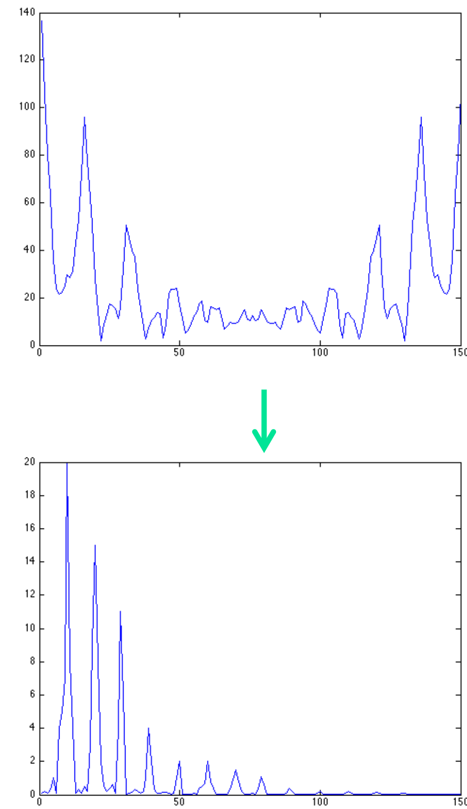
Recent Developments in the Sparse Fourier Transform

Piotr Indyk

MIT

Joint work with Fadel Adib, Badih Ghazi, Haitham Hassanieh, Michael Kapralov, Dina Katabi, Eric Price, Lixin Shi

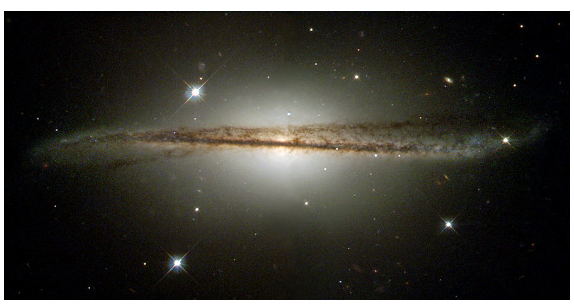
Fourier Transform



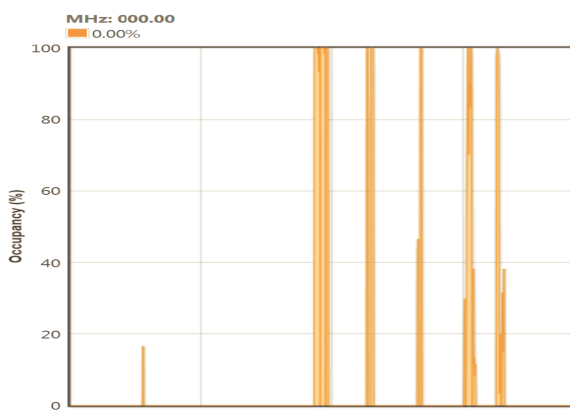
- Discrete Fourier Transform:
  - Given: a signal x[1…n]
  - Goal: compute the frequency vector x’ where

$$x'_f = \Sigma_t \, x_t \, e^{-2\pi i \, tf/n}$$

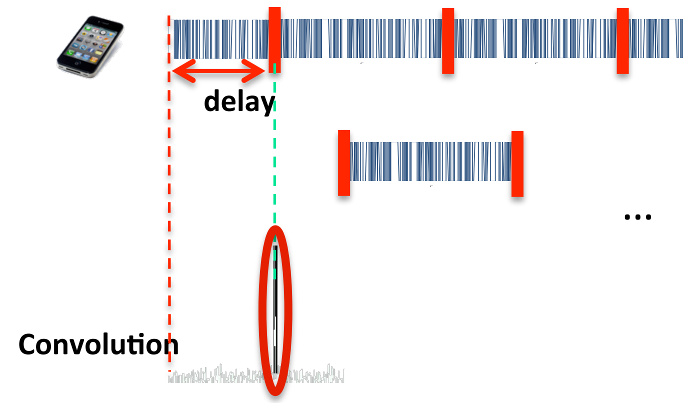
- Very useful tool:



Video / Audio

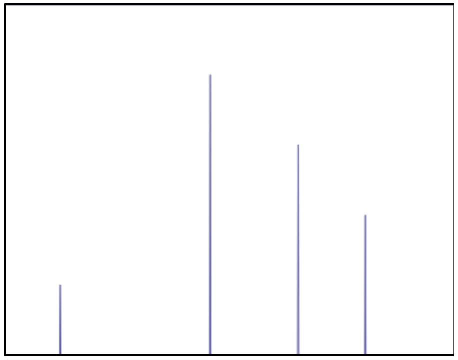


Communications



GPS

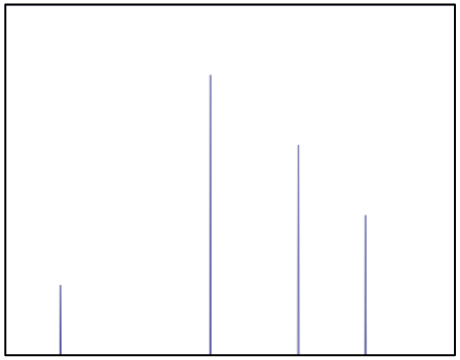
Known algorithms



- Fast Fourier Transform (FFT) computes the frequencies in time $O(n \log n)$
- But, we can do better if we only care about small number k of "dominant frequencies"
  - E.g., recover assuming it is k-sparse (only k non-zero entries)
- Plenty of algorithms known:
  - Boolean cube (Hadamard Transform): [KM'91] (cf. [GL])
  - Complex FT: [Mansour'92, GGIMS'02, AGS'03, GMS'05, Iwen'10, Akavia'10]
- Best running time: $k \log^c n$ for some $c=O(1)$ [GMS05]
  - Improve over FFT for $n/k >> \log^{c-1} n$

*Assuming entries of x are integers with $O(\log n)$ bits of precision.

Challenges



- Run-time: : $k \log^c n$ [GMS05]
- Problem:
  - $c \approx 4$
  - Need $k < 100$ to beat FFTW for n=4,000,000
- Goal:
  - Theory: improve over FFT for all values of $k=o(n)$
  - Improve in practice

Guarantees

- All algorithms randomized, with constant probability of success, n is a power of 2
- Approximation guarantees:
  - Exactly k-sparse case: report exact answer*
  - Approximately k-sparse case: report y' that satisfies the l_2/l_2 guarantee:

$$||x'-y'||_2 \leq C \min_{k\text{-sparse } z'} ||x'-z'||_2$$

  - Approximately k-sparse case: l_∞/l_2 guarantee:

$$||x'-y'||_\infty \leq C \min_{k\text{-sparse } z'} ||x'-z'||_2 / k^{1/2}$$

*Assuming entries of x are integers with $O(\log n)$ bits of precision.

Results

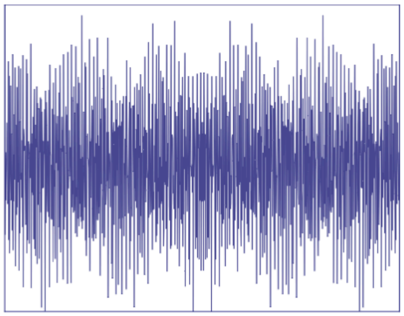
	Time	Guarantee	Comments	Samples
HIKP'12	$(nk)^{1/2} \log^{3/2} n$	l_∞/l_2	Faster than FFTW if k<2000 (n=4,000,000)	$(nk)^{1/2} \log^{3/2} n$
HIKP'12b	k log n	Exact	Faster than FFTW if k<100,000*	k log n
	k log n log(n/k)	l_2/l_2		k log n log(n/k)
GHIKPS'13	k log n	Exact	Average case, $k<n^{1/2}$	k
	$k \log^2 n$	l_2/l_2	Average case, $k<n^{1/2}$	k log n
IKP'14	$k \log^2 n$	l_2/l_2		$k \log n * \log^c \log n$

*Further efficiency improvement by 2-5x was achieved by Pueschel-Schumacher'13

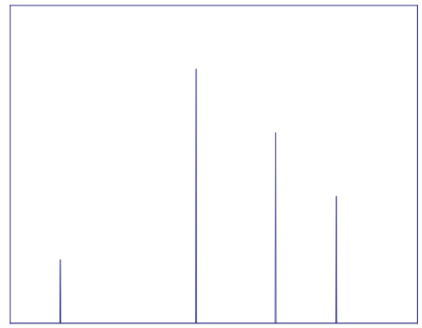
Exact Sparsity

	Time	Guarantee	Comments
HIKP’12b	k log n	Exact	Faster than FFTW if k<100,000

First attempt



Time Domain Signal

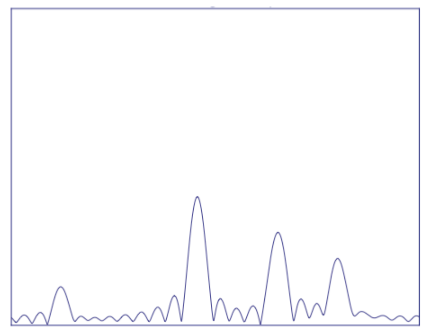


Frequency Domain

n-point DFT : $n\log(n)$

$\mathbf{x} \rightarrow \mathbf{x}'$

Cut off Time signal

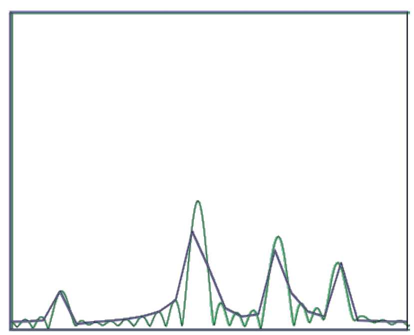


Frequency Domain

n-point DFT of first B terms : $n\log(n)$

$\mathbf{x}\times$ **Boxcar** $\rightarrow$ $\mathbf{x}' *$ **sinc**

First B samples



Frequency Domain

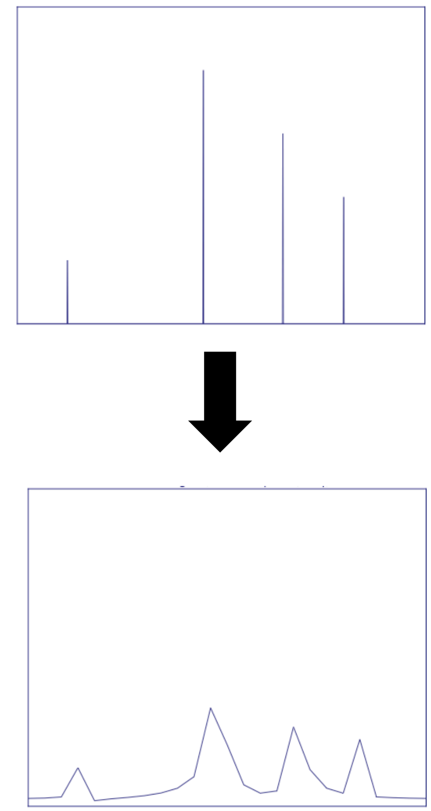
B-point DFT of first B terms: $B\log(B)$

Alias $(\mathbf{x}\times\ Boxcar)$

$\downarrow$

Subsample $(\mathbf{x}' * sinc)$

Issues



- Issues:
  - Two non-zero coefficients can be very close

    Can permute the spectrum pseudo-randomly by permuting the signal [GGIMS’02,GMS’05]
  - Leakage

    Replace Boxcar filter by a nicer function
  - Finding the support

    Recover the index from the phase
    - “OFDM trick”
    - Matrix Pencil, Prony method [Chiu-Demanet’13, Potts-Kunis-Heider-Veit’13]

Close non-zero coefficients

Pseudo-random Spectrum Permutation

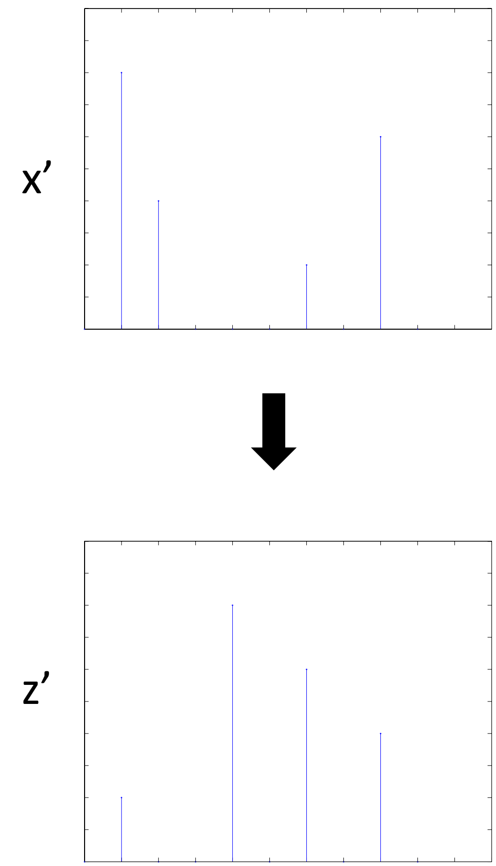
- Permute time domain signal → permute frequency domain
- Let

$$z_t = x_{\sigma t}\, e^{-2\pi i\, t\, \beta/n}$$

- If σ is invertible mod n

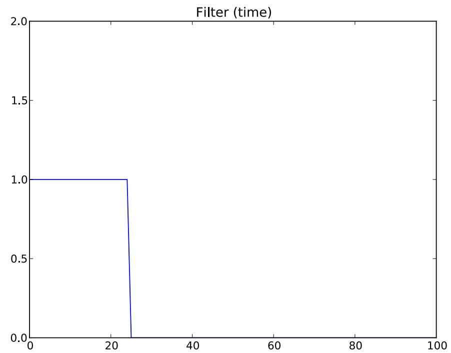
$$z'_f = x'_{1/\sigma\, f+\beta}$$

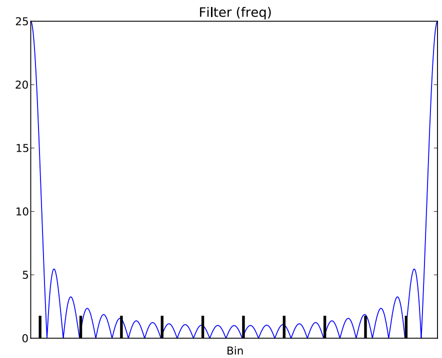
  - If n is a power of 2, any odd σ is OK
- Pseudo-random permutation: select
  - β uniformly at random from {0…n-1}
  - σ uniformly at random from odd numbers in {0…n-1}
- Each access to a coordinate of z_t can be simulated by accessing $x_{\sigma t}$ and multiplication



Reducing leakage

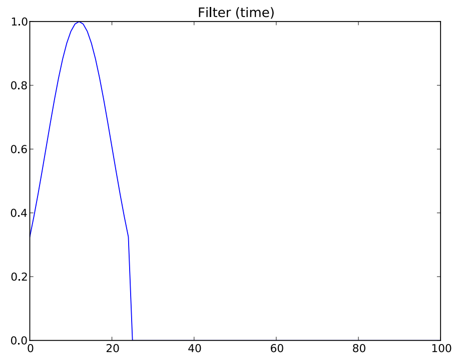
Filters: boxcar filter (used in[GGIMS02,GMS05])

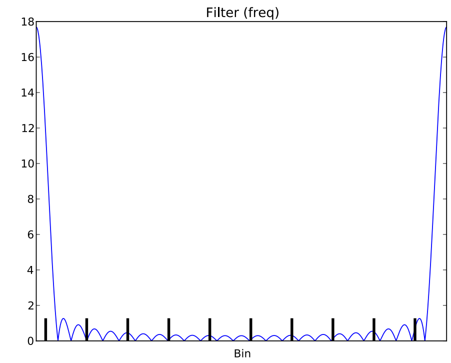




- Boxcar -> Sinc
  - Polynomial decay
  - Leaking to many buckets

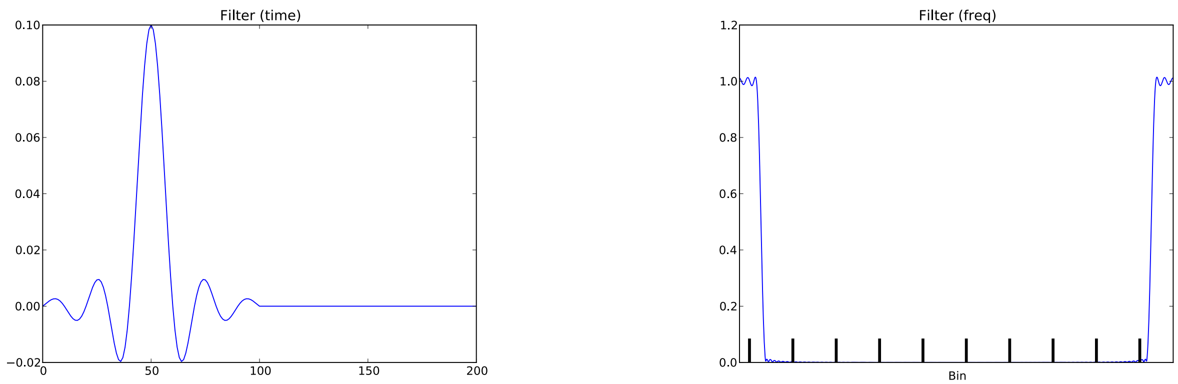
Filters: Gaussian





- Gaussian -> Gaussian
  - Exponential decay
  - Leaking to $(\log n)^{1/2}$ buckets

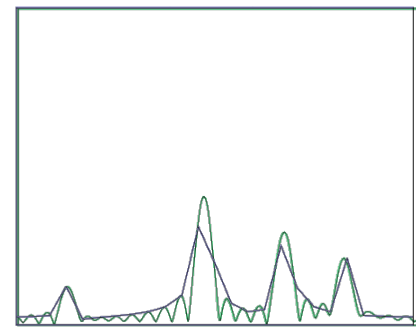
Filters: Sinc × Gaussian



- Sinc × Gaussian -> Boxcar*Gaussian
  - Still exponential decay
  - Leaking to <1 buckets
  - Sufficient contribution to the correct bucket
- Actually we use Dolph-Chebyshev filters

Finding the support

Finding the support



- $y' =$ B-point DFT $(x \times F)$

  $= \text{Subsample}(x' * F')$

- Assume no collisions:
  - At most one large frequency hashes into each bucket.
  - Large frequency f_1 hashes to bucket b_1

    $y'_{b1} = x'_{f1} F'_{\Delta}$ + ~~leakage~~

- Let x^{τ} be the signal time-shifted by τ, i.e. $x^{\tau}_{t} = x_{t-\tau}$
- Recall $\text{DFT}(x^{\tau})_f = x'_f \, e^{-2\pi i \, \tau \, f/n}$
- $y^{\tau}{}' =$ B-point DFT $(x^{\tau} \times F)$

  $y^{\tau}{}'_{b1} = x'_{f1} e^{-2\pi i \, \tau \, f1/n} \, F'_{\Delta}$ + ~~leakage~~

Finding the support, ctd

- At most one non-zero frequency f_1 per bucket b_1
- We have

$$y'_{b1}=x'_{f1}F'_{\Delta}$$

and

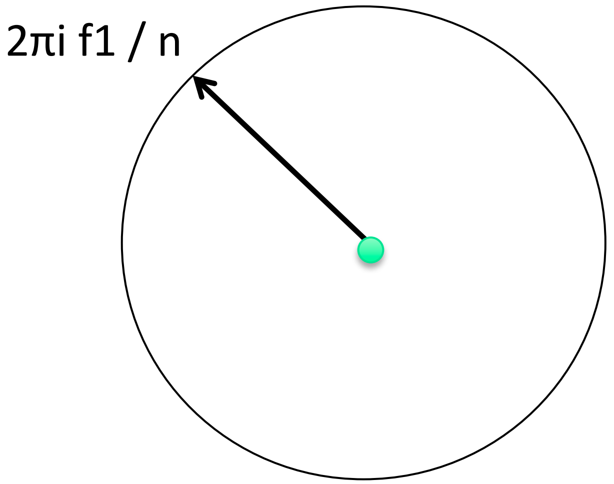
$$y'^{\tau}_{b1}=x'_{f1}e^{-2\pi i\,\tau\, f1/n}\,F'_{\Delta}$$

- So, for $\tau=1$ we have

$$y'_{b1}/y'^{1}_{b1} = e^{-2\pi i\, f1\,/n}$$

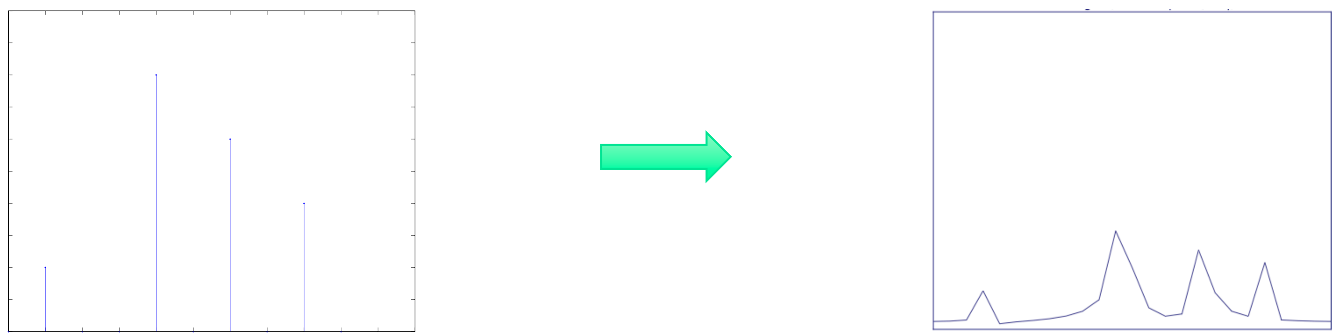
- Can get f1 from the phase
- Digression:
  - Cannot do this when the noise too large (approximately k-sparse case)
  - Instead, read bit by bit, multiply the runtime and sample complexity

  by log(n/k)



Putting it together

- We made this:



… act like this:

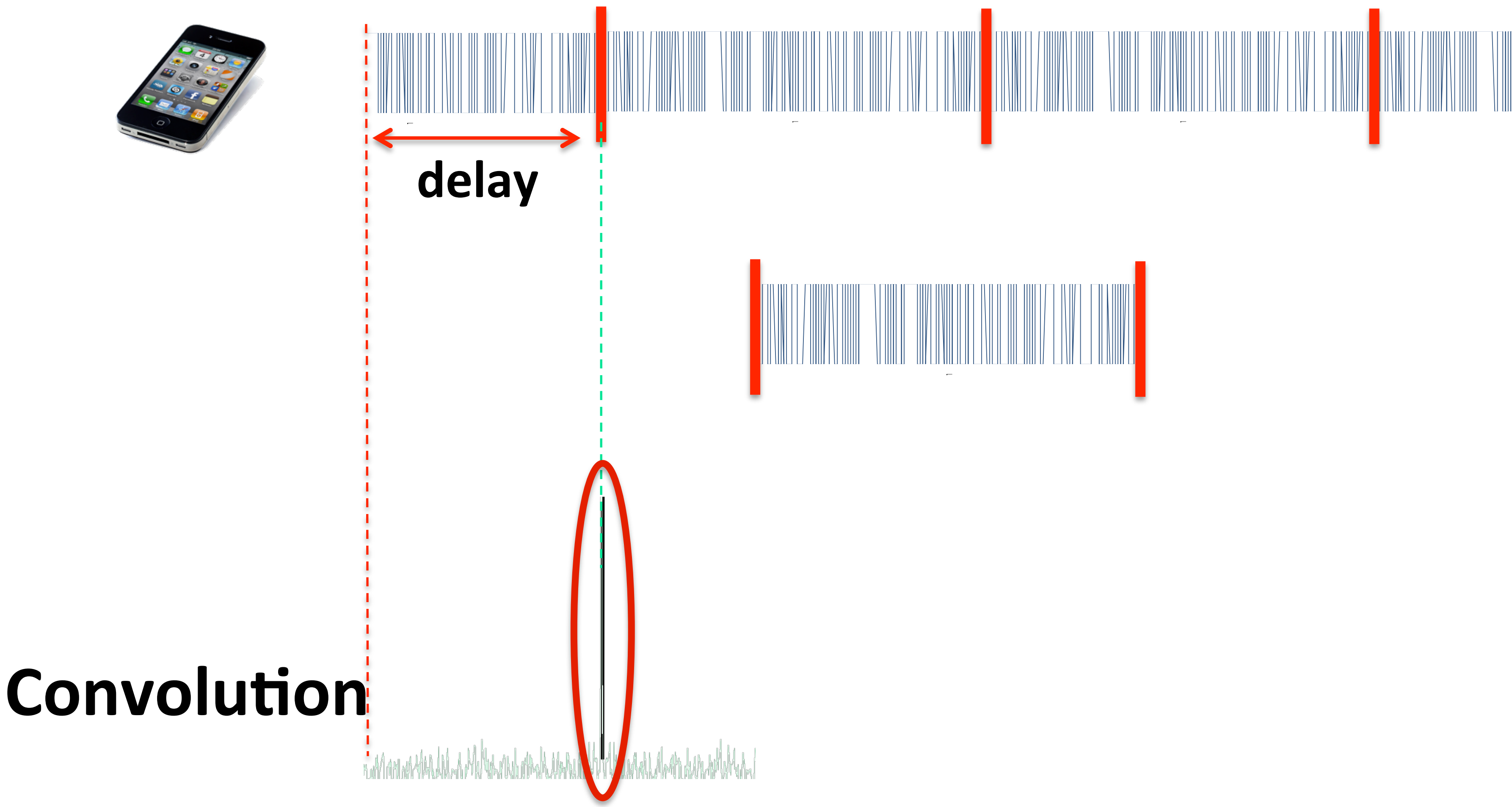
Now we can apply hashing-like compressive sensing methods
(using sparse matrices)

Applications

Applications

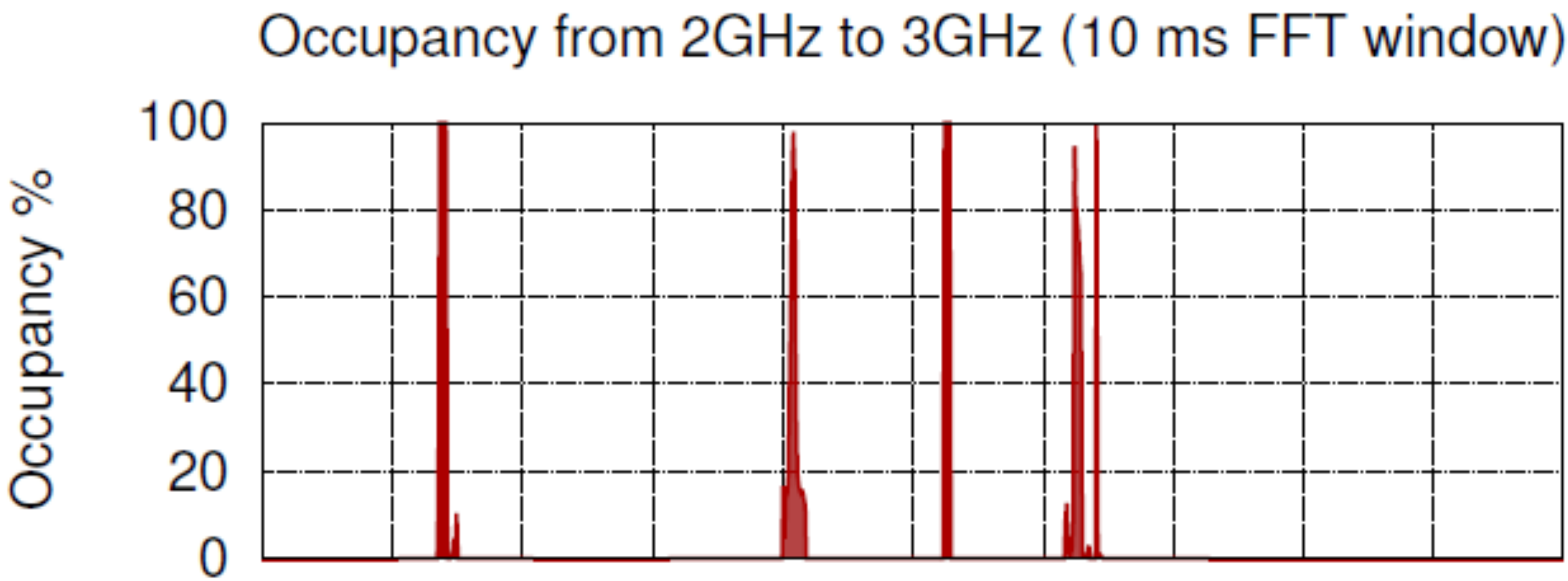
- GPS synchronization [Hassanieh-Adib-Katabi-Indyk, MOBICOM'12]
- Spectrum sensing [Hassanieh-Shi-Abari-Hamed-Katabi, INFOCOM'14]
- Magnetic Resonance Spectroscopy [Shi-Andronesi-Hassanieh-Ghazi-Katabi-Adalsteinsson' ISMRM'13]
- Exploiting Sparseness in Speech for Fast Acoustic Feature Extraction [Nirjon-Dickerson- Stankovic-Shen-Jiang, Workshop on Mobile Computing Systems and Applications'13]
- …

