Compressed Sensing

Sven Dorkenwald

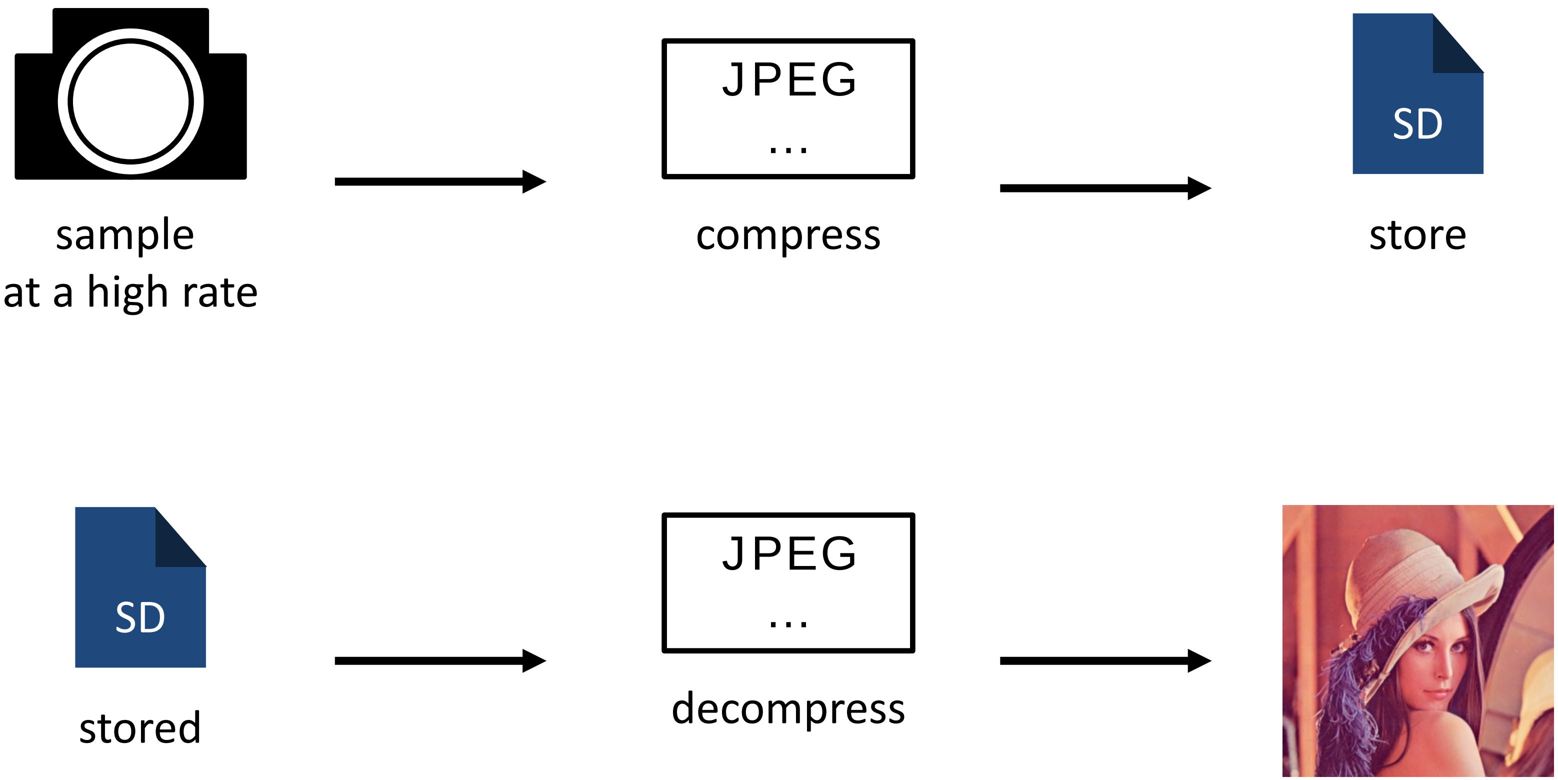
Advanced Seminar
02/06/2017

ziti, Heidelberg University
Supervisor: Prof. Dr. Peter Fischer

Outline

1. Compression after sensing
2. Why we can compress
3. Theory on Compressed Sensing
4. Example applications

Compression after sensing



Problems of „Compression after Sensing“

We likely gather too much data

Sampling with Shannon rate

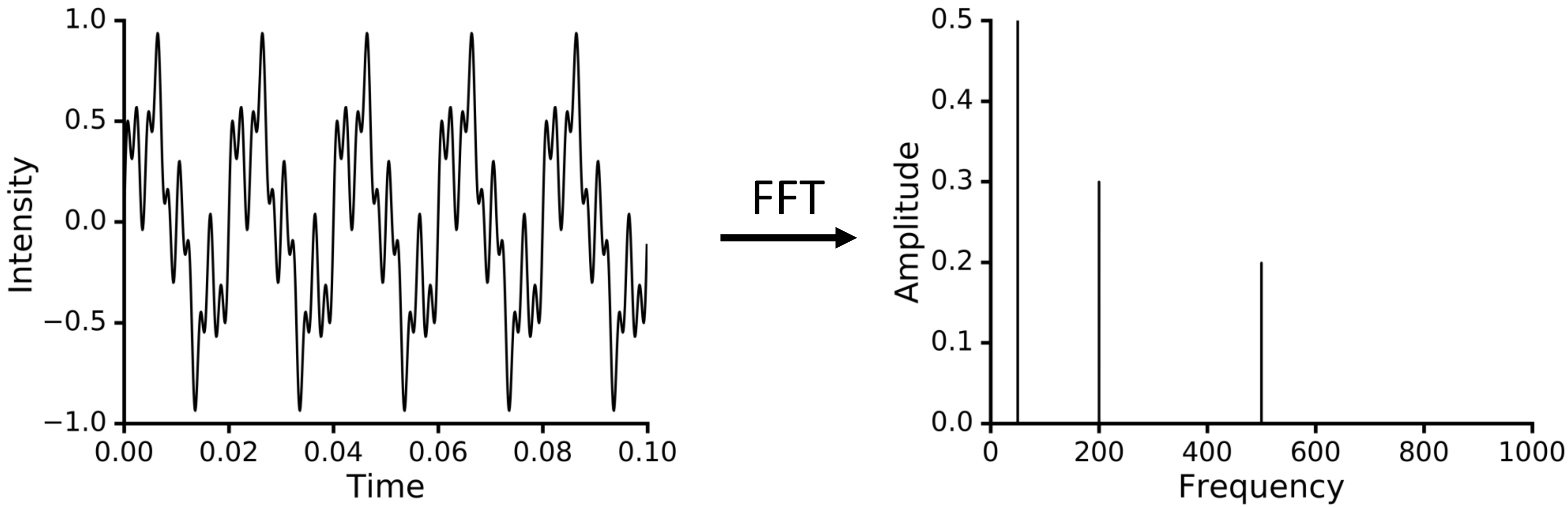
- slow
- expensive
- problematic (eg. medical applications)

Compression on recorder side

- expensive (satellites)

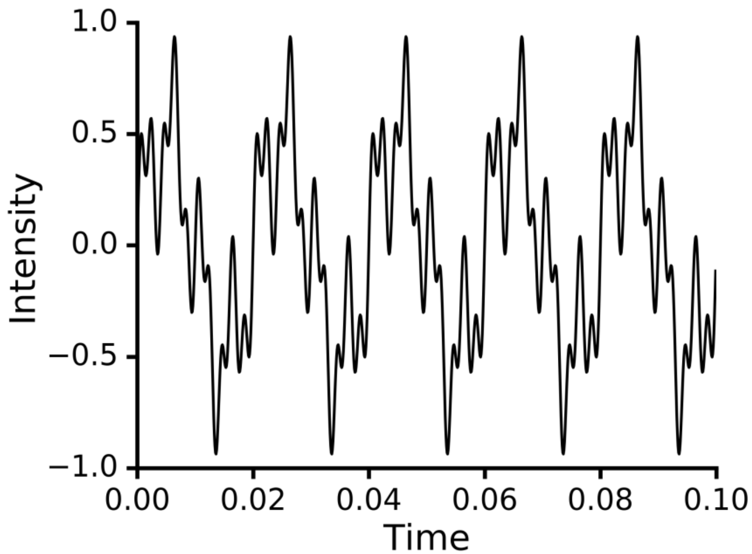
Sparsity – why we can compress

Superposition of 3 sinus waves:

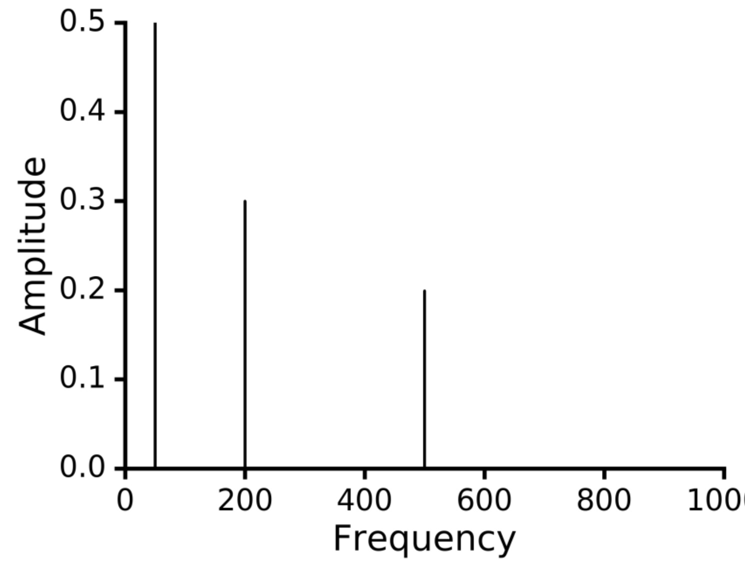


Sparsity – why we can compress

Superposition of 3 sinus waves:



FFT



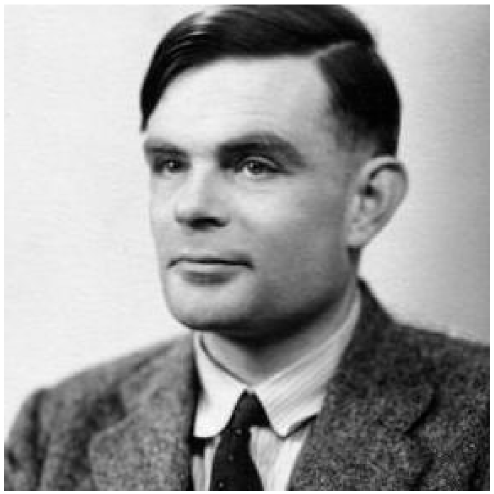
→ sampling at at least 1000Hz

twice the highest frequency
according to Shannon

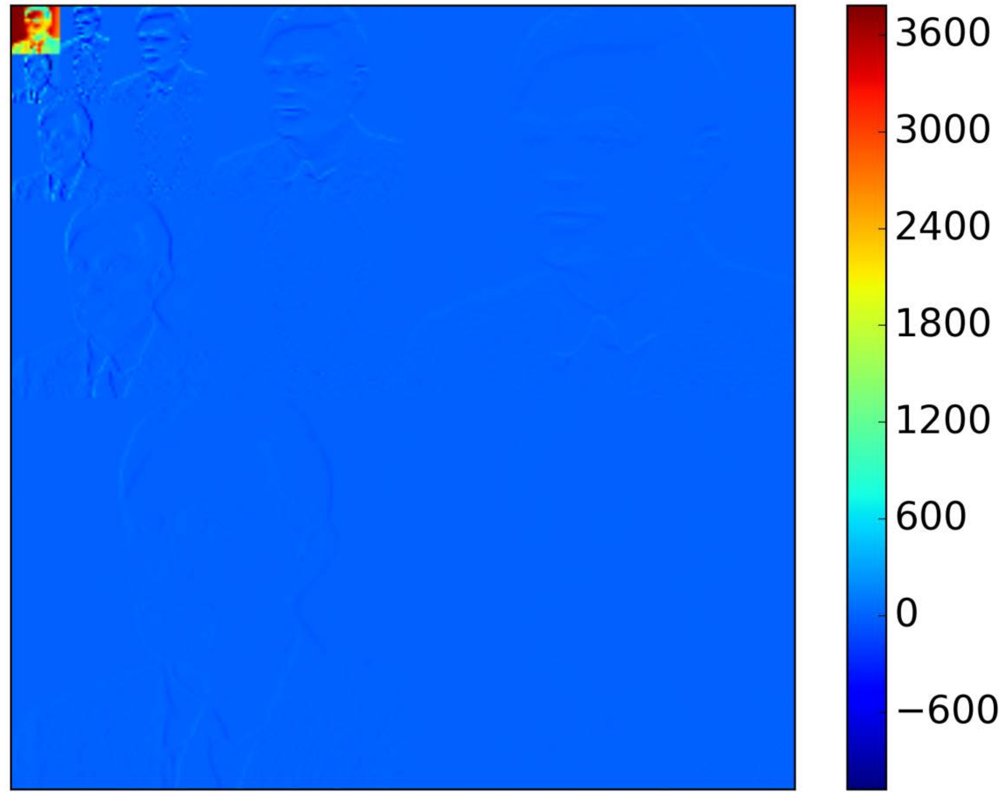
→ 6 parameters:
3 frequencies
3 amplitudes

Sparsity – why we can compress

8bit 288x288 image:

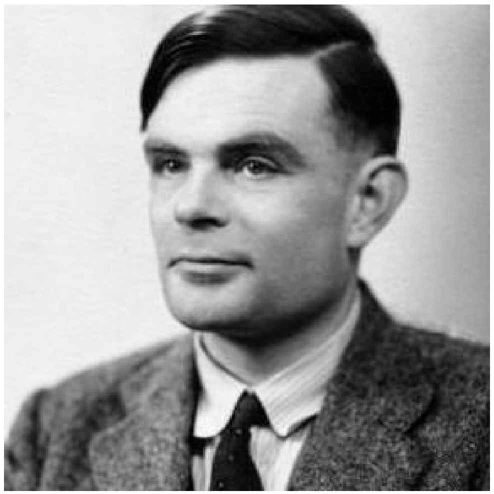


Wavelet
Trafo

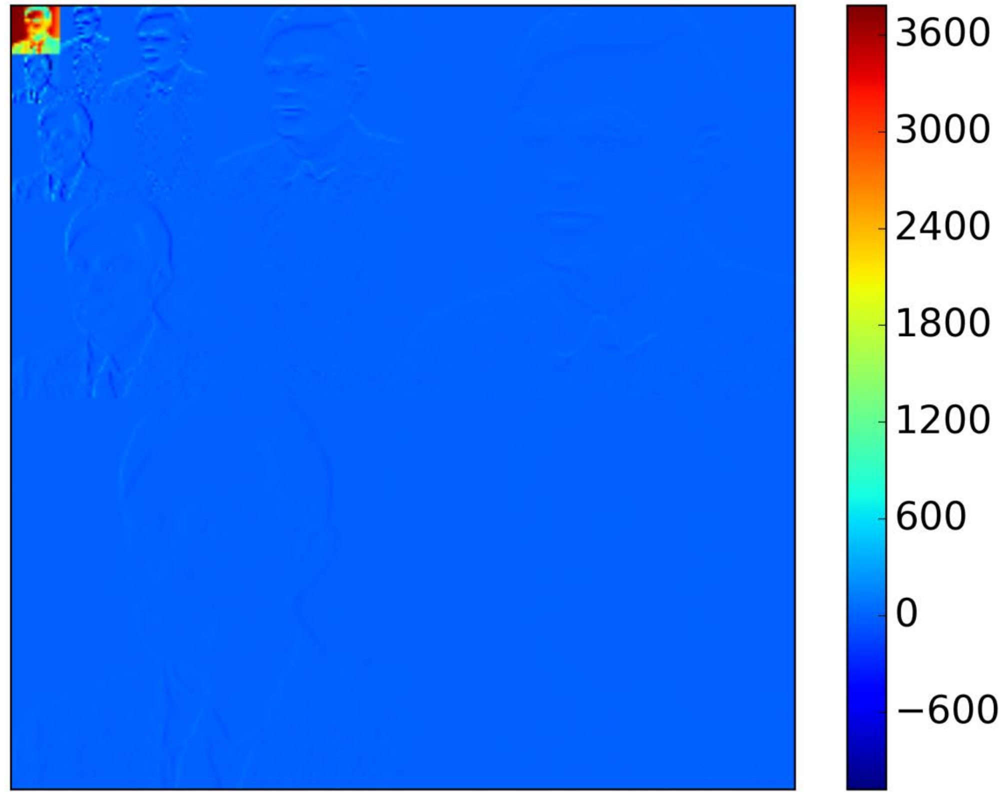


Sparsity – why we can compress

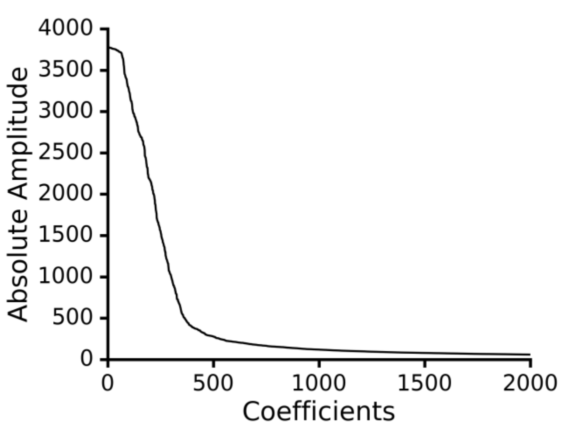
8bit 288x288 image:



Wavelet
Trafo



First 2000 of ~83k coefficients:



→ Most coefficients close to 0

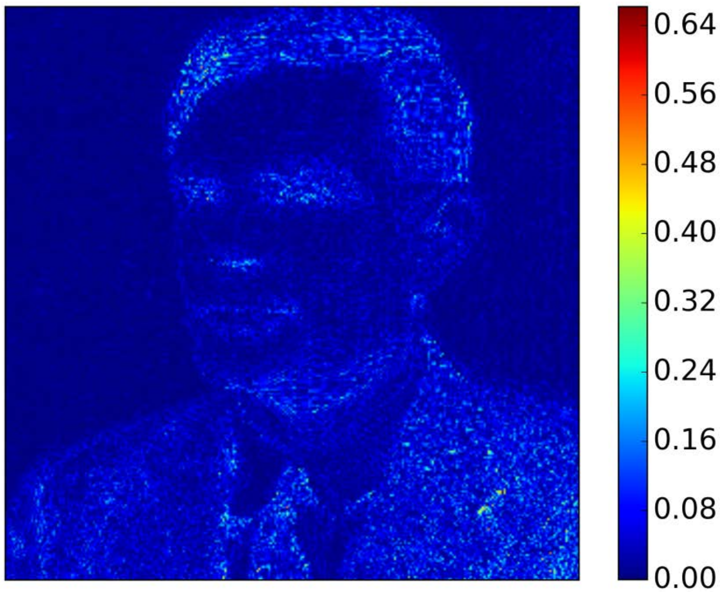
Sparsity – why we can compress

Omitting the 90% smallest coefficients
and transforming back:

Sparsity – why we can compress

Omitting the 90% smallest coefficients and transforming back:

Relative difference to original image:

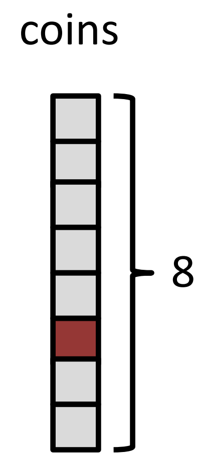


Sparsity – why we can compress

most signals are sparse in some basis

(e.g. Fourier, Wavelet)

Counterfeit example



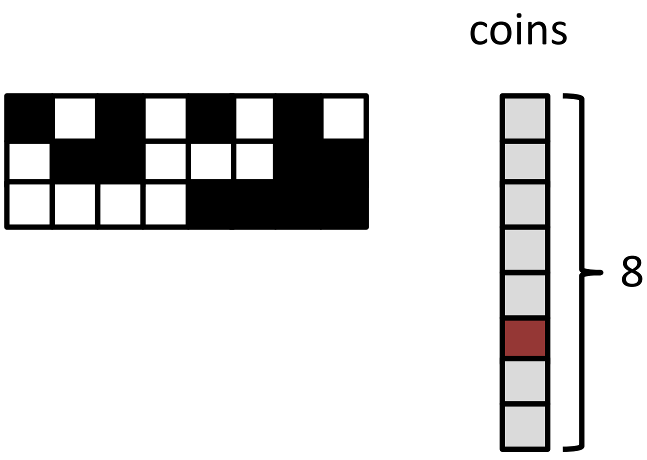
How many weightings do we need
to always find the counterfeit?

Hint 1: it is not 8

Hint 2: you may want to weigh multiple coins at once

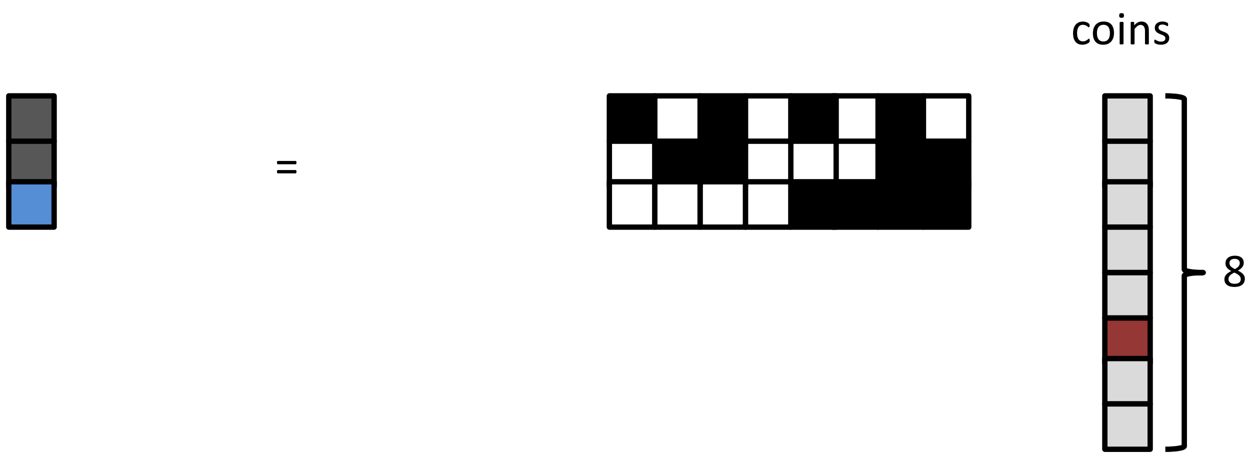
[Brein, Leise: Making Do with Less: An Introduction to Compressed Sensing]

Counterfeit example



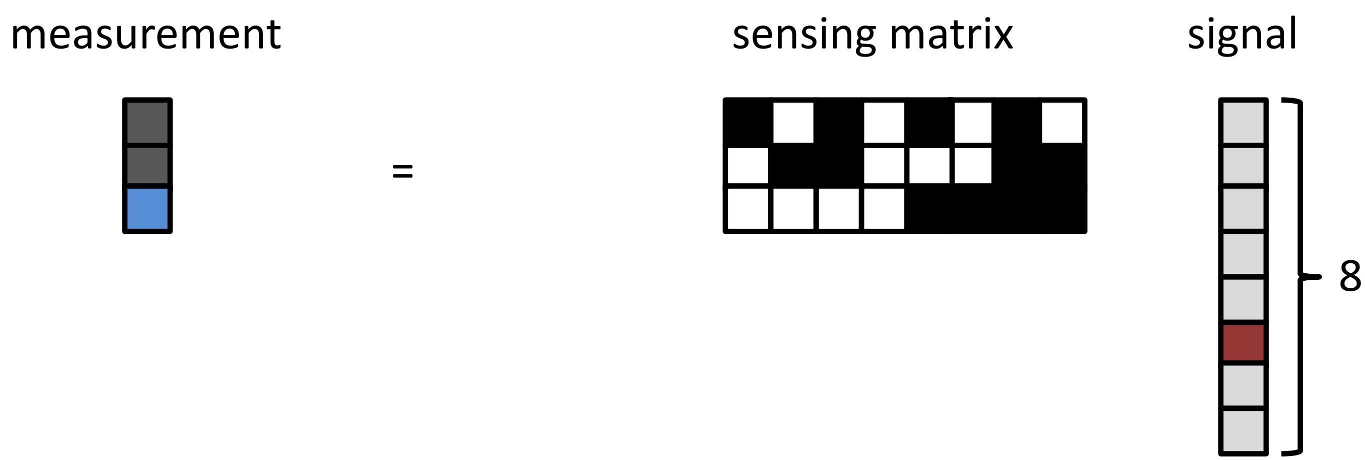
[Brein, Leise: Making Do with Less: An Introduction to Compressed Sensing]

Counterfeit example



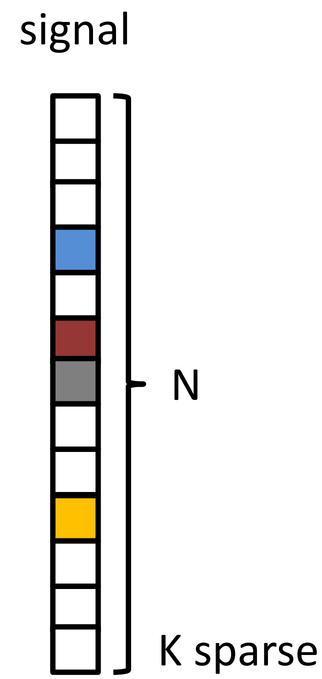
[Brein, Leise: Making Do with Less: An Introduction to Compressed Sensing]

