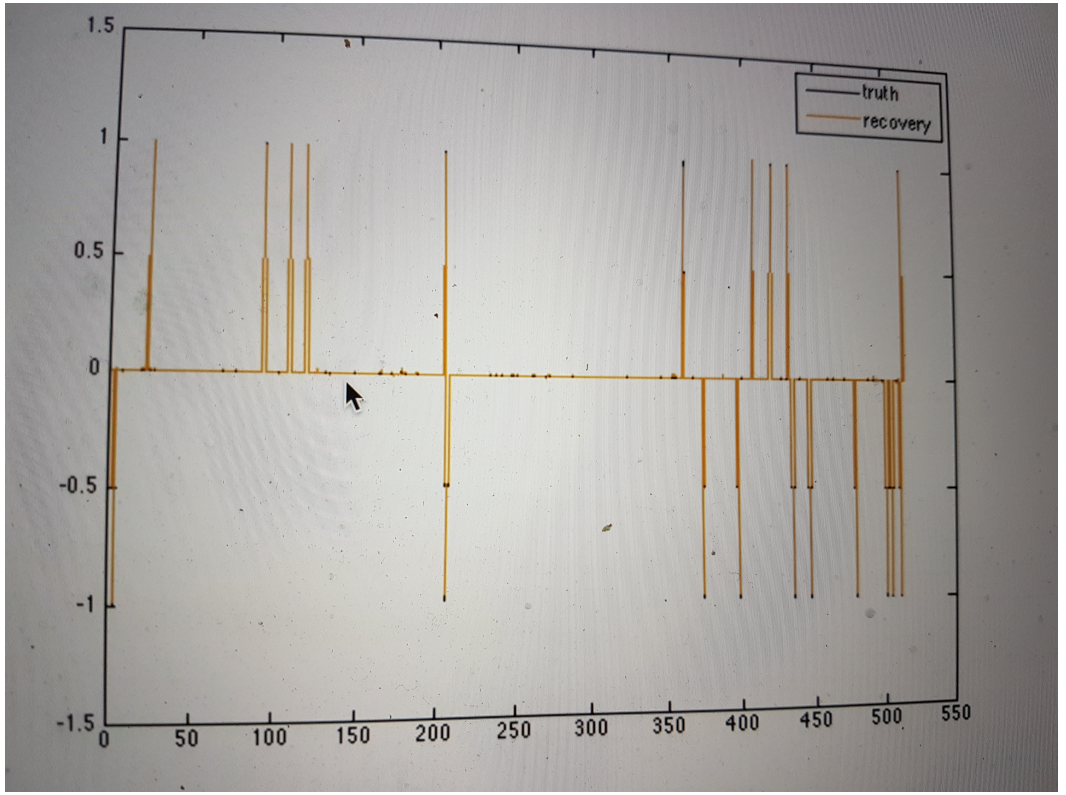
THE SIGNAL AND RECONSTRUCTION OVERLAP ALMOST COMPLETELY, SO OUR RECONSTRUCTION IS ACCURATE.



