Compressive Sensing: A Potentially Revolutionary Technology

H. T. Kung

Harvard School

of Engineering and Applied Sciences

Version June 3, 2011

Historical Perspective:
The Avalanche Trend of Sensing Data

- Early on, sensing data was relatively expensive, and focus was on extraction of information from a relatively small amount of data
- Many applications of nowadays have high-rate sensing and measurement requirements. This has changed the picture

Processing, transporting and storing large-scale sensing/measurement data has become a major issue

In this presentation we discuss how we can record a reduced form of the data that somehow still retains the original information of interest

Outline of the Presentation

1. Three motivating examples
2. A brief introduction to compressive sensing
3. A glance at some ongoing work at Harvard

I recently redirected my lab at Harvard to work on systems and applications of compressive sensing:

Dario Vlah, Kevin Chen, Youngjune Gwon,
Chit-Kwan Lin, Tsung-Han Lin and Steve Tarsa

Our papers in this area have started to appear (e.g., **[HotCloud 2011]**).

Motivating Example 1: Spectrum Sensing

- Advanced wireless communications depends on spectrum sensing (e.g., cognitive radio and white space)
- However, accurate **wideband spectrum sensing is expensive**
  - 10-MHz band sampled at Nyquist rate means 20 MHz * 32 bits per sample = 640 Mbps!
  - Progressive scanning of wideband spectrum with energy detector in fine resolution can take up to seconds
    - This is true even assuming best-case voltage-controlled oscillator (VCO) settling time of 20 usec

Motivating Example 2: IC Trojans Detection

IC Trojans are a serious problem. Detecting them is difficult:

- Trojans are tiny and can be dormant
- We can detect them by measuring the power they consume to allow wake-up
- However, this background power consumption is very small and can be easily buried in variations of fab process
- Furthermore, different test vectors will induce varying current leakages
- We must use enough statistics to assure confidence in detection
- This means a large number of test vectors. But we don't know a priori which test vectors can better reveal Trojans
- Due to limited chip I/O, **outputting power consumption measurements from a large number of test vectors for off-chip analysis is expensive**

Motivating Example 3: Stragglers Detection in Cloud Computing

- Continuous monitoring of server progress in a cloud data center is important. It enables rapid response to anomalies
- But handling the resulting torrent of measurement data poses a significant challenge
- For example, in MapReduce computation, a few Map stragglers can significantly delay the start of the Reduce phase. Since stragglers are a relative notion, their detection requires global decision
- But **gathering status information from a larger number of servers is expensive**

Fortunately, These Problems Are Often Sparse. Compressive Sensing Exploits This Property

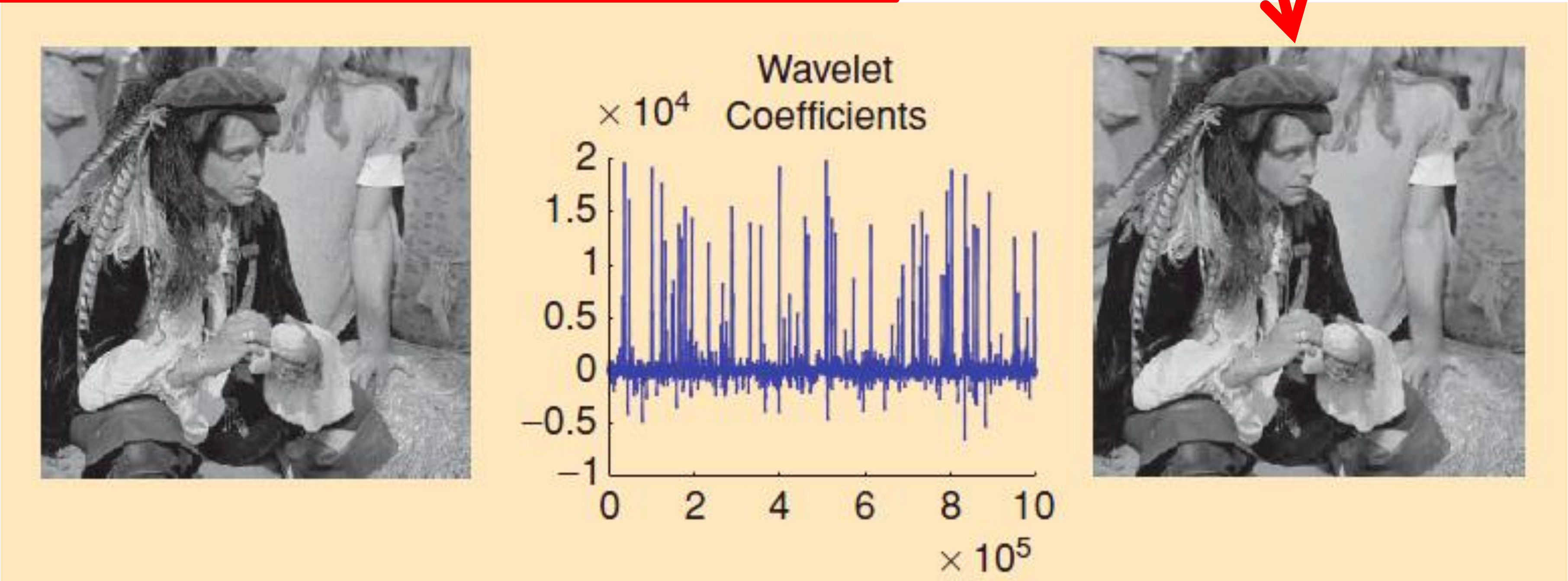
- **Spectrum sensing**: Usually at any given moment, only a small percentage of available spectrum is occupied
- **IC Trojan detection**: Trojans clearly reveal themselves only under a small fraction of test vectors. Further, they are only cover a very small portion of a chip
- **Stragglers detection**: We only need to detect stragglers which are minority