Counterfeit example

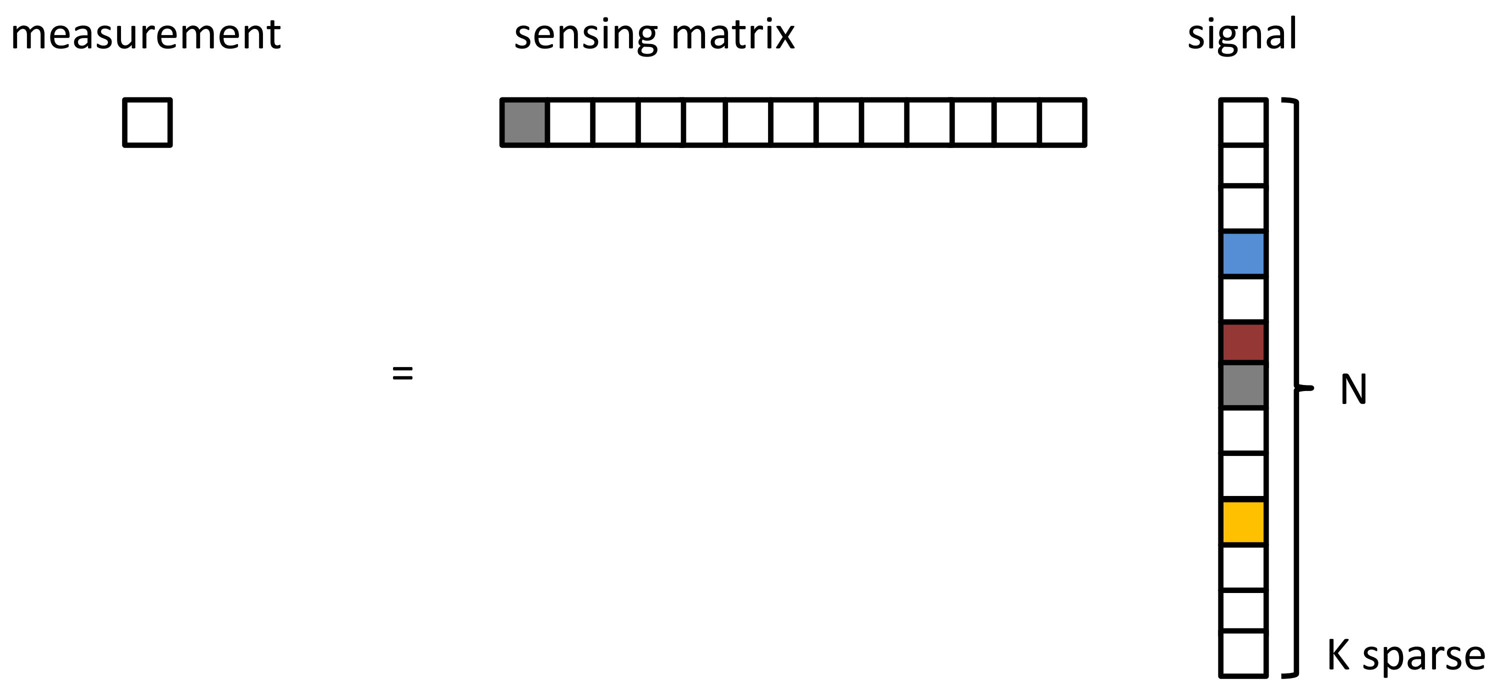


[Brein, Leise: Making Do with Less: An Introduction to Compressed Sensing]

