

# "Magic" Reconstruction: Compressed Sensing

By Cleve Moler

When I first heard about compressed sensing, I was skeptical. There were claims that it reduced the amount of data required to represent signals and images by huge factors and then restored the originals exactly. I knew from the Nyquist-Shannon sampling theorem that this is impossible. But after learning more about compressed sensing, I've come to realize that, under the right conditions, both the claims *and* the theorem are true.

The Nyquist-Shannon sampling theorem states that to restore a signal exactly and uniquely, you need to have sampled with at least twice its frequency. Of course, this theorem is still valid; if you skip one byte in a signal or image of white noise, you can't restore the original. But most interesting signals and images are not white noise. When represented in terms of appropriate basis functions, such as trig functions or wavelets, many signals have relatively few non-zero coefficients. In compressed (or compressed) sensing terminology, they are *sparse*.

## The Underlying Matrix Problem

Naturally, the aspect of compressed sensing that first caught my attention was the underlying matrix computation. A raw signal or image can be regarded as a vector  $f$  with millions of components. We assume that  $f$  can be represented as a linear combination of certain *basis functions*:

$$f = \Psi c$$

The basis functions must be suited to a particular application. In our example,  $\Psi$  is the discrete cosine transform. We also assume that most of the coefficients  $c$  are effectively zero, so that  $c$  is sparse. Sampling the signal involves another linear operator,

$$b = \Phi f$$

In our example,  $b$  is a few random samples of  $f$ , so  $\Phi$  is a subset of the rows of the identity operator. But more complicated sampling operators are possible.

To reconstruct the signal, we must try to recover the coefficients by solving

$$Ax = b, \text{ where } A = \Phi\Psi$$

Once we have the coefficients, we can recover the signal itself by computing

$$f \approx \Psi x$$

Since this is a compression,  $A$  is rectangular, with many more columns than rows. Computing the coefficients  $x$  involves solving an underdetermined system of simultaneous linear equations,  $Ax = b$ . In this situation, there are many more unknowns than equations. The key to the almost magical reconstruction process is to impose a *non-linear* regularization involving the  $l_1$  (that's "ell-sub-1") norm.

I described a tiny version of this situation in a 1990 Cleve's Corner column entitled "The World's Simplest Impossible Problem." I am thinking of two numbers whose average is 3. What are the numbers? After complaining that I haven't given you enough information, you might answer 2 and 4. If you do, you have unconsciously imposed a kind of regularization that requires the result to be two distinct integers.

MATLAB® can provide two different answers. Algebraically, the problem is a 1-by-2 system of linear equations with matrix

$$A = [1/2 \ 1/2]$$

and right-hand side

$$b = 3$$

We want to find a 2-vector  $y$  that solves  $Ay = b$ . The minimum norm least squares solution is computed by the pseudoinverse,

$$y = \text{pinv}(A) * b$$

$$y =$$

$$3$$

$$3$$

But backslash generates a different solution:

$$x = A \backslash b$$

$$x =$$

$$6$$

$$0$$

Both solutions are valid, but human puzzle-solvers rarely mention them. Notice that the solution computed by backslash is sparse; one of its components is zero.

## A Larger Instance of the Same Task

The signal or image restoration problem is a larger instance of the same task. We