"Anomalies are norm"
Compressive sensing exploits this

Key Observation: Signal Is Often Sparse

We know it is sufficient to keep just a few pieces of information but don't necessarily know a priori their locations

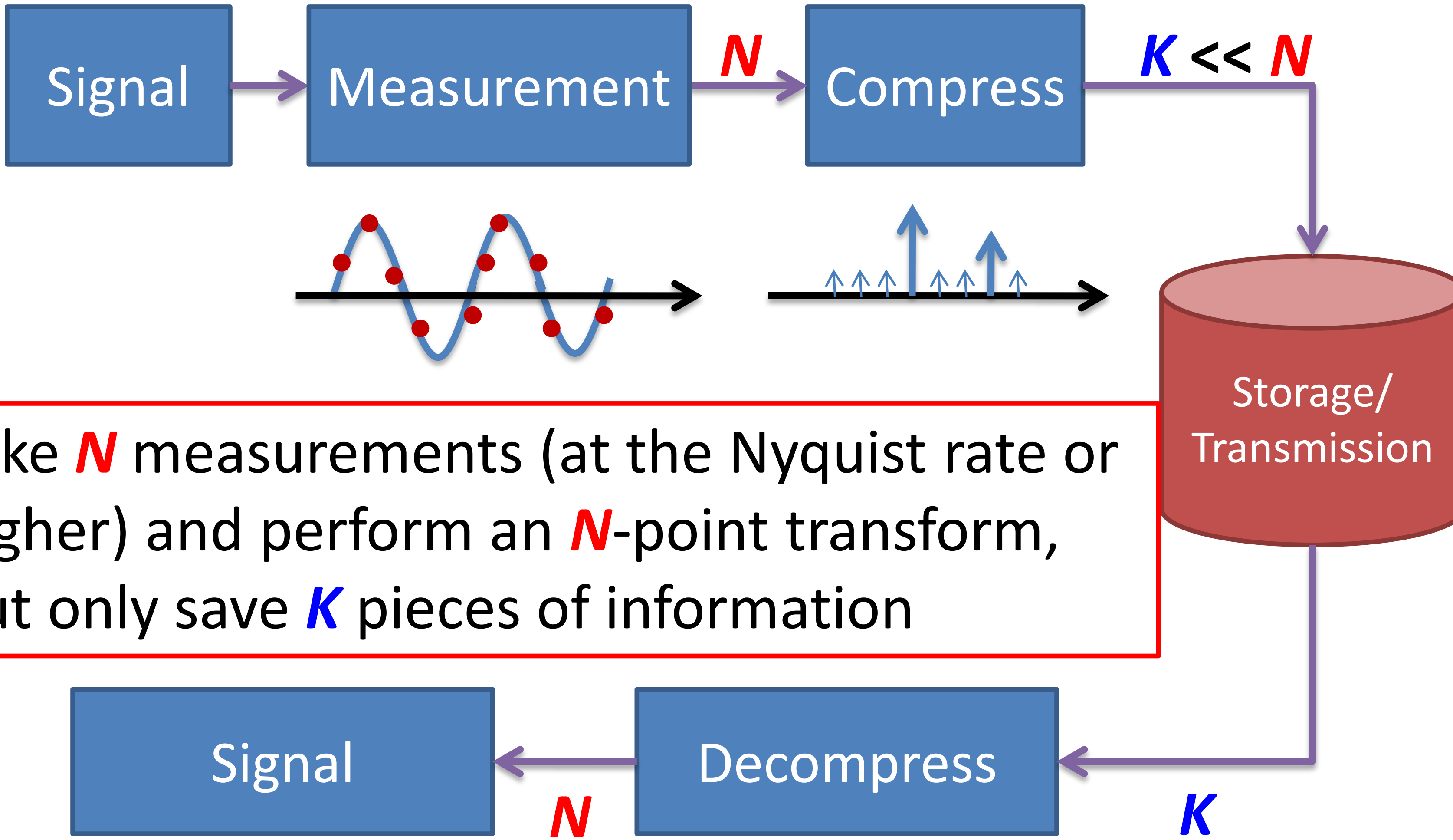
97.5% of coefficients thrown out



(a) (b) (c)

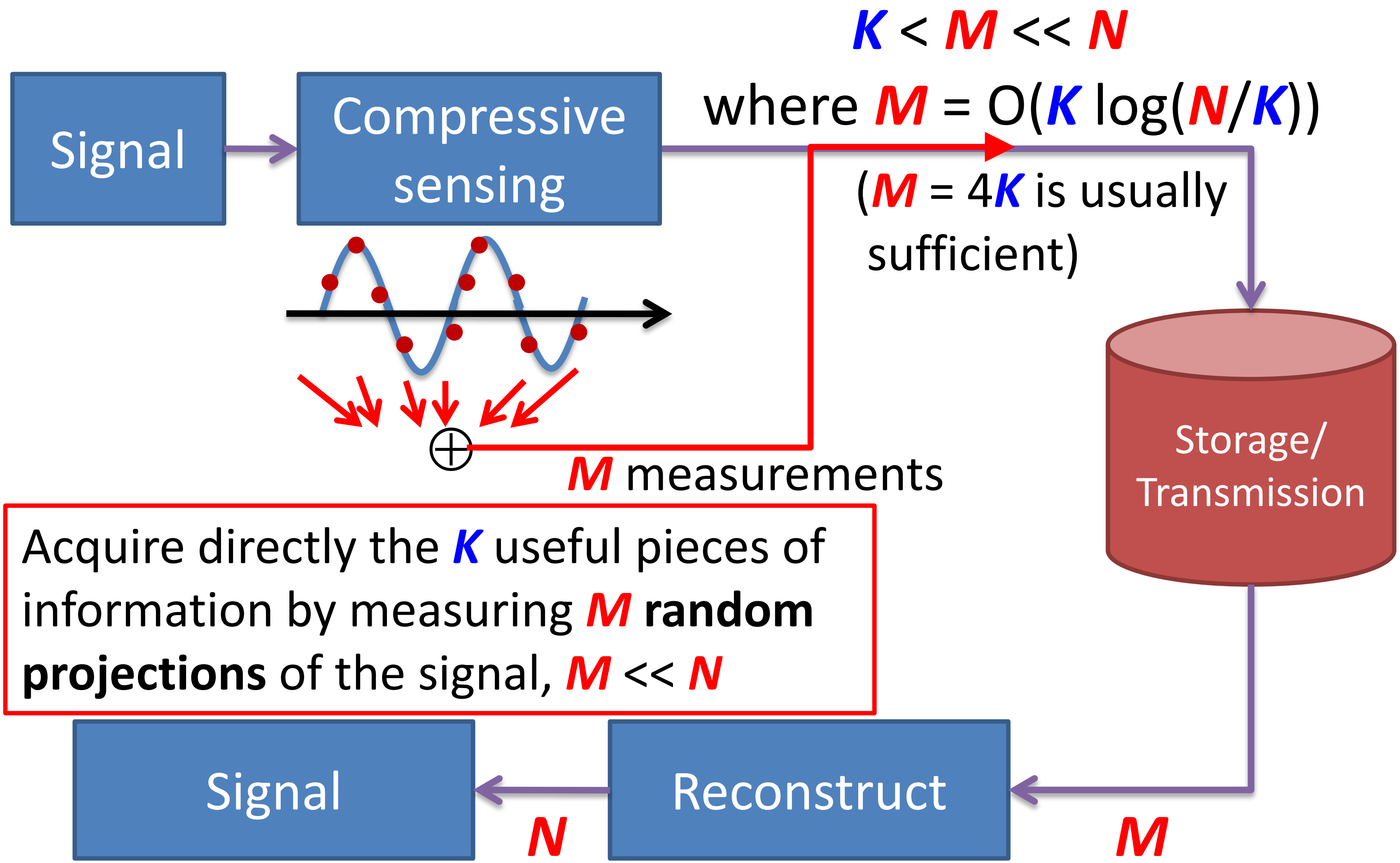
Source: [Candes, et al. 2008]