Original Signal



Recovered Signal



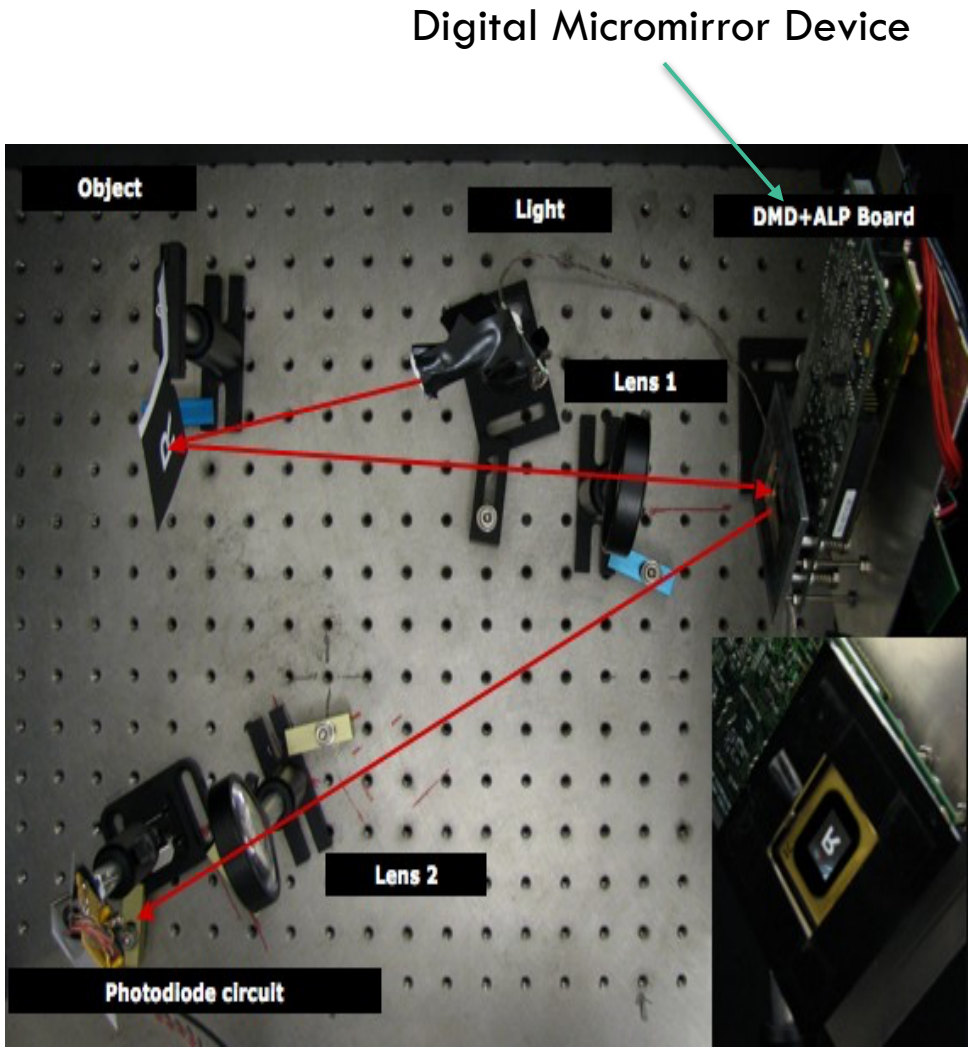
COMPRESSIVE IMAGING

Single Pixel Camera

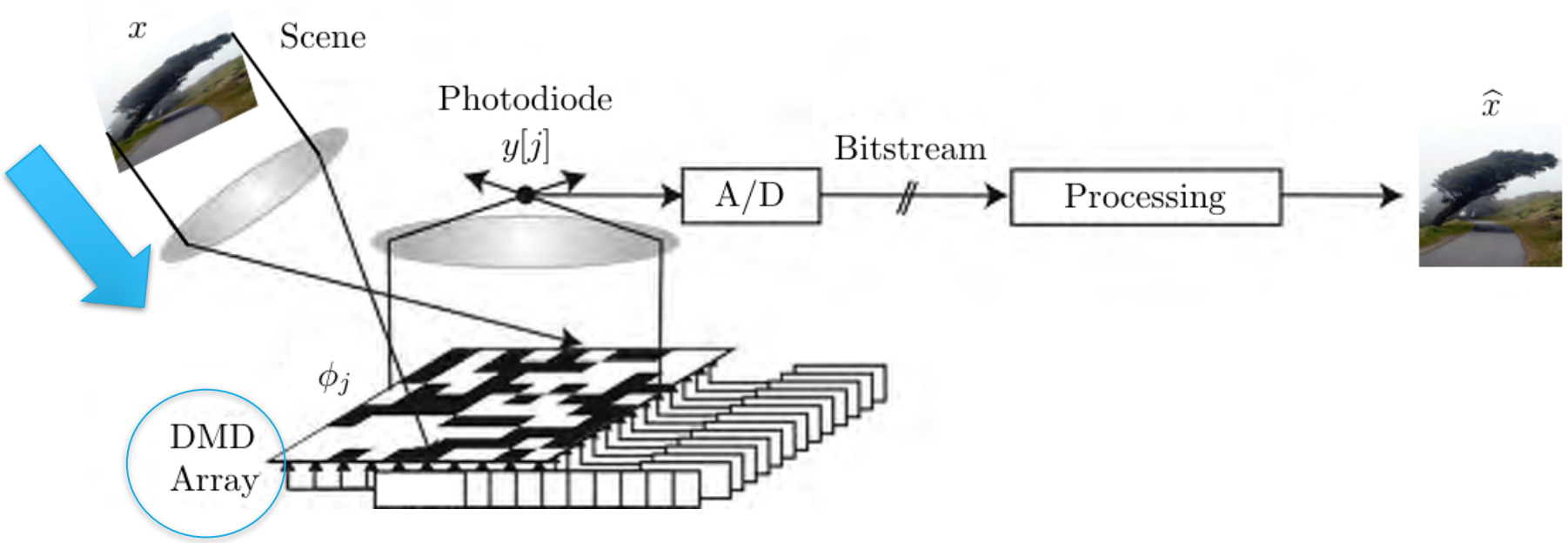
SINGLE PIXEL CAMERA

What is a single pixel camera

- An optical computer
- sequentially measures the $y[i]$