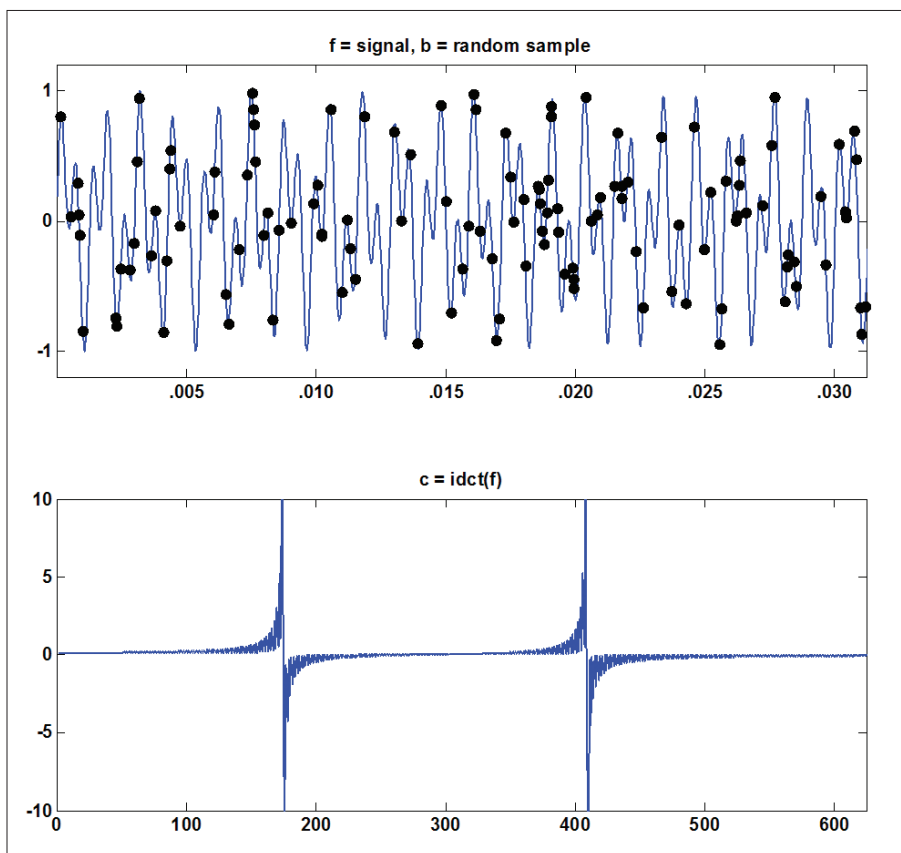


FIGURE 1. Top: Random samples of the original signal generated by the “A” key on a touch-tone phone. Bottom: The inverse discrete cosine transform of the signal.

are given thousands of weighted averages of millions of signal or pixel values. Our job is to re-generate the original signal or image. We have the compressed sample,  $b$ , and we need to solve  $Ax = b$ . There is a huge number of possible solutions. The basis for picking the right one involves vector norms. The familiar Euclidean distance norm,  $l_2$ , is

$$l_2: \text{norm}(x, 2) = \sqrt{\text{sum}(x.^2)}$$

The “Manhattan” norm,  $l_1$ , is named after travel time along the square grid formed by city streets.

$$l_1: \text{norm}(x, 1) = \text{sum}(\text{abs}(x))$$

The notation  $l_0$  is used for a function that only counts the number of nonzero components. It’s actually not a norm.

$$l_0: \text{norm}_0(x) = \text{sum}(x \neq 0)$$

These norms impose three different nonlinear regularizations, or constraints, on our underdetermined linear system.

From all  $x$  that satisfy  $Ax = b$ , find an  $x$  that minimizes  $l_p(x)$

For our 20-year-old example,  $l_2(y)$  is less than  $l_2(x)$ , and  $l_1(y)$  and  $l_1(x)$  happen to be equal, but  $l_0(y)$  is greater than  $l_0(x)$ .

## Sparsity and the $l_1$ Norm

The keys to compressed sensing are sparsity and the  $l_1$  norm. If the expansion of the original signal or image as a linear combination of the selected basis functions has many zero coefficients, then it’s often possible to reconstruct the signal exactly. In principle, computing this reconstruction should involve counting nonzeros with  $l_0$ . This is a combinatorial problem whose computational complexity makes it impractical. (It’s NP-hard.) Fortunately, as David Donoho, Emmanuel Candès, and Terence Tao have shown,  $l_0$  can be replaced by  $l_1$ . They explain that “with overwhelming probability” the two problems have the same solution. The  $l_1$  computation is practical because it can be posed as a linear programming problem and solved with the traditional simplex algorithm or modern interior point methods.

## An Example

Our example uses the discrete cosine transform (DCT) as the basis. The signal generated by the “A” key on a touch-tone telephone is the sum of two sinusoids with incommensurate frequencies,

$$f(t) = \sin(1394\pi t) + \sin(3266\pi t)$$

If we sample this tone for 1/8 of a second at 40000 Hz, the result is a vector  $f$  of length  $n = 5000$ . The upper plot in Figure 1 shows a portion of this signal, together with some of the  $m = 500$  random samples that we have taken. The lower plot shows the coefficients  $c$ , obtained by taking the inverse discrete cosine transform of  $f$ , with two spikes at the appropriate frequencies. Because the two frequencies are incommensurate, this signal does not fall exactly within the space spanned by the DCT basis functions, and so there are a few dozen significant nonzero coefficients.

For our example, the condensed signal is a vector  $b$  of  $m$  random samples of the

original signal. We construct a matrix  $A$  by extracting  $m$  rows from the  $n$ -by- $n$  DCT matrix

```
D = dct(eye(n,n));
A = D(k,:)
```

where  $k$  is the vector of indices used for the sample  $b$ . The resulting linear system,  $Ax = b$ , is  $m$ -by- $n$ , which is 500-by-5000. There are 10 times as many unknowns as equations.

To reconstruct the signal, we need to find the solution to  $Ax = b$  that minimizes the  $l_1$  norm of  $x$ . This is a nonlinear optimization problem, and there are several MATLAB based programs available to solve it. I have chosen to use  $l_1$ -MAGIC, written by Justin Romberg and Emmanuel Candès when they were at Caltech. The upper plot in Figure 2 shows the resulting solution,  $x$ . We see that it has relatively few large components and that it closely resembles the DCT of the original signal. Moreover, the discrete cosine transform of  $x$ , shown in the lower plot, closely resembles the original signal. If audio was available, you would be able to hear that the two commands `sound(f)` and `sound(dct(x))` are nearly the same.