GPS locking (simplified)



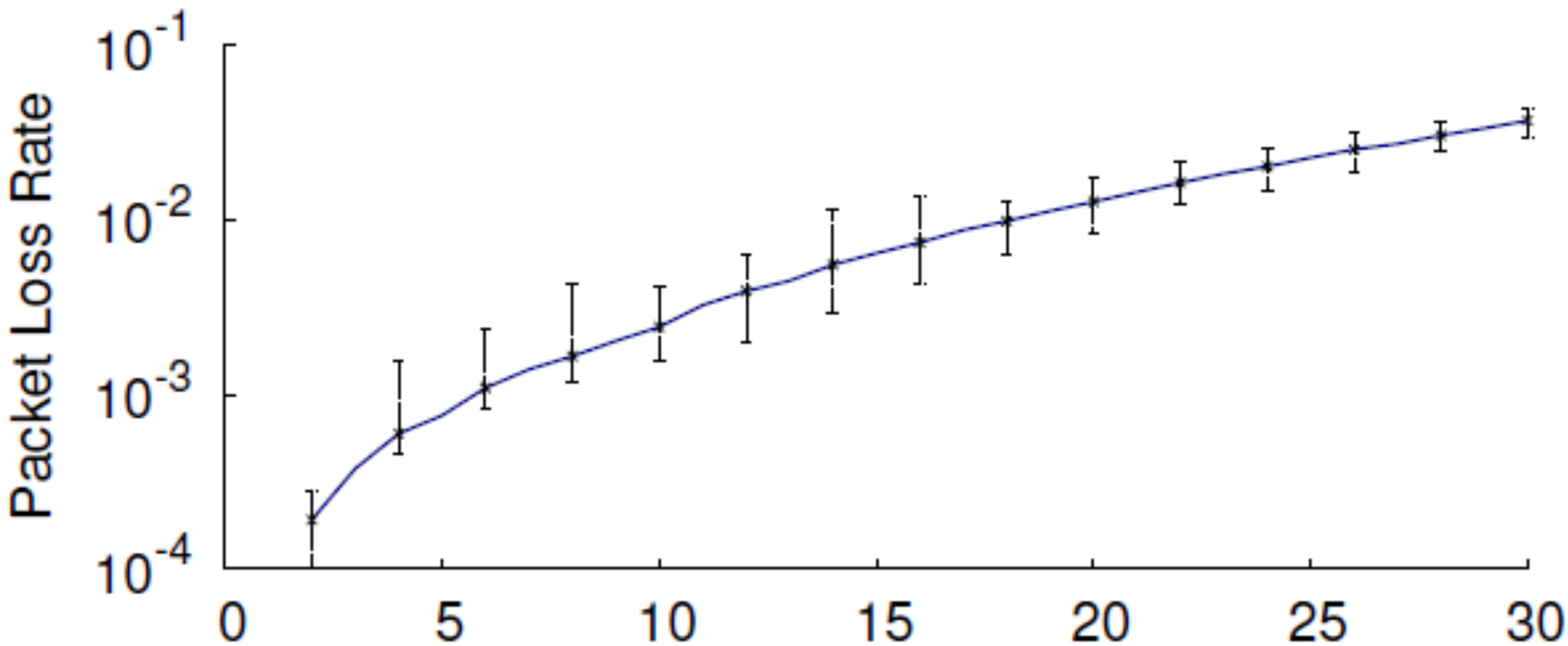
Realtime GHz Spectrum Sensing

Cambridge, MA January 15 2013



3 ADCs with a combined digital Bandwidth of 150 MHz can acquire a GHz

Decoding Senders Randomly Hopping in a GHz



sFFT enables realtime GHz sensing and decoding for low-power portable devices

Conclusions

- O(k log n) times achievable for the k-sparse case
- O(k log n log(n/k)) achievable for the L2/L2 guarantee
- Better sample bounds
- Questions:
  - Higher dimensions

    (recent work with M. Kapralov extends the results to any fixed dimension)
  - Uniform (a la compressive sensing)
  - Model-based

    (see also my talk at 1:30 pm on "Approximation-Tolerant Model-Based) Compressive Sensing", with C. Hegde and L. Schmidt)