Traditional Signal Acquisition



Take ***N*** measurements (at the Nyquist rate or higher) and perform an ***N***-point transform, but only save ***K*** pieces of information

Compressive Sensing: A New Approach

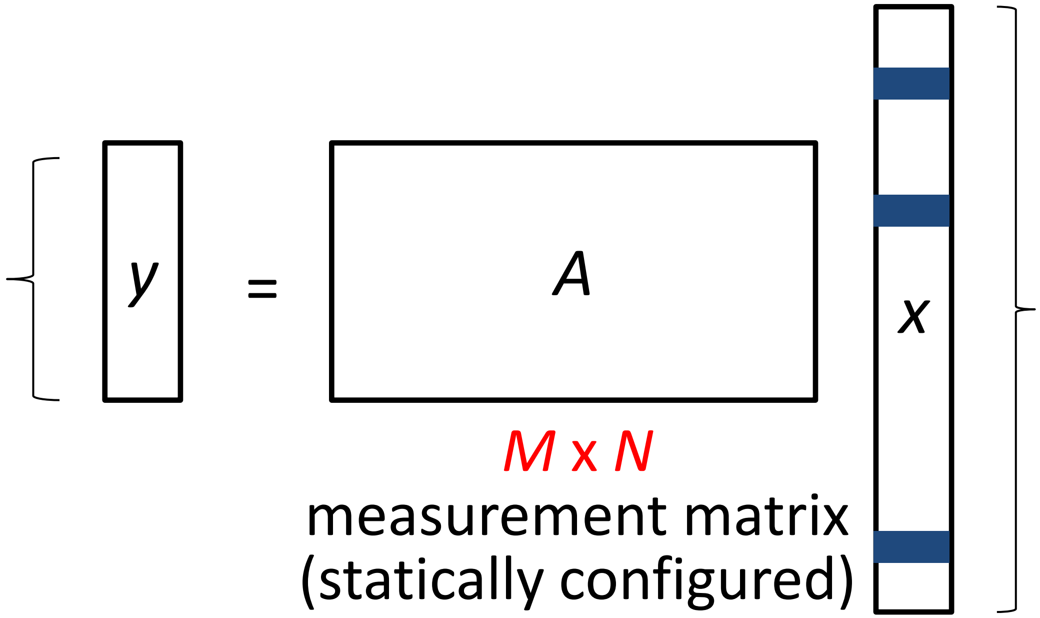


[E. Candès, J. Romberg, and T. Tao 2006], [D. Donoho 2006]

A Mathematical Formulation of Compressive Sensing

M measurements

(with *M* << *N*)



M x *N*
measurement matrix
(statically configured)