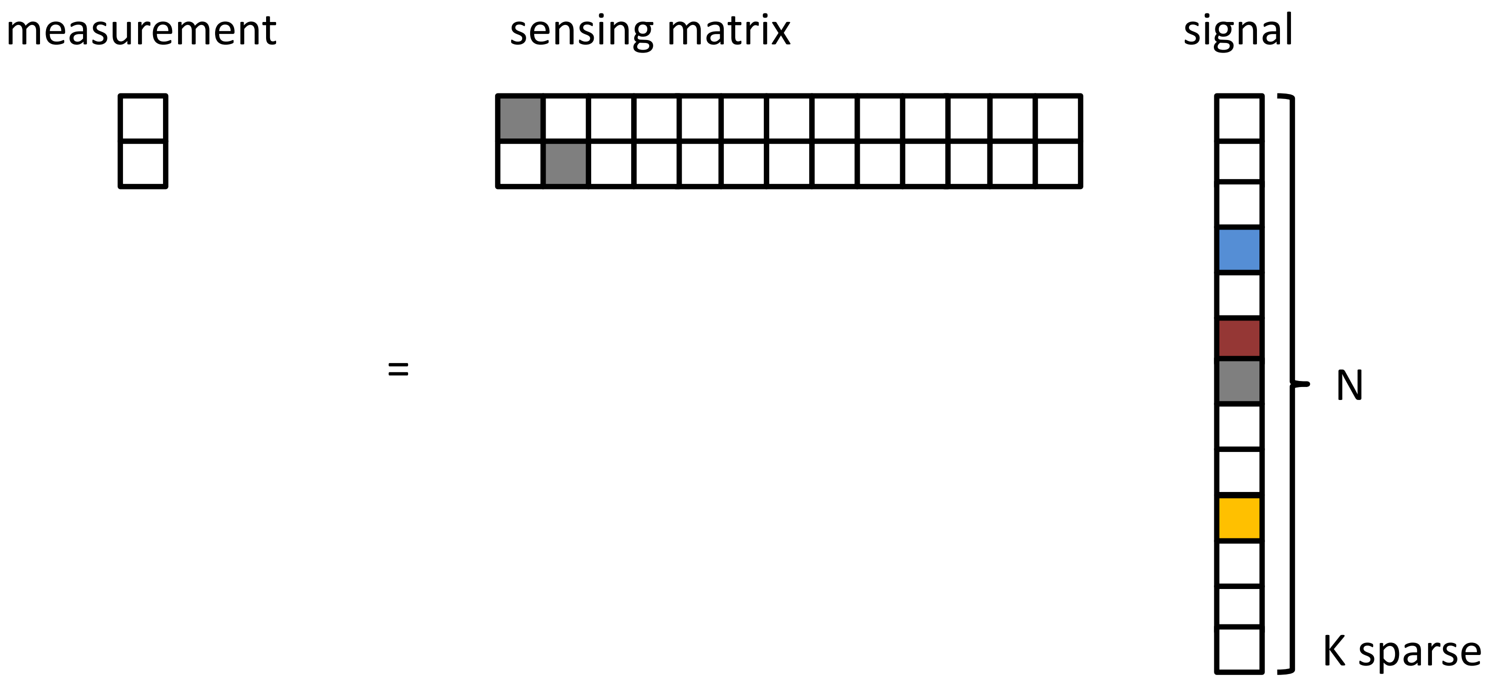
Sensing a sparse vector



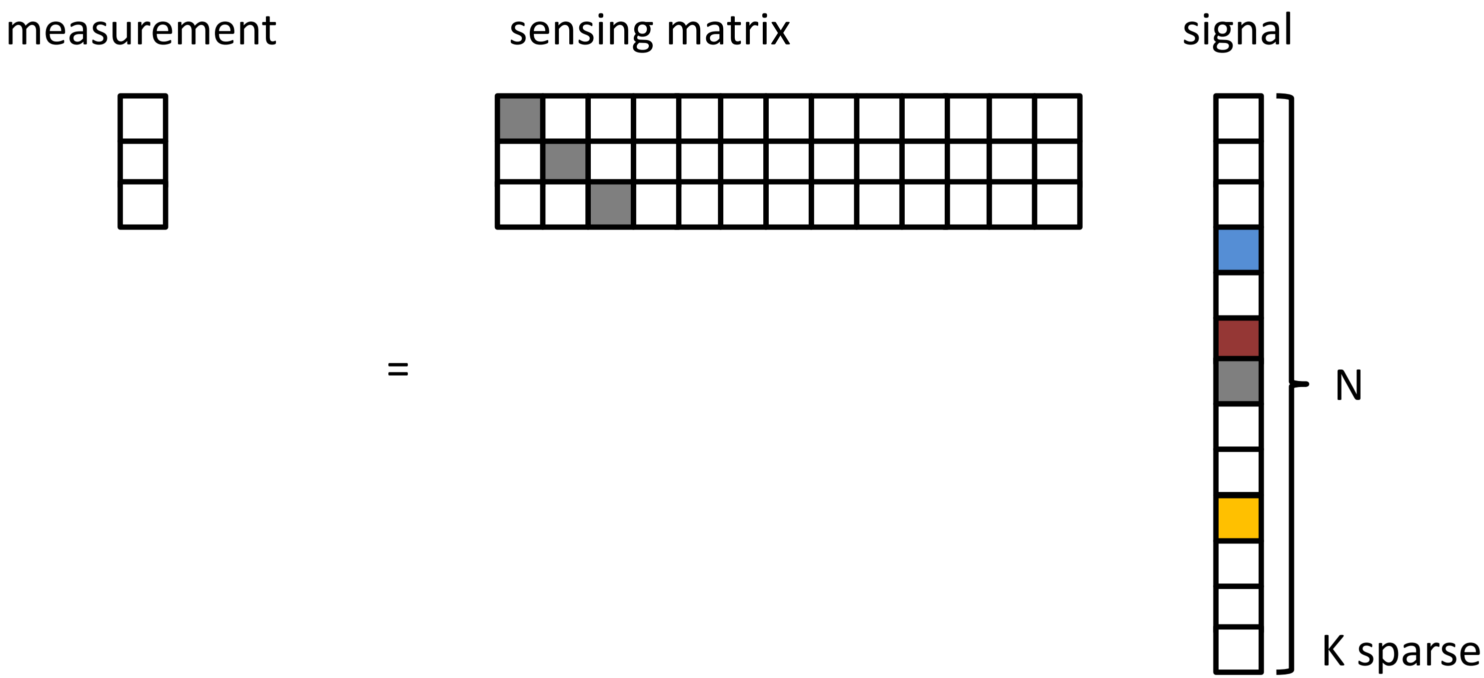
Standard approach



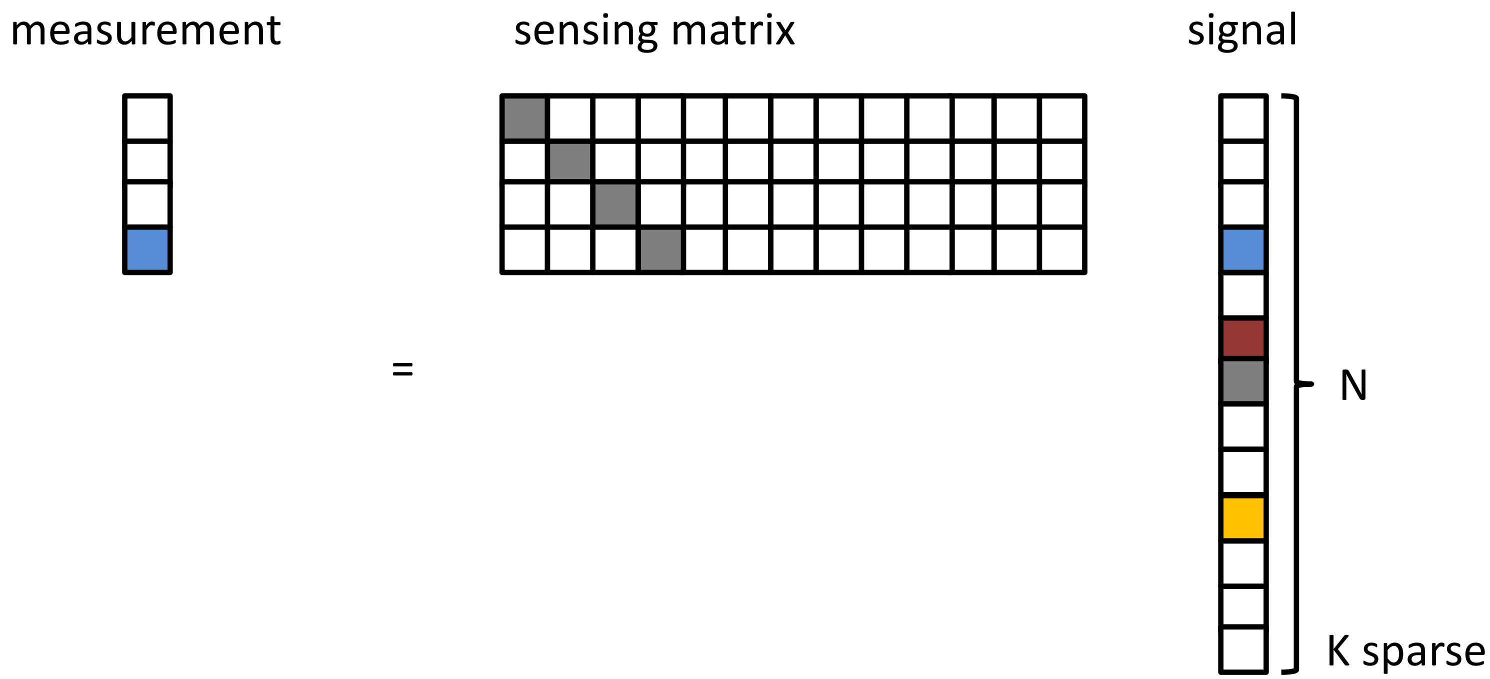
Standard approach



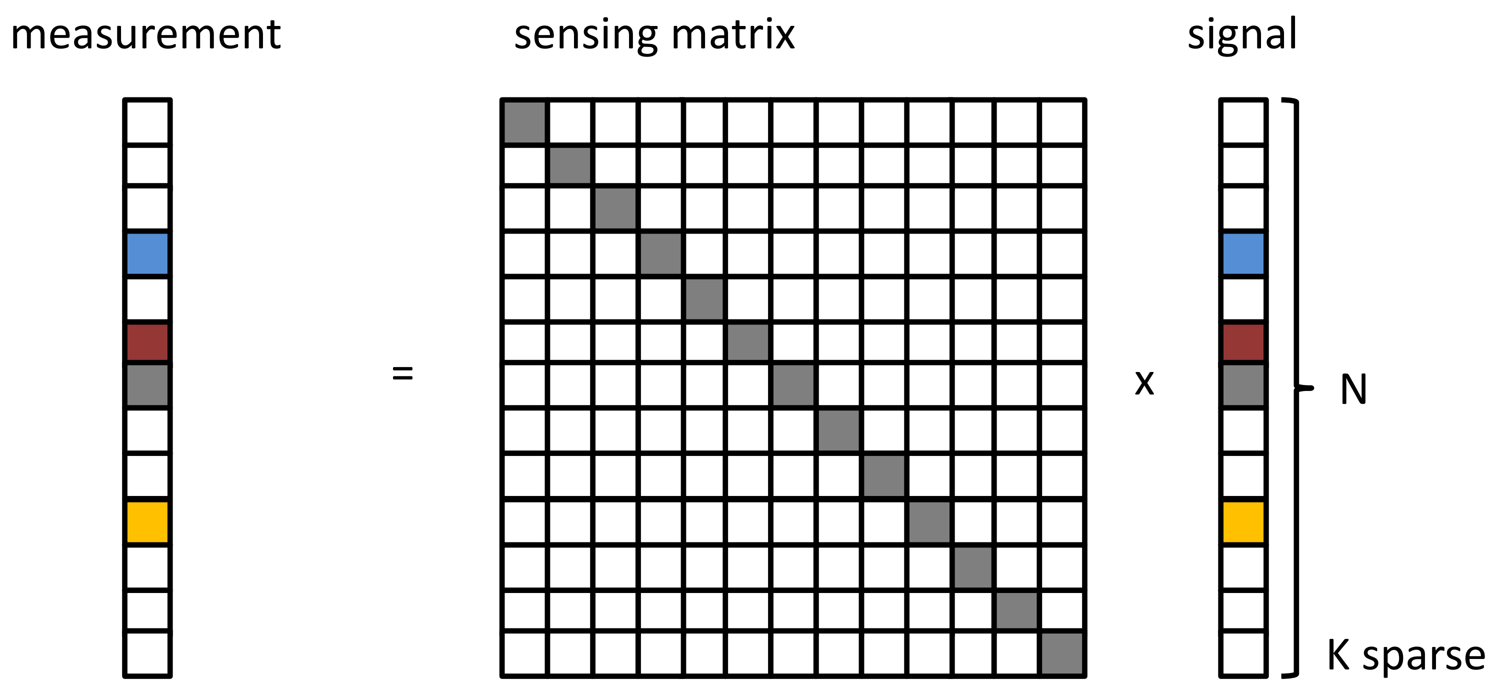
Standard approach



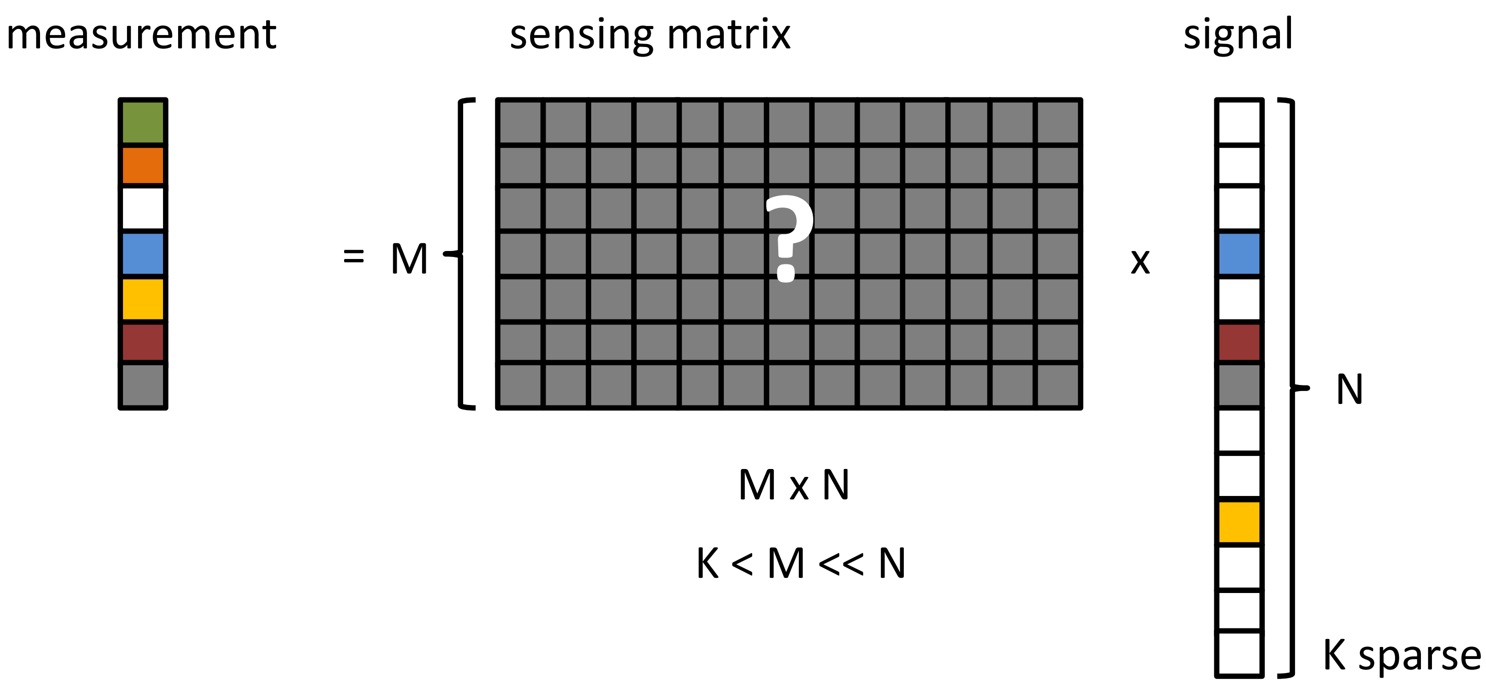
Standard approach



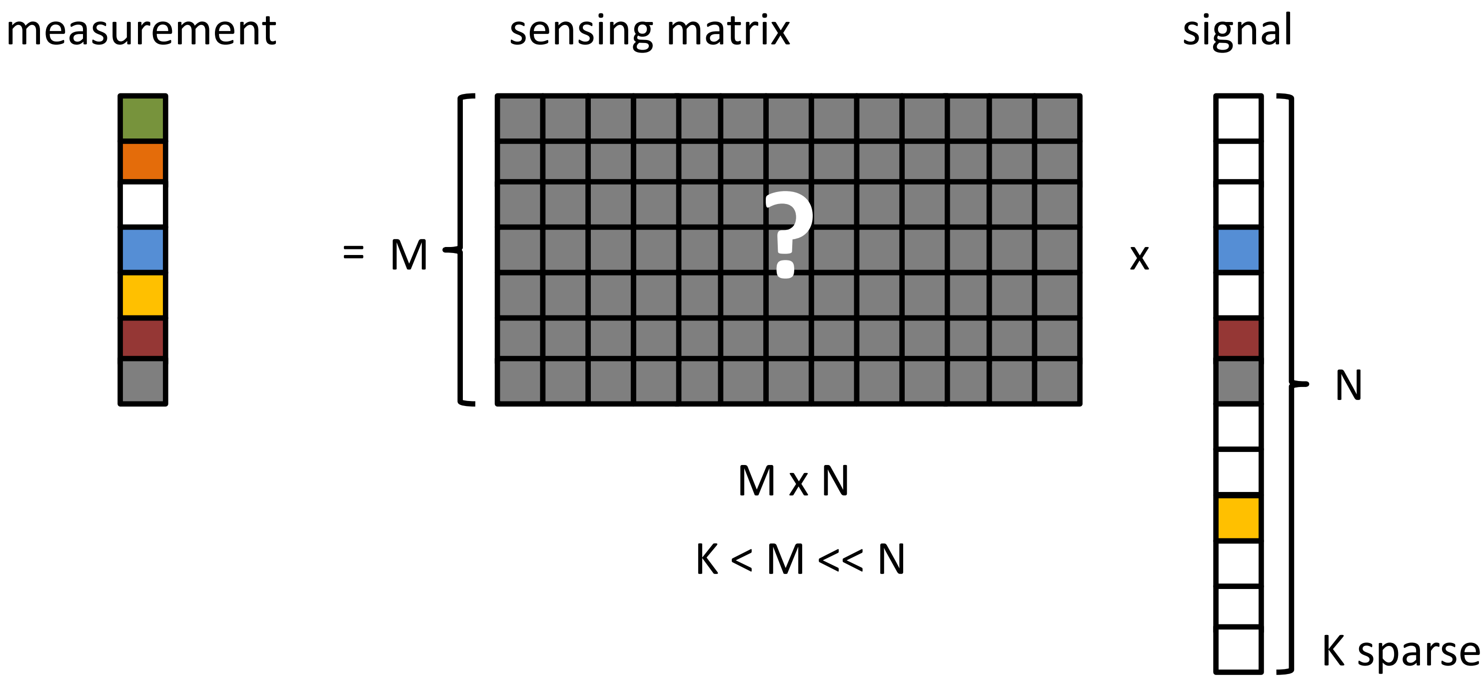
Standard approach



Compressed Sensing

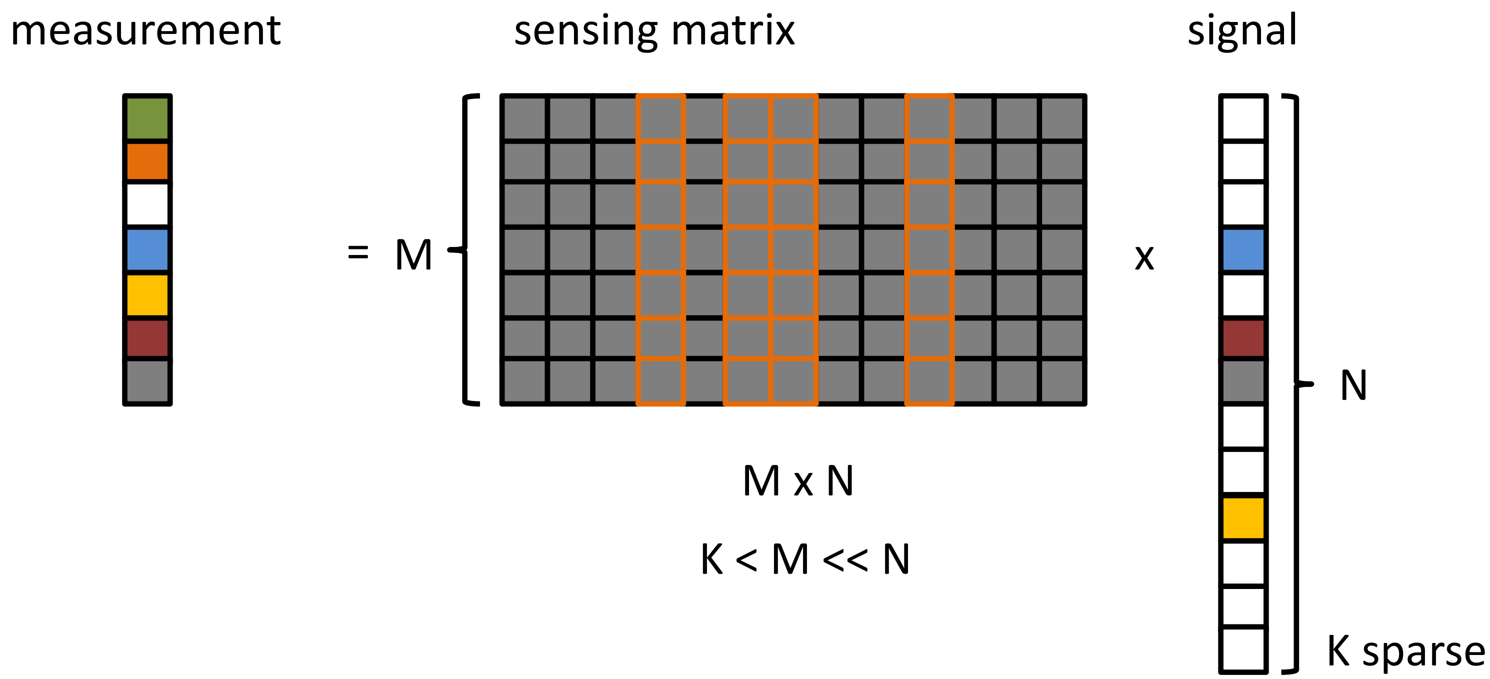


Compressed Sensing



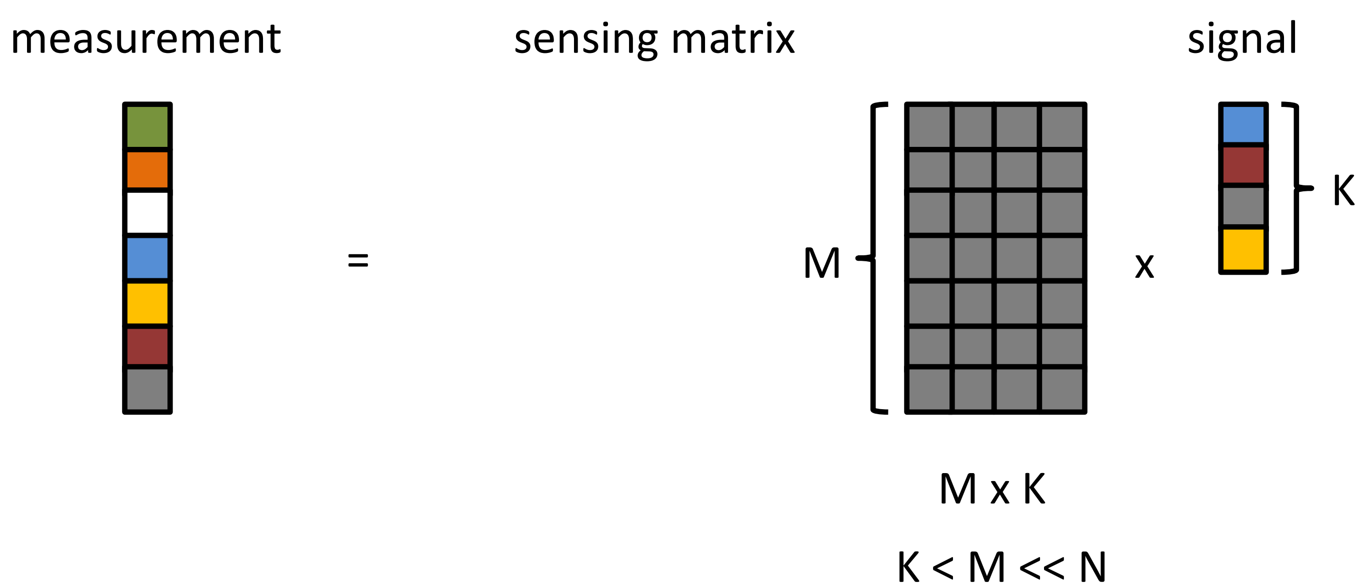
→ M < N: information loss (not full rank)