For comparison, Figure 3 shows the traditional  $l_2$ , or least squares, solution to  $Ay = b$ , computed by

```
y = pinv(A)*b
```

There is a slight hint of the original signal in the plots, but `sound(dct(y))` is just a noisy buzz.

It is too early to predict when, or if, we might see compressed sensing in our cell phones, digital cameras, and magnetic resonance scanners, but I find the underlying mathematics and software fascinating. ■

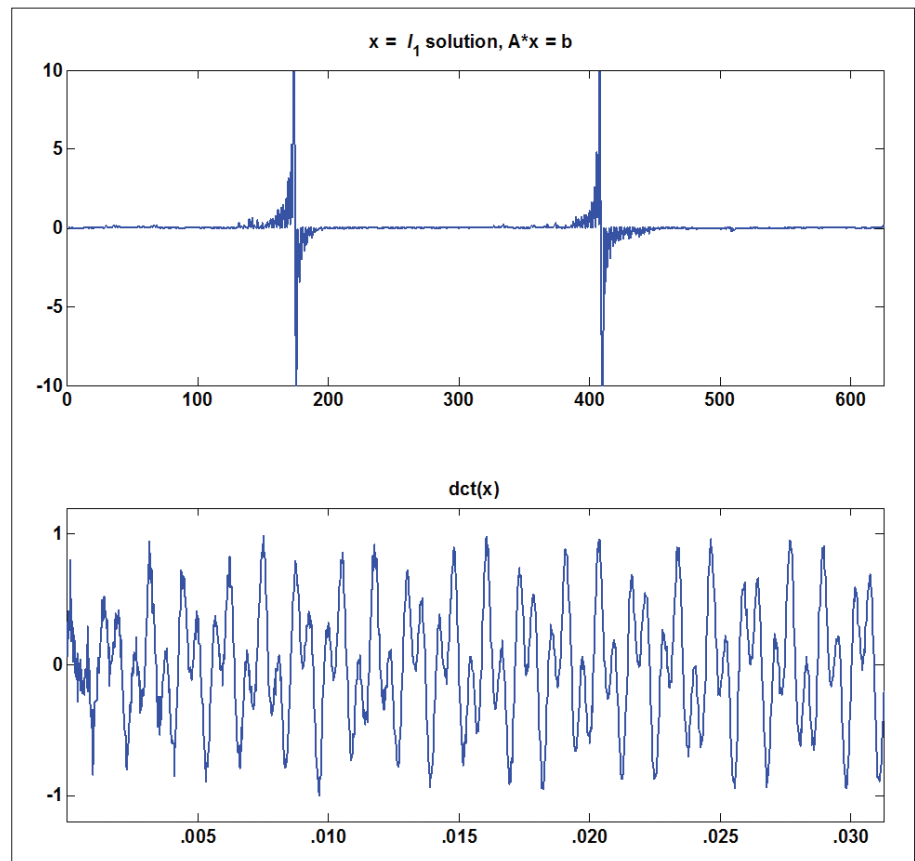


FIGURE 2. The  $l_1$  solution to  $Ax = b$  leads to  $\text{dct}(x)$ , a signal that is nearly identical to the original.

## References

### A twenty-year-old Cleve's Corner:

"The World's Simplest Impossible Problem," *News & Notes* (1990)  
[www.mathworks.com/clevescorner/dec1990](http://www.mathworks.com/clevescorner/dec1990)

### A resource providing over 600 links, including 36 that reference MATLAB:

"Compressive Sensing Resources,"  
 Rice University  
<http://dsp.rice.edu/cs>

### Two survey papers for general technical audiences:

Dana Mackenzie, "Compressed Sensing Makes Every Pixel Count," *What's Happen-*

*ing in the Mathematical Sciences* 7 (2009), American Math Society  
[www.ams.org/happening-series/hap7-pixel.pdf](http://www.ams.org/happening-series/hap7-pixel.pdf)

Bryan Hayes, "The Best Bits," *American Scientist* 97, no.4 (2009)  
[www.americanscientist.org/issues/pub/the-best-bits/7](http://www.americanscientist.org/issues/pub/the-best-bits/7)

### One of many tutorials:

Richard Baraniuk, Justin Romberg, and Michael Wakin, "Tutorial on Compressive Sensing," Information Theory and Applications Workshop (2008)  
[www.ece.rice.edu/~richb/talks/cs-tutorial-ITA-feb08-complete.pdf](http://www.ece.rice.edu/~richb/talks/cs-tutorial-ITA-feb08-complete.pdf)