Input signal of length *N*

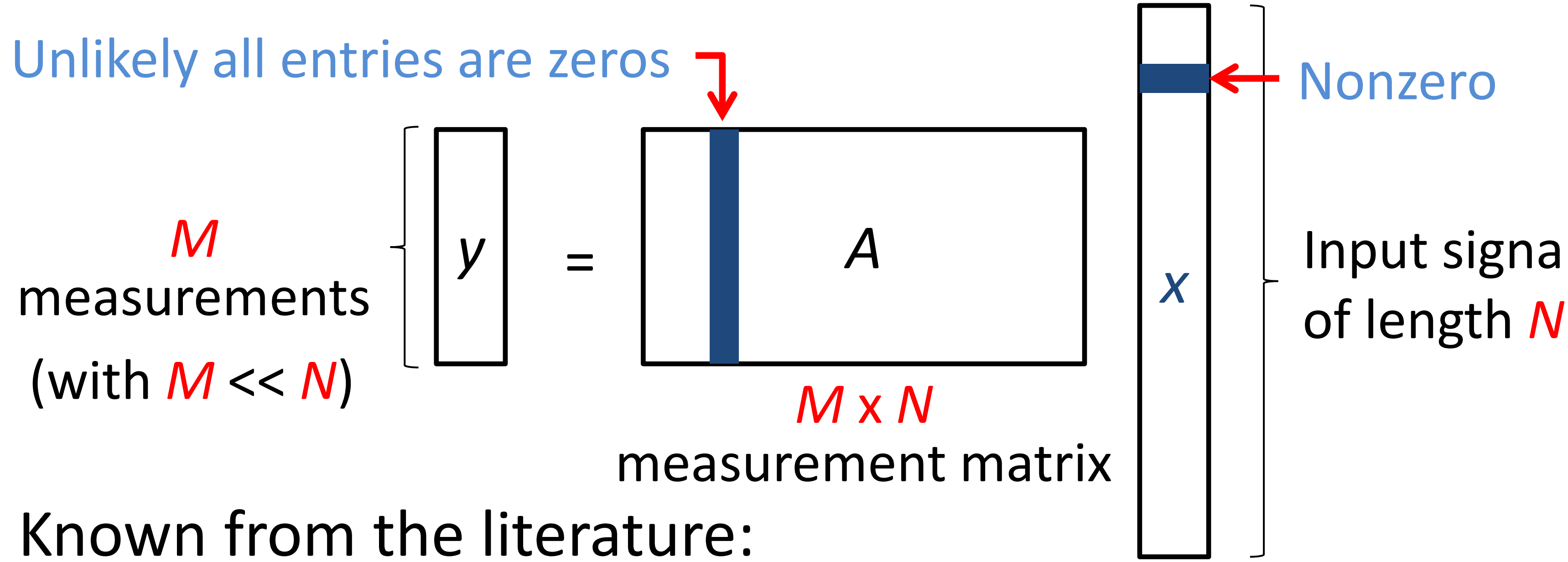
E.g., only *K* = 3 nonzeros, and *N* = 1,000

We want to recover *x* from *y*

This is an ill-posed inverse problem since *M* << *N*; there are infinitely many solutions for *x*.

However, under the assumption that *x* is sparse, finding a unique solution is still possible

Solution Is Attainable Via Optimization



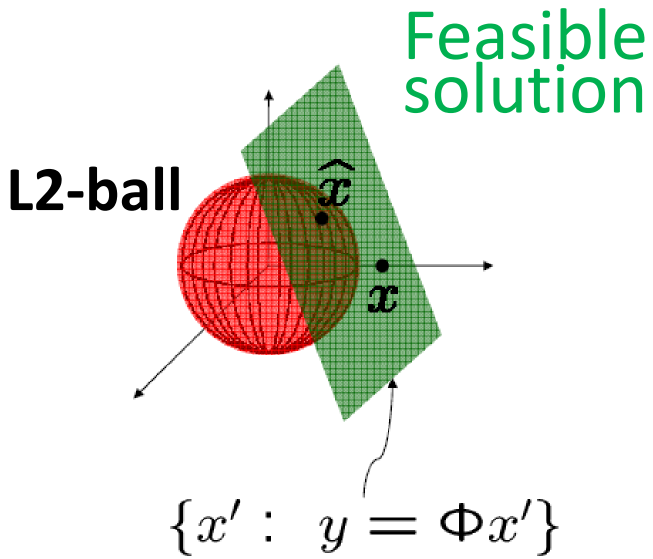
Known from the literature:

(1) If A is a random matrix with $\boxed{M \geq O(K \log(N/K))}$, then A satisfies the *Restricted Isometry Property* (RIP) with high probability

