15-826: Multimedia Databases and Data Mining

Lecture #24: DSP tools –

Fourier and Wavelets

C. Faloutsos

Must-read Material

- DFT/DCT: In PTVF ch. 12.1, 12.3, 12.4; in Textbook Appendix B.
- Wavelets: In PTVF ch. 13.10; in MM Textbook Appendix C

Outline

Goal: 'Find similar / interesting things'

- Intro to DB
- Indexing - similarity search
- Data Mining

Indexing - Detailed outline

- primary key indexing
- ..
- Multimedia –
  - Digital Signal Processing (DSP) tools
    - Discrete Fourier Transform (DFT)
    - Discrete Wavelet Transform