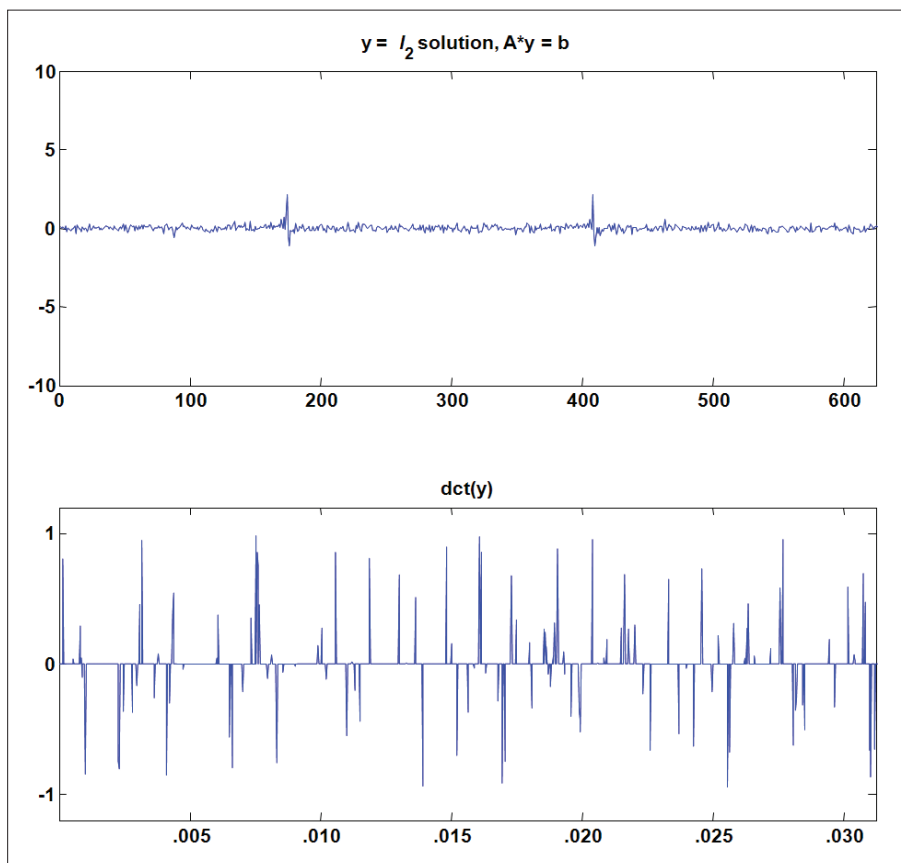


FIGURE 3. The  $l_2$  solution to  $Ay = b$  leads to  $\text{dct}(y)$ , a signal that bears little resemblance to the original.

#### MATLAB based software:

Justin Romberg,  $l_1$ -MAGIC  
[www.acm.caltech.edu/l1magic](http://www.acm.caltech.edu/l1magic)

#### The two original papers:

Emmanuel Candès, Justin Romberg, and Terence Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on Information Theory*, 52(2): 489 - 509, February 2006  
[www.acm.caltech.edu/~jrom/publications/CandesRombergTao\\_revisedNov2005.pdf](http://www.acm.caltech.edu/~jrom/publications/CandesRombergTao_revisedNov2005.pdf)

David Donoho. "Compressed Sensing," *IEEE Transactions on Information Theory*, 52(4): 1289 - 1306, April 2006  
[www-stat.stanford.edu/~donoho/Reports/2004/CompressedSensing091604.pdf](http://www-stat.stanford.edu/~donoho/Reports/2004/CompressedSensing091604.pdf)

#### An audio demonstration:

Laura Balzano, Robert Nowak, and Jordan Ellenberg, "Compressed Sensing Audio Demonstration"  
<http://sunbeam.ece.wisc.edu/csaudio/>

### Compressed Sensing

Compressed sensing promises, in theory, to reconstruct a signal or image from surprisingly few samples. Discovered just five years ago by Candès and Tao and by Donoho, the subject is a very active research area. Practical devices that implement the theory are just now being developed.

It is important to realize that compressed sensing can be done only by a compressing sensor, and that it requires new recording technology and file formats. The MP3 and JPEG files used by today's audio systems and digital cameras are already compressed in such a way that exact reconstruction of the original signals and images is impossible. Some of the Web postings and magazine articles about compressed sensing fail to acknowledge this fact.

### Learn More Online

**MATLAB CODE FOR DOWNLOAD**  
[www.mathworks.com/mlc28250](http://www.mathworks.com/mlc28250)

**CLEVE'S CORNER COLLECTION**  
[www.mathworks.com/clevescorner](http://www.mathworks.com/clevescorner)