Compressed Sensing



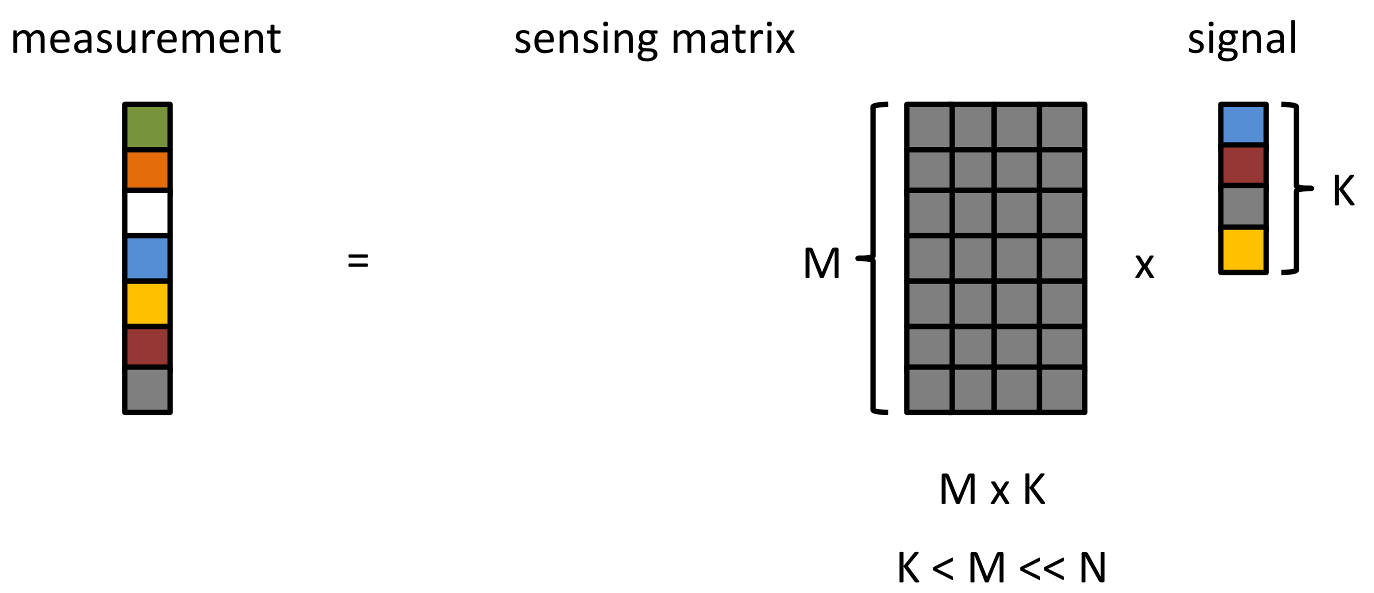
→ M < N: information loss (not full rank)

Compressed Sensing



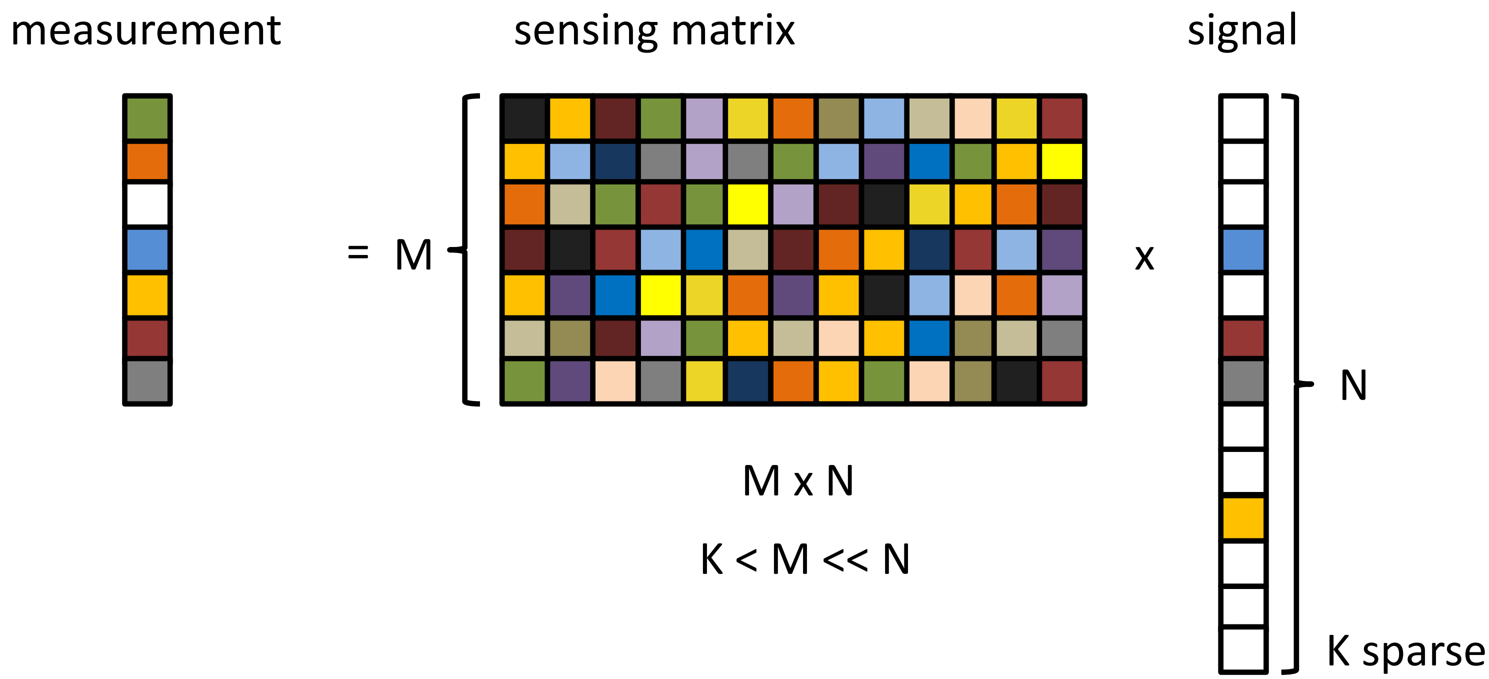
→ M > K: we preserve all information if M x K has full rank

Compressed Sensing



- M > K: we preserve all information if M x K has full rank
- **Design problem**: Find a M x N matrix such that all M x K submatrices have full rank and are almost orthogonal to each other
  Restricted Isometry Problem (RIP)

Using a random sampling matrix



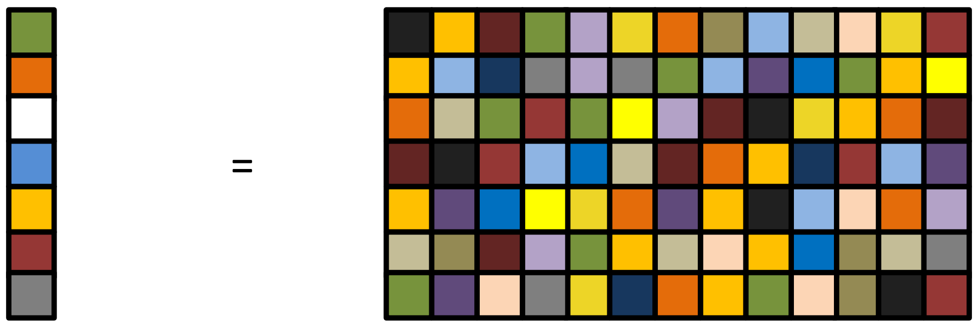
→ a randomized (iid gaussian) matrix has this property to a very high likelihood if:

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

[Kashin, Gluskin]

Recovering the input signal

measurement | sensing matrix | signal



= x ?

M x N

$$y = \Phi \times x$$

Recovery with regression

Find $\hat{x}$ that solves $\mathrm{y} = \Phi x$

Least squares: $\min\{\|\Phi\hat{x} - y\|_2\}$

Recovery with regression

Find $\hat{x}$ that solves $\mathrm{y} = \Phi x$

Least squares: $\min\{\|\Phi\hat{x} - y\|_2\}$

finds perfect $\hat{x}$, but not a sparse one

Ridge regression: $\min\{\|\Phi\hat{x} - y\|_2 + \lambda\|\hat{x}\|_2\}$

Recovery with regression

Find $\hat{x}$ that solves $\mathrm{y} = \Phi x$

Least squares: $\min\{\|\Phi\hat{x} - y\|_2\}$

finds perfect $\hat{x}$, but not a sparse one

Ridge regression: $\min\{\|\Phi\hat{x} - y\|_2 + \lambda\|\hat{x}\|_2\}$

finds small $\hat{x}$, but not a sparse one

Lasso regression: $\min\{\|\Phi\hat{x} - y\|_2 + \lambda\|\hat{x}\|_1\}$

[Candes, Donohan]

Recovery with regression

Example:

$\Phi\hat{x} = y$

Φ = random(200, 1000)

y = random(200)

[Steve Brunton]

Recovery with regression

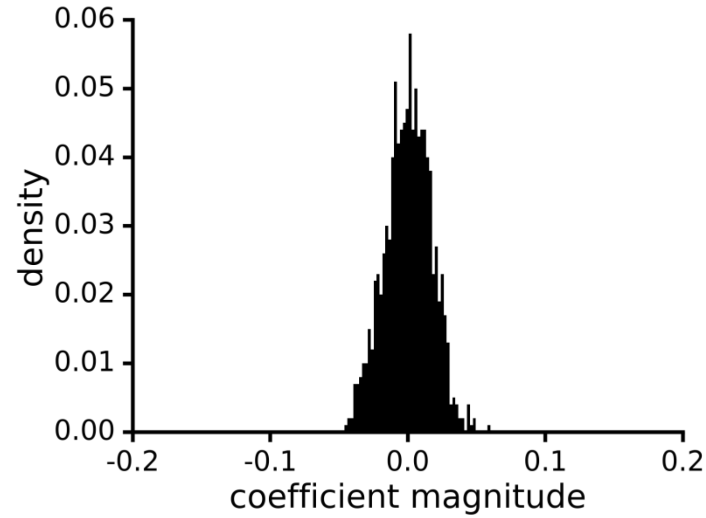
Example:

$\Phi\hat{x} = y$

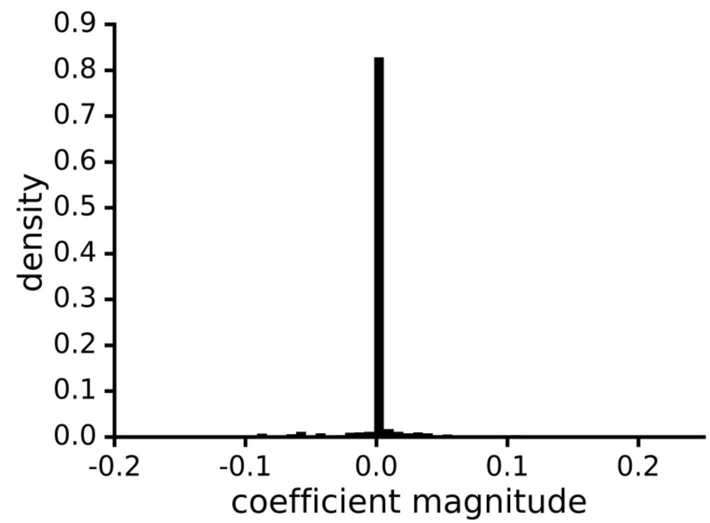
Φ = random(200, 1000)

y = random(200)