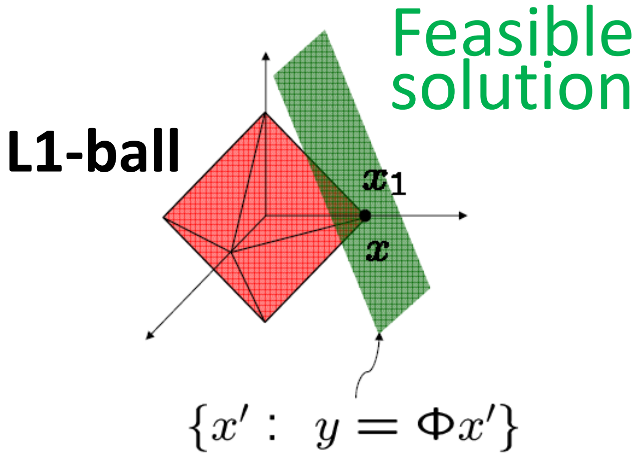
(2) If A satisfies the RIP, then a $\boxed{K\text{-sparse signal } x}$ can be recovered exactly by solving a linear programming problem

$$\min_{x \in R^N} \|x\|_1 \quad \text{subject to} \quad y = Ax$$

Why L1-norm Minimization, Not L2?



Intersecting point $\hat{x}$ is not on an axis, so it is not a sparse solution



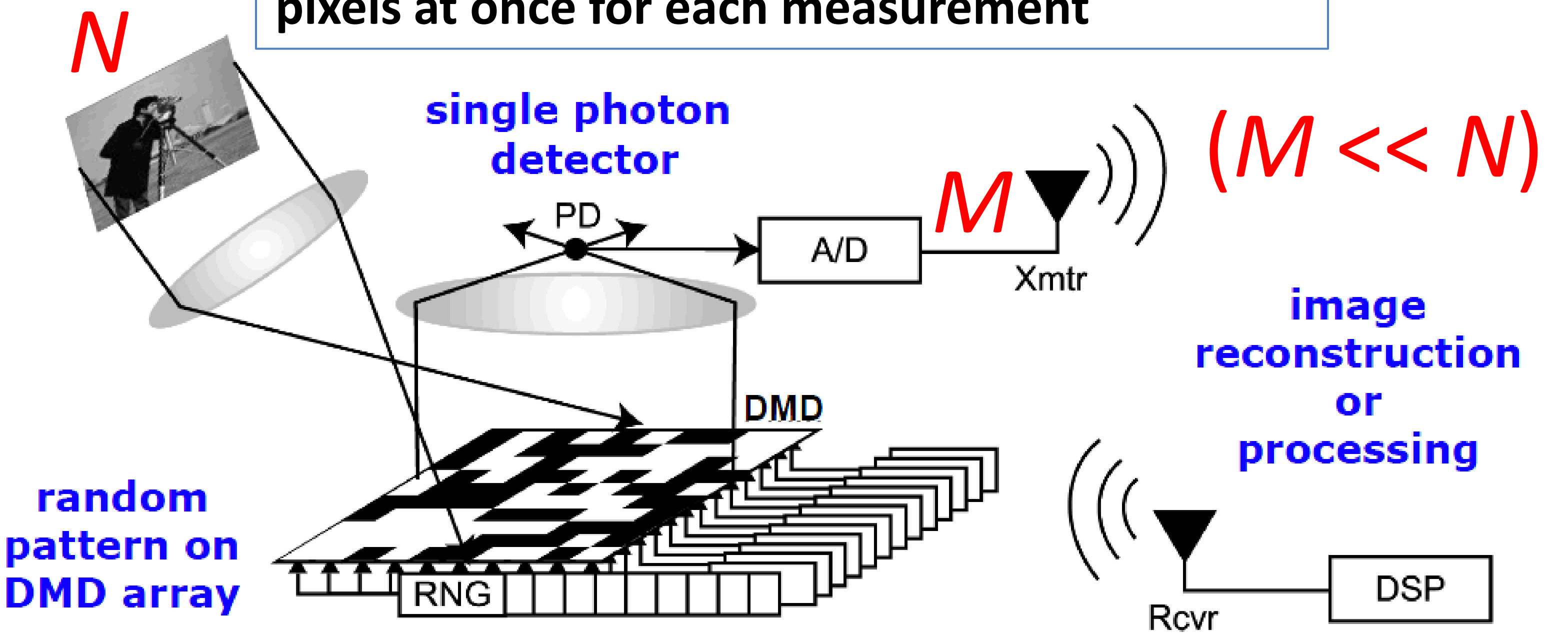
Intersecting point x is on an axis, so it is a sparse solution