- Directly acquires M random linear measurements without first collecting the N pixel values



SINGLE PIXEL CAMERA- ARCHITECTURE



SINGLE PIXEL CAMERA- DMD ARRAY

Digital Micro mirror Device

A type of a reflective spatial light modulator

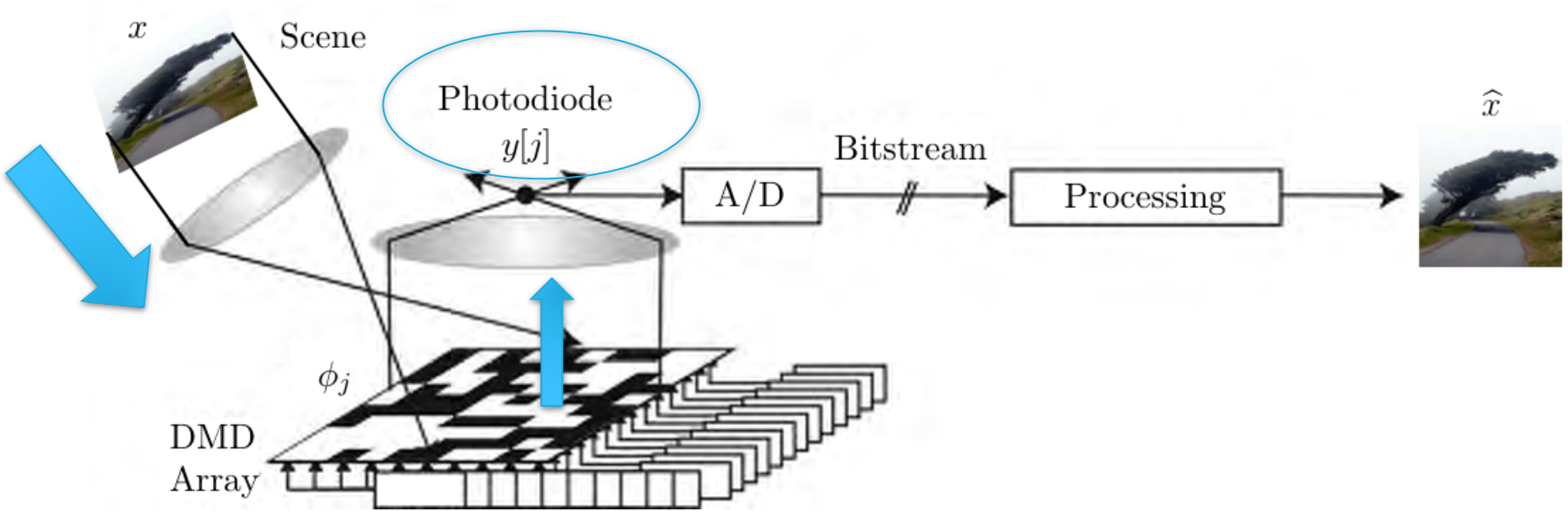
Selectively redirects parts of the light beam

Consisting of an array of N tiny mirrors

Each mirror can be positioned in one of two states$(+/-10$ degrees$)$

Orients the light towards or away from the second lens

SINGLE PIXEL CAMERA- ARCHITECTURE



SINGLE PIXEL CAMERA- PHOTODIODE

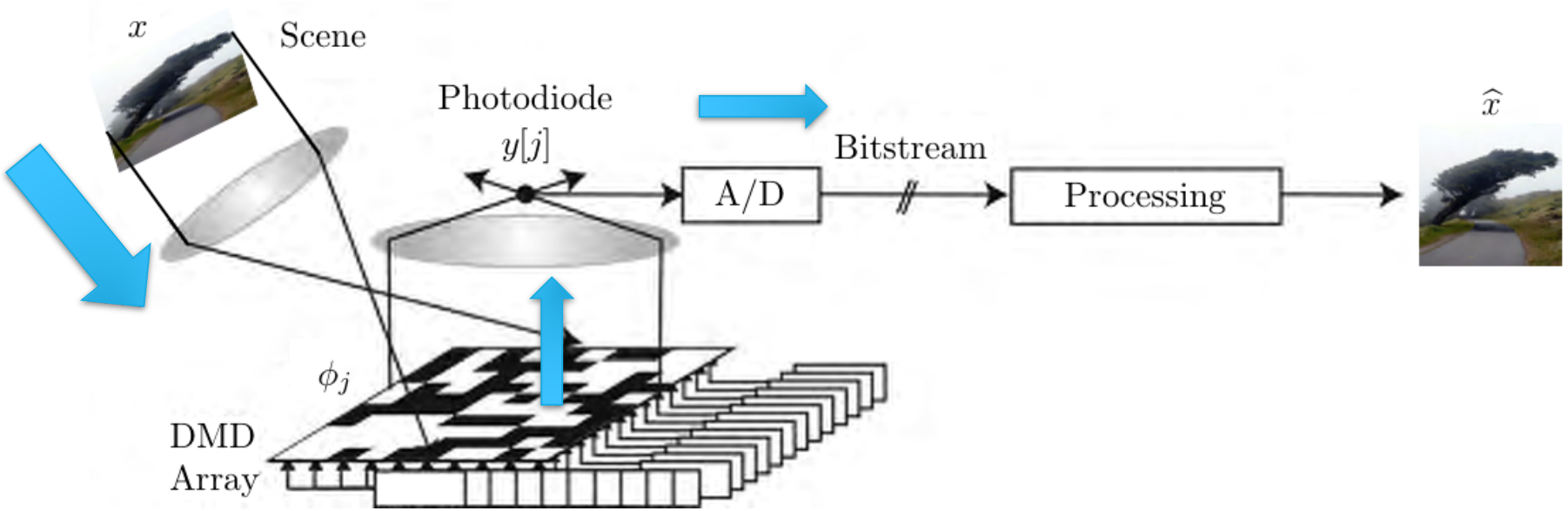
Find the focal point of the second lens

Place a photodiode at this point

Measure the output voltage of the photodiode

The voltage equals y_j, which is the inner product between ϕ_j and the desired image x.

SINGLE PIXEL CAMERA- ARCHITECTURE



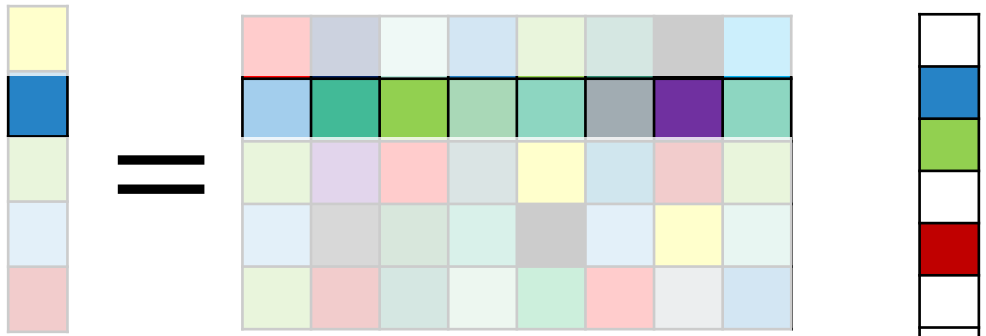
SINGLE PIXEL CAMERA- MEASUREMENTS

A random number generator (RNG) sets the mirror orientations in a pseudorandom $1/0$ pattern

Repeats the above process for M times

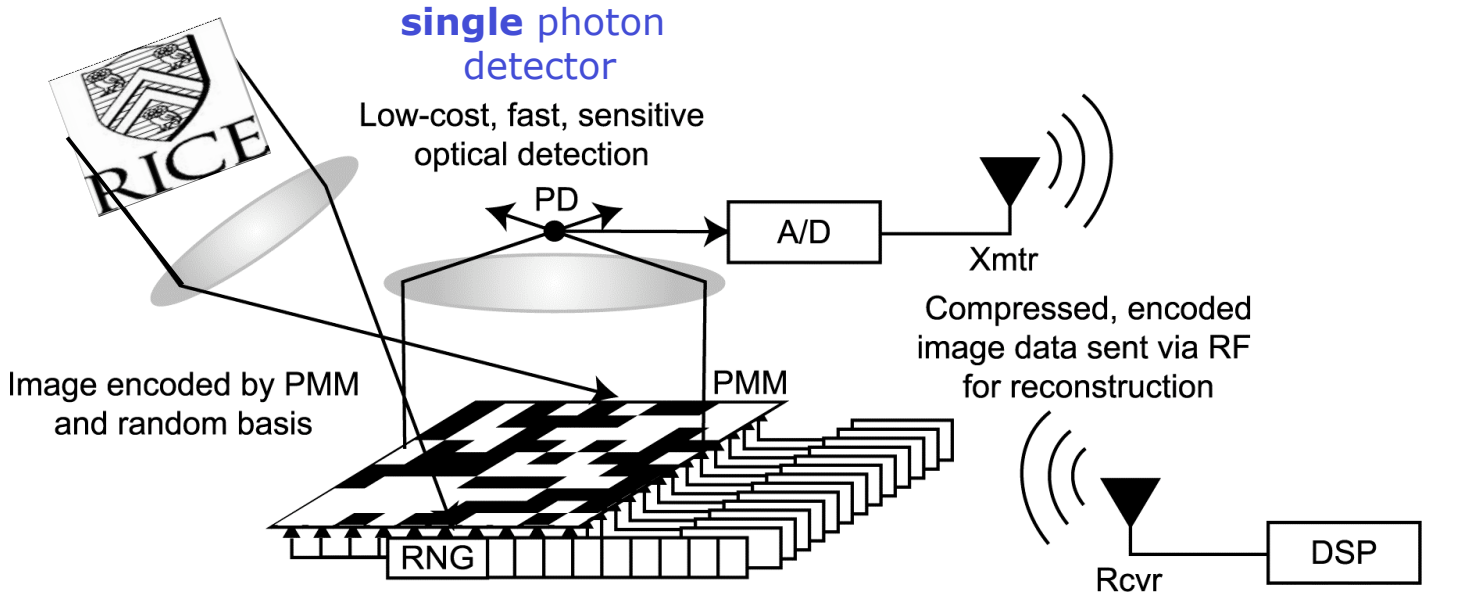
Obtains the measurement vector y and ϕ

Now we can construct the system in the



y_j Φ_j x

Rice CI Camera



Other CI implementations:
D. Brady (Duke)

SAMPLE IMAGE RECONSTRUCTIONS

$256*256$ conventional image of black and white 'R'

Image reconstructed from $M = 1300$

How can we improve the reconstruction further?