Ridge

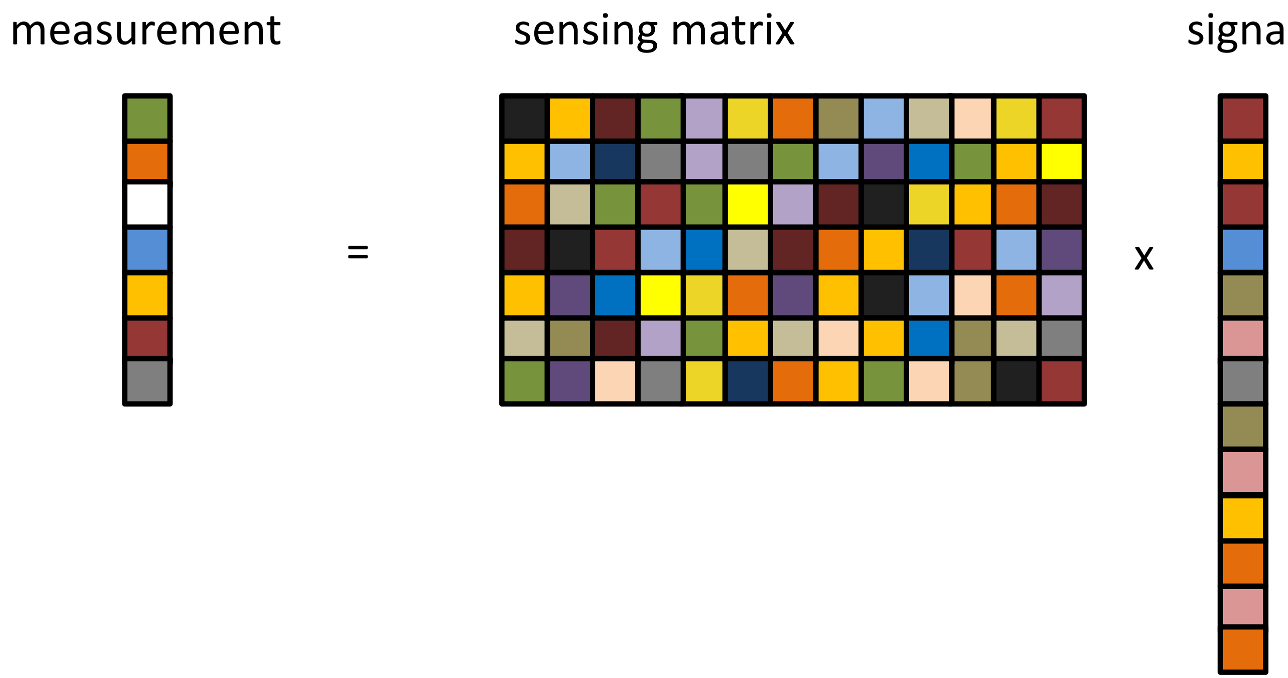


Lasso



[Steve Brunton]

Dealing with non-sparse signals



Dealing with non-sparse signals

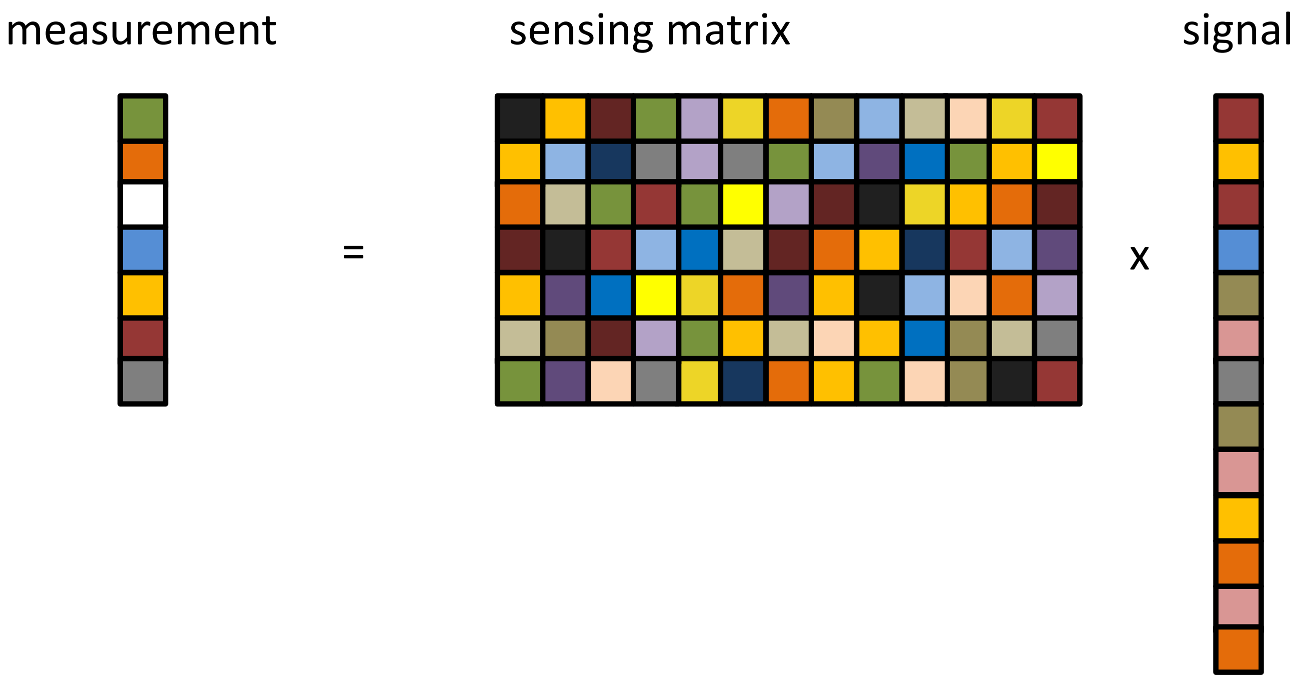
Let x' be the representation of x in a sparse basis Ψ

$$\Psi x' = x$$

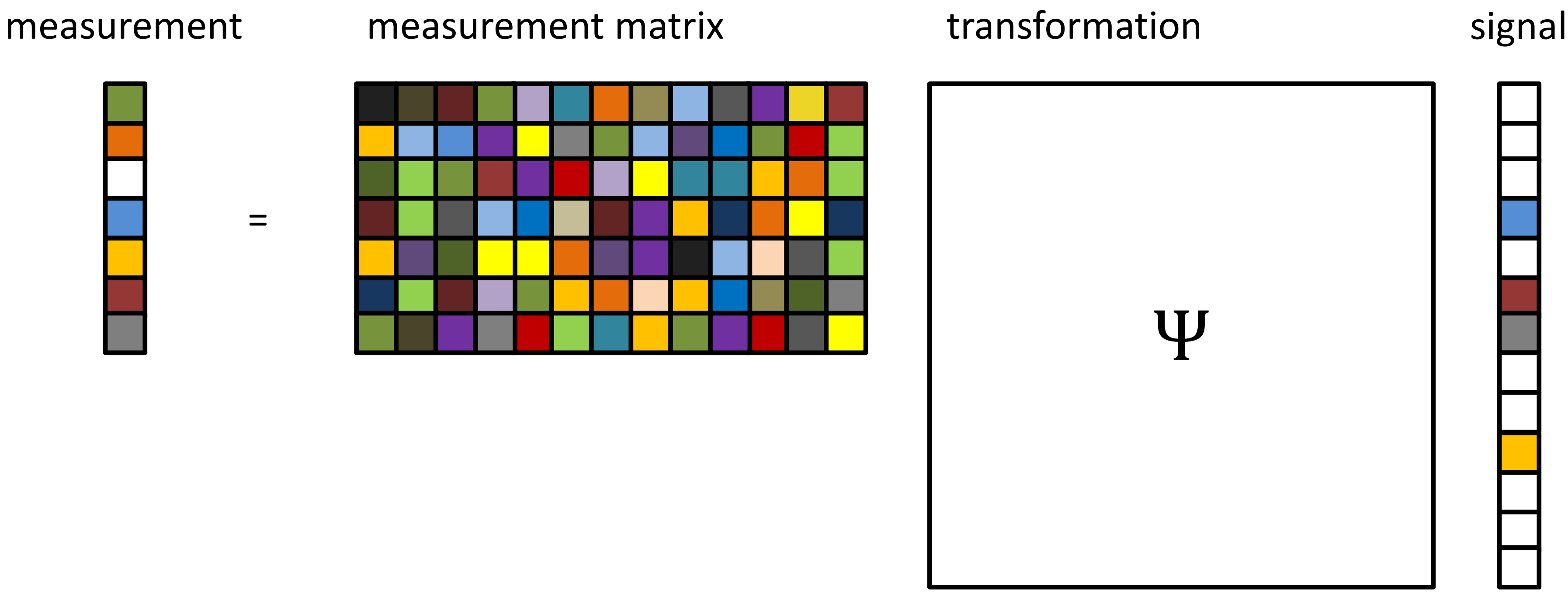
Same sampling process

$$y = \Phi \Psi x'$$

Dealing with non-sparse signals



Dealing with non-sparse signals



Dealing with non-sparse signals

Let x' be the representation of x in a sparse basis Ψ

$$\Psi x' = x$$

Same sampling process

$$y = \Phi\Psi x'$$

Update measurement matrix when recovering

$$\min\{\|\boxed{\Phi\Psi}\hat{x}' - y\|_2 + \lambda\|\hat{x}'\|_1\}$$

Dealing with non-sparse signals

Let x' be the representation of x in a sparse basis Ψ

$$\Psi x' = x$$

Same sampling process

$$y = \Phi\Psi x'$$

Update measurement matrix when recovering

$$\min\{\|\boxed{\Phi\Psi}\hat{x}' - y\|_2 + \lambda\|\hat{x}'\|_1\}$$

Transform back after recovering: $\hat{x} = \Psi\hat{x}'$

Dealing with non-sparse signals

Let x' be the representation of x in a sparse basis Ψ

$$\Psi x' = x$$

Same sampling process

$$y = \Phi\Psi x'$$

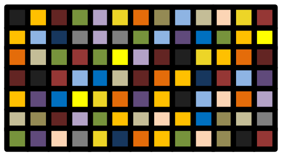
Update measurement matrix when recovering

$$\min\{\|\boxed{\Phi\Psi}\hat{x}' - y\|_2 + \lambda\|\hat{x}'\|_1\}$$

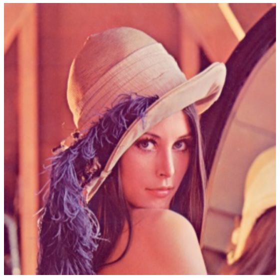
Transform back after recovering: $\hat{x} = \Psi\hat{x}'$

$\longrightarrow$ Ψ does not have to be known when sensing!