Source: [Candes, et al. 2008]

An Application Example

Rice Single-Pixel CS Camera

Photo detector sums *N* randomly selected pixels at once for each measurement



Compressive Sensing Encoding: *M* Random Linear Combinations of *N* Inputs with *M* << *N*

$$\begin{array}{cccc} & \text{Measurements} & \text{Measurement Matrix} & \text{Input Signal} \\ M \{ & [y] & = [\quad \varphi \quad] & [x] \; \} N \end{array}$$

$$\boxed{y = \varphi x}$$

$$y_1 \quad y_2 \quad \cdots \quad y_M \leftarrow \bigoplus \leftarrow \times a_{i1}\, x_1,\ \times a_{i2}\, x_2,\ \ldots,\ \times a_{iN}\, x_N$$

Encoding is nonadaptive in the sense that it is static and there is no need to comprehend the input signal

Compressive Sensing Decoding: May Incorporate Any Transform ψ That Reveals Sparsity

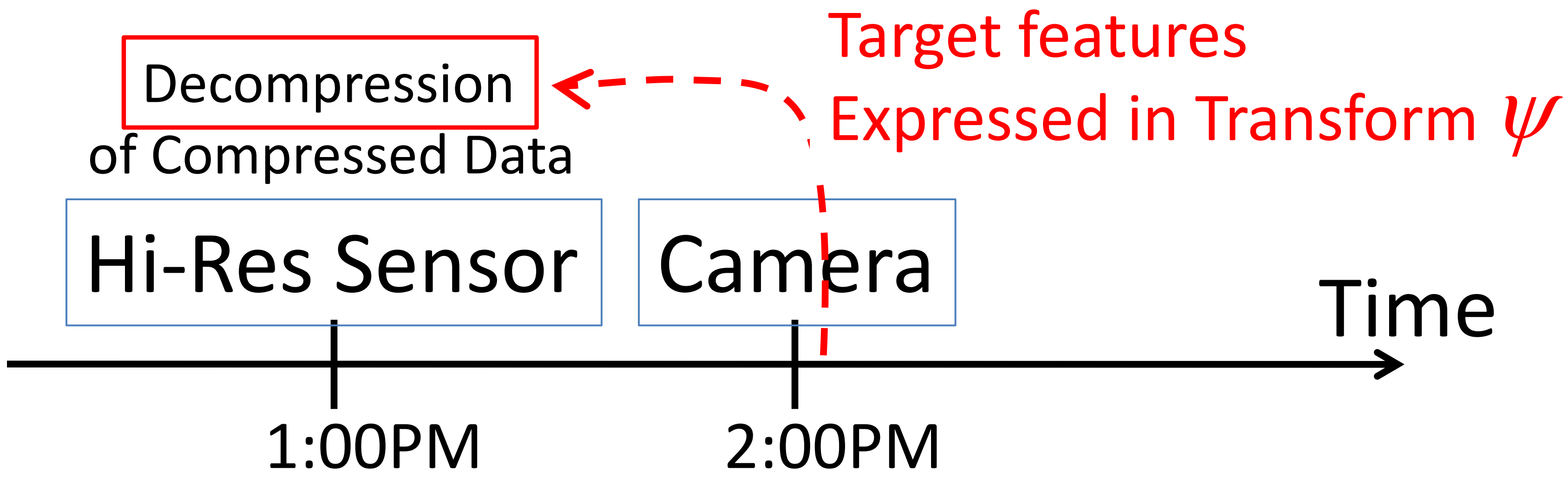
- Given measurements $\boxed{y = \varphi x}$
- Suppose x is sparse in transform ψ, i.e., $x = \psi s$, where $\boldsymbol{s}$ is sparse
  - ψ can be a discrete cosine transform (DCT), a wavelet transform, and any other transform
- Then we decode $\boldsymbol{s}$ (and thus $x = \psi s$) using:

$$\boxed{y = \varphi x = (\varphi\psi)s}$$

- This means at the decoding time we can choose an appropriate ψ for the decompression. Different applications may use different ψ tailored to each individual application