Wrap Up

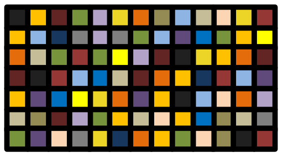


$$y = \Phi\Psi x'$$

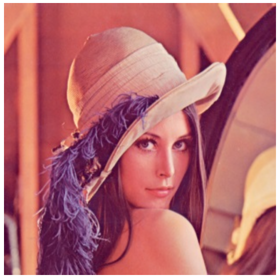


sense randomly | store | recover

Wrap Up



$$\boxed{y = \Phi\Psi x'}$$



sense randomly | store | recover

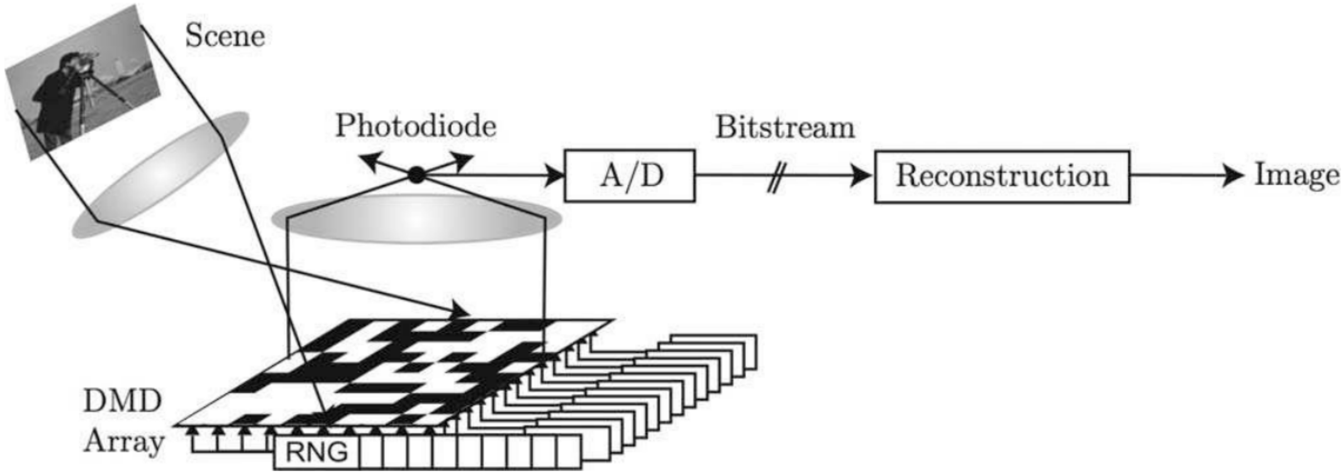
Sampling below Shannon frequency

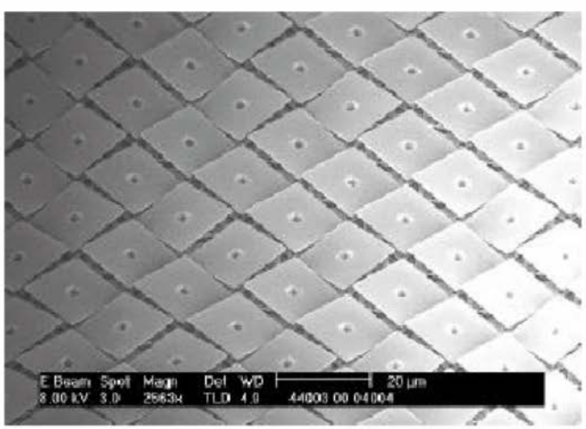
– faster

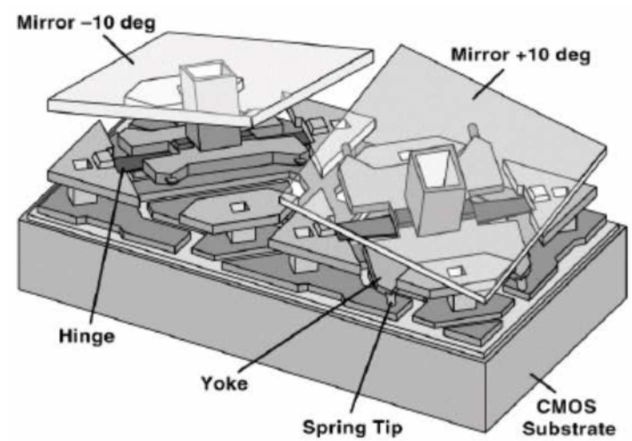
Asymmetric

– the decoder has to do most of the work

One pixel camera

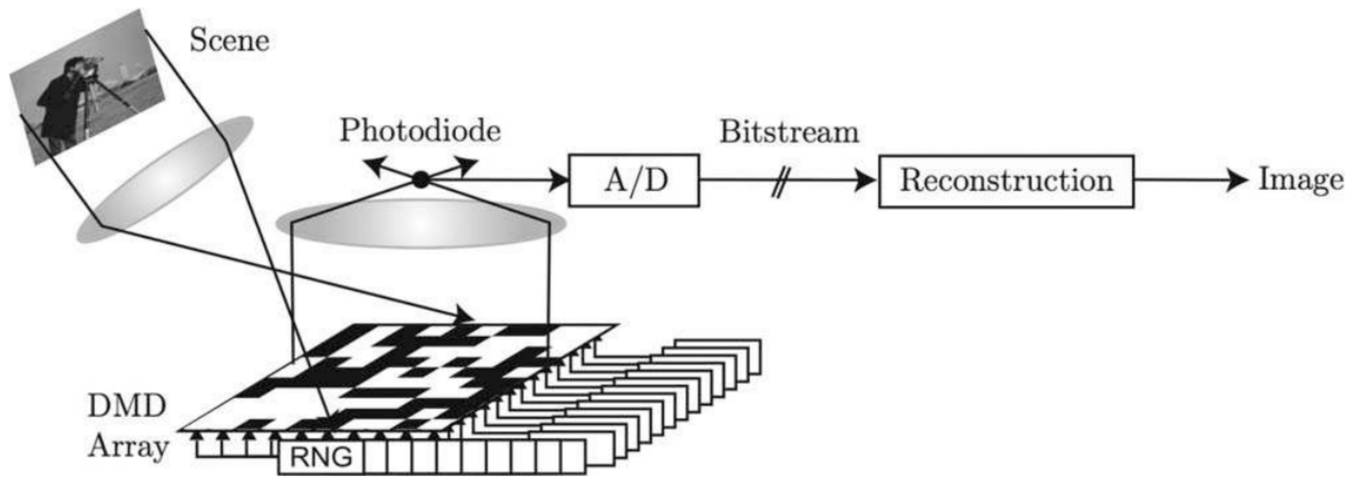


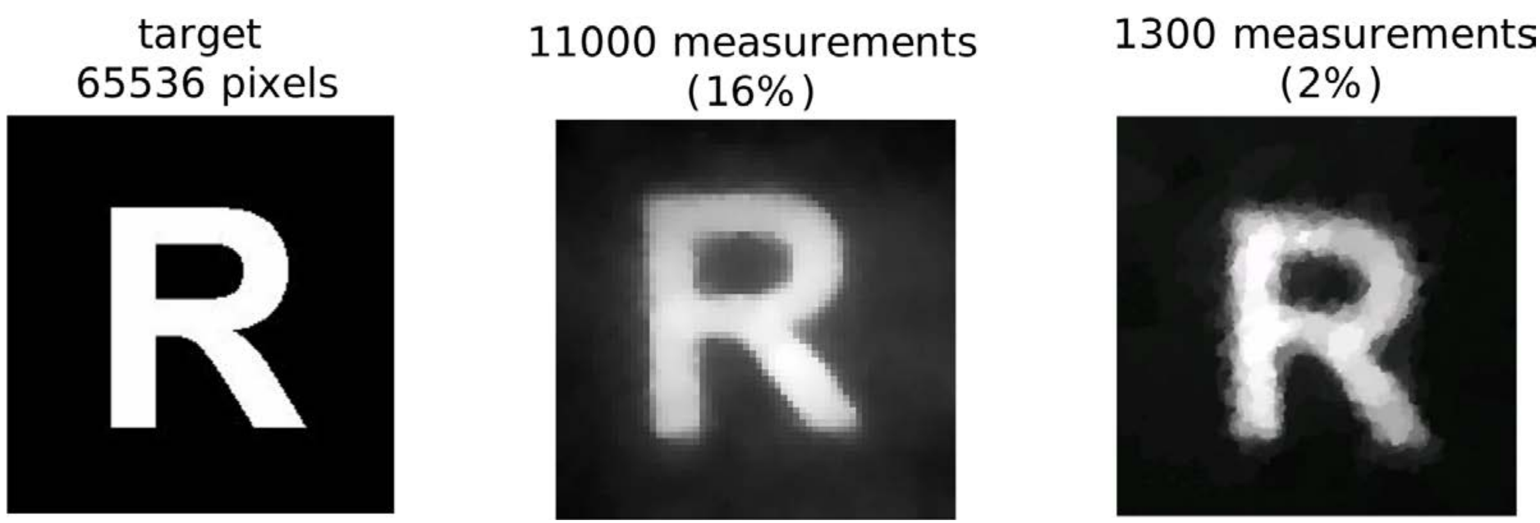




[Baranuik, Compressive Sensing Lecture Notes, 2007]

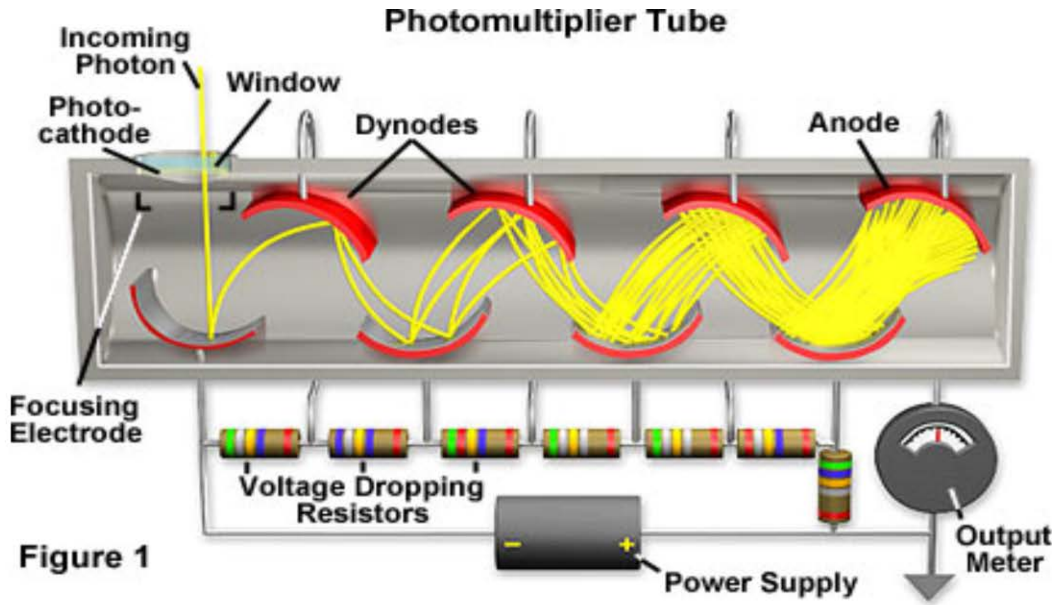
One pixel camera

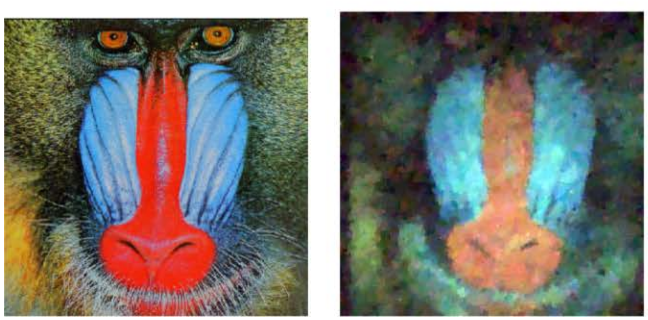




[Baranuik, Compressive Sensing Lecture Notes, 2007]

Low light one pixel camera





true color low-light imaging

256 x 256 image with 10:1 compression

[Baranuik, Compressive Sensing Lecture Notes, 2007]

Little experiment

10% highes wavelets coefficients

RMSE: 0.010

RMSE of random image: 0.423

Little experiment

10% highest wavelets coefficients

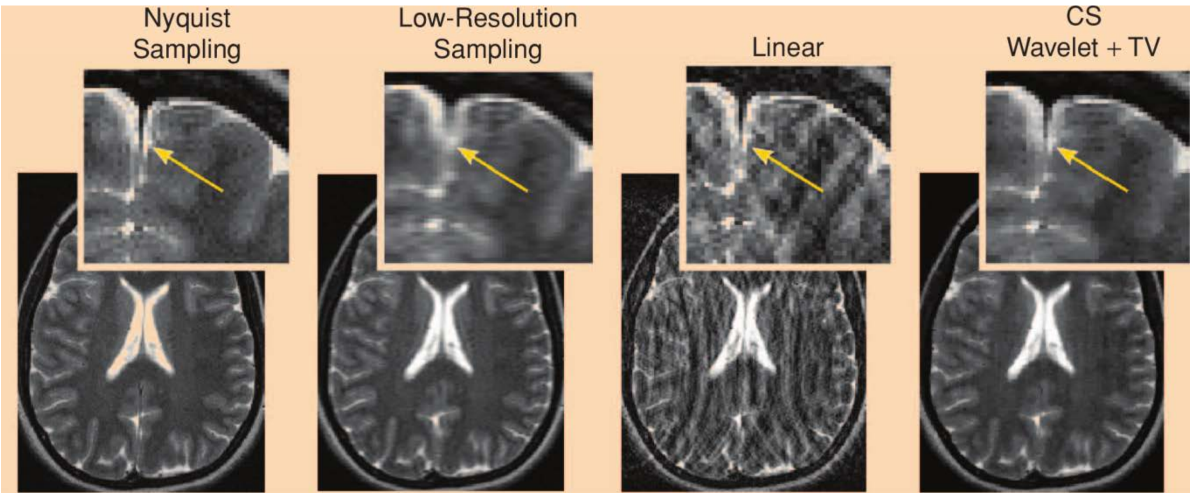
RMSE: 0.010

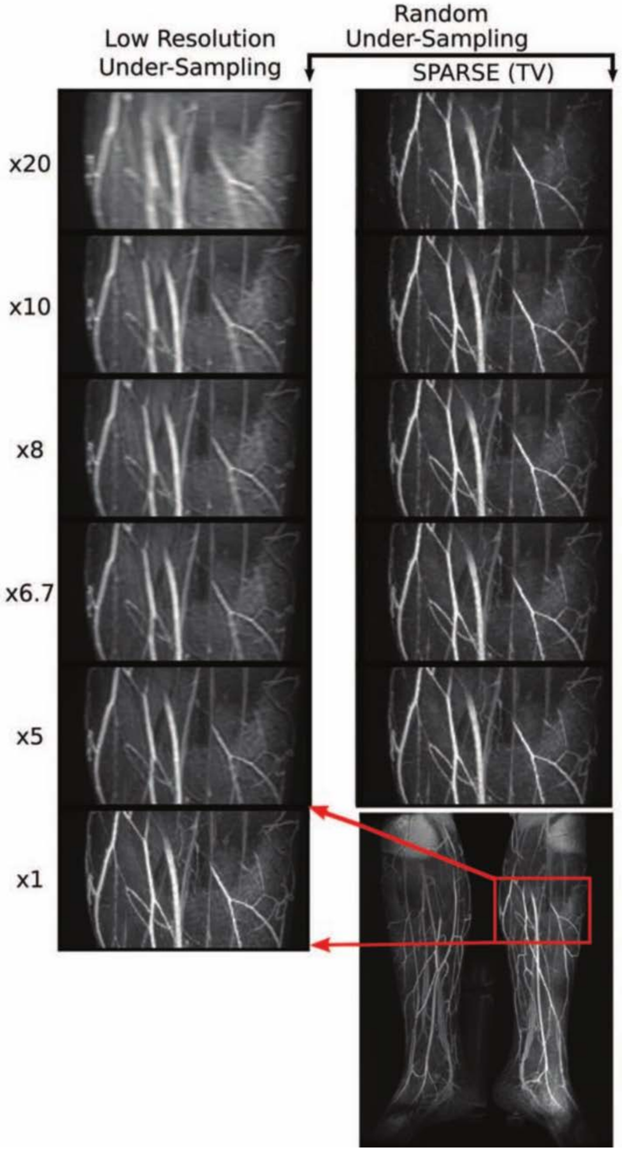
30% as many measurements

RMSE: 0.039

RMSE of random image: 0.423

Speeding up MRI acquisition





[Lustig et al., Compressed Sensing MRI, 2008]

Recovery of missing data

original image

Recovery of missing data

original image

corruption rate: 0.1

Recovery of missing data

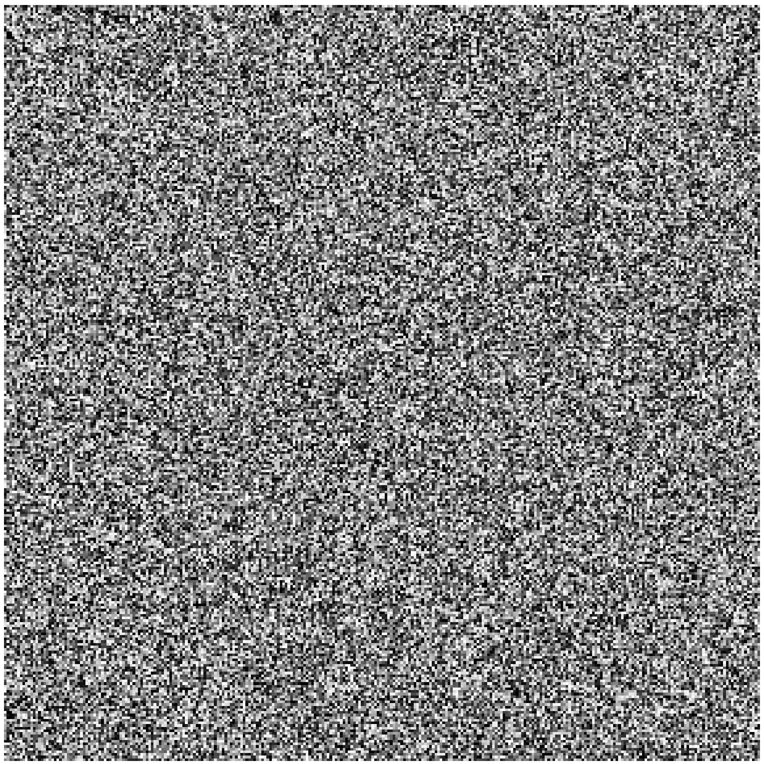
recovered image

RMSE: 0.00611

corruption rate: 0.1

Recovery of missing data

random image



RMSE: 0.423

corruption rate: 0.1

Recovery of missing data

recovered image

RMSE: 0.00611

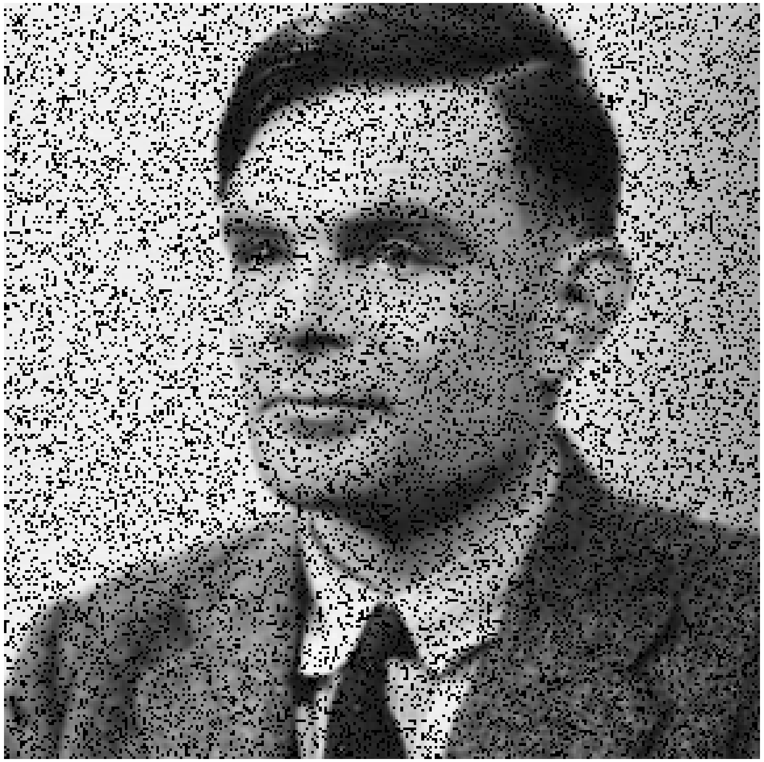
corruption rate: 0.1

Recovery of missing data

recovered image

RMSE: 0.00974

corruption rate: 0.2



Recovery of missing data

recovered image

RMSE: 0.0139

corruption rate: 0.3

Recovery of missing data

recovered image

RMSE: 0.0197

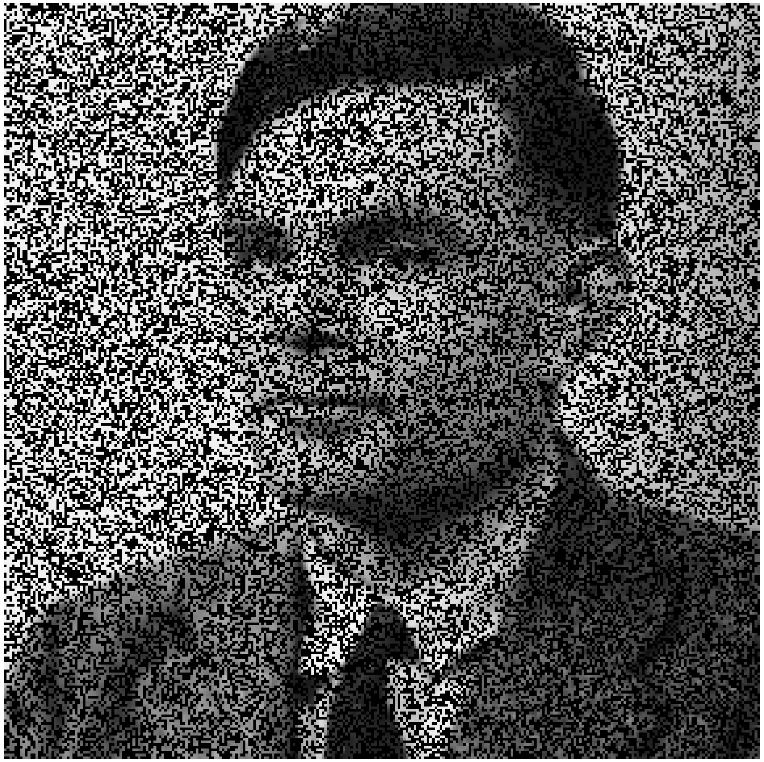
corruption rate: 0.4

Recovery of missing data

recovered image

RMSE: 0.0357

corruption rate: 0.5



Recovery of missing data

recovered image

RMSE: 0.0426

corruption rate: 0.6

Recovery of missing data

recovered image

RMSE: 0.063

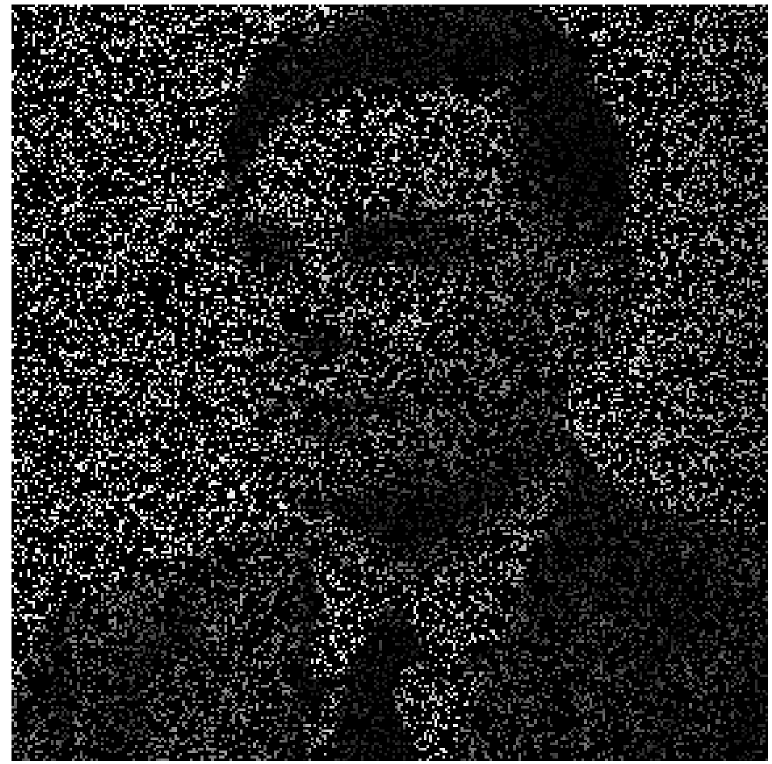
corruption rate: 0.7

Recovery of missing data

recovered image

RMSE: 0.0773

corruption rate: 0.8

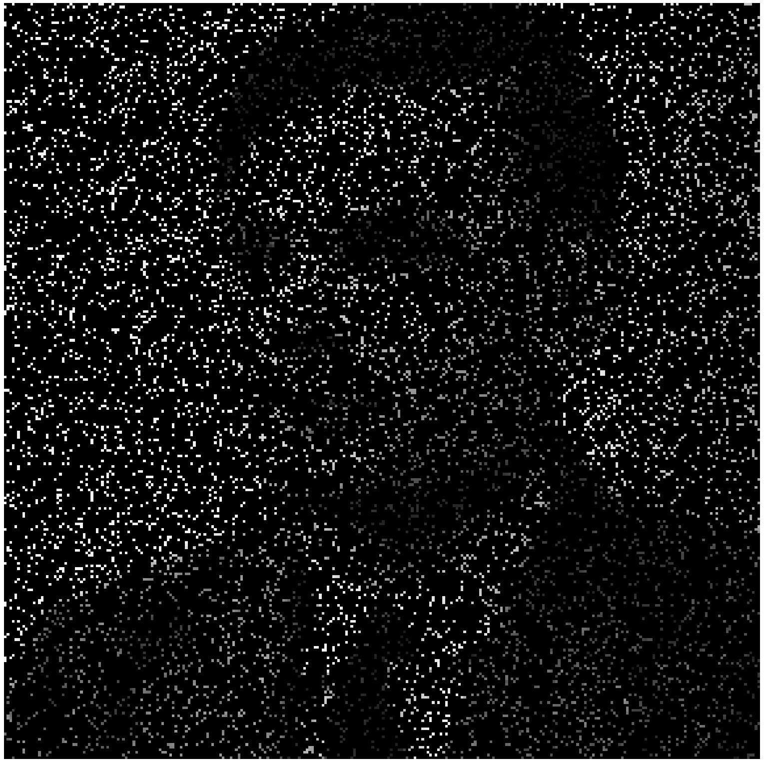


Recovery of missing data

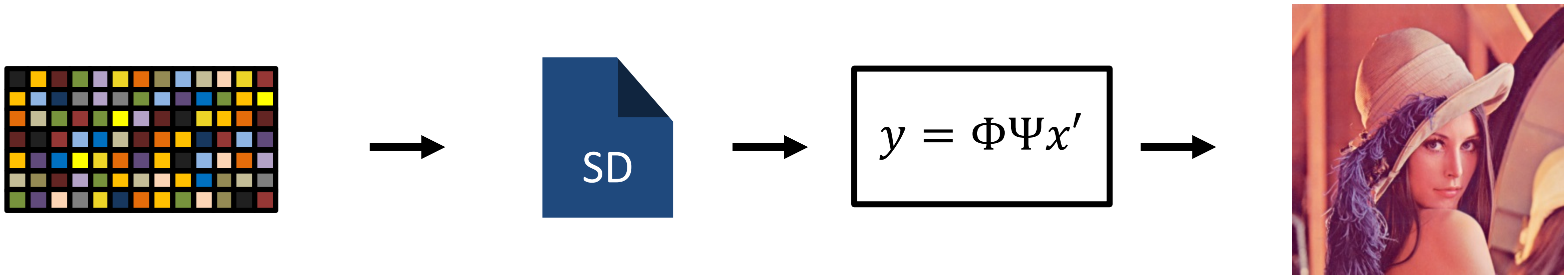
recovered image

RMSE: 0.116

corruption rate: 0.9



Conclusion



sense randomly　　store　　recover

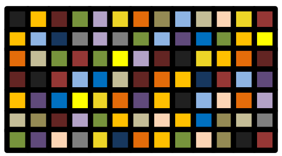
CS can beat the Nyquist Shannon limit

The computational burdon is shifted to the decrypter

Adjustments to the sensor have to be made

Future: Combination with classification

Conclusion

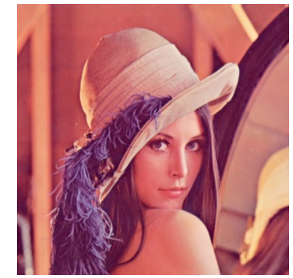


sense randomly

store

$$y = \Phi\Psi x'$$

recover



Code and suggested talks and reading:

https://github.com/sdorkenw/CompressedSensingSeminar