When we write $y = Ax$, we actually mean A is $\varphi\psi$ and x is s

We Can Go Back to Decode Compressed Traces Using Features Identified after Encoding



- 1:00PM: Sense an area Q with compressive sensing
- 2:00PM: Detect target *X* by a camera
- 2:01PM: Decompress the measurements taken in area Q with special transforms corresponding to characteristics associated with target *X*

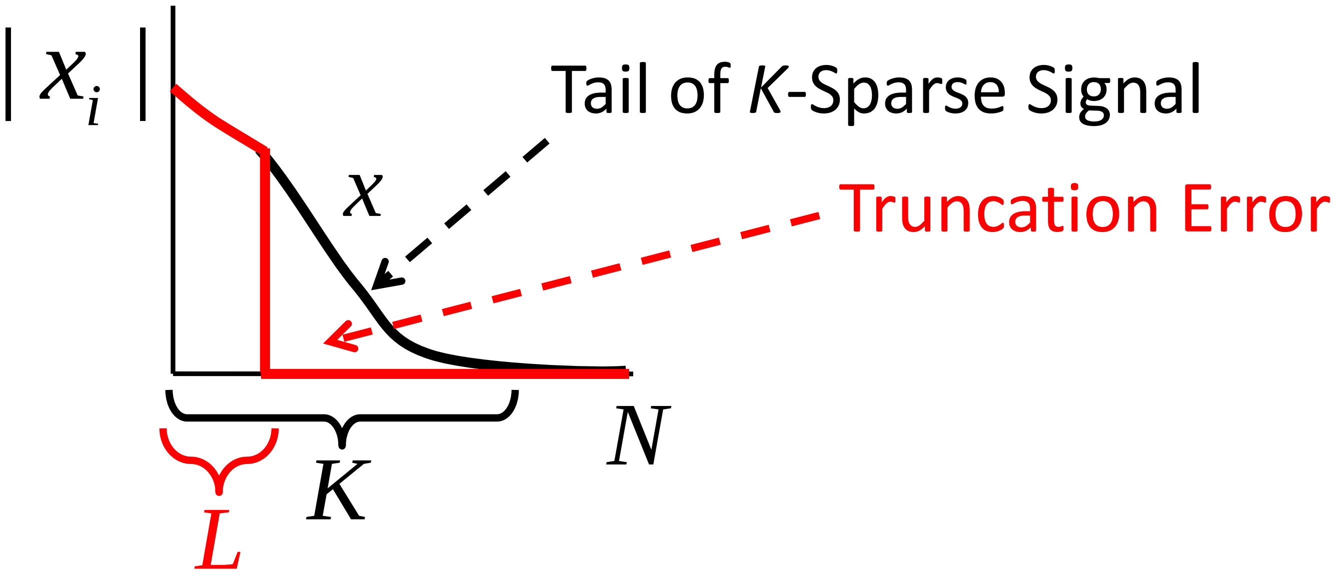
Compressive Sensing: A Summary

- Simple encoding: Allow nonadaptive encoding without comprehending the signal (amenable to high-speed analog computing)
- Adaptive decoding: Allow decoding in optimizing transforms which we may know only after the encoding time

Hundreds of references on the subject (exponentially increasing!)

See, e.g., “Compressive Sensing: The Big Picture” (http://sites.google.com/site/igorcarron2/cs)

A Fundamental Error Property of Compressive Sensing



- Error in Recovered Solution = Area of the Tail
- "Largest First" property in recovery

Many Potential Applications: Beyond Traditional Image Processing

Work in progress at Harvard:

1. Collaborative sensing via UAV
2. Combining over software-defined radio
3. Status reporting for clouds
4. Compressive spectrum sensing
5. IC Trojan detection

Conclusion

- Often in practice, anomalies are norm. Compressive sensing exploits this
- Compressive sensing is a potentially revolutionary technology---measurements are not required to satisfy Nyquist
- It is a new game. E.g., given measurements $\mathbf{y} = A\mathbf{x}$, we recover $\boldsymbol{x}$ using minimization rather than A^{-1}
- While theory is still under active development, there is enough evidence already that the approach has broad applicability
- It is a time to explore applications and technologies (e.g., combining in the analog domain and parallelizing decoding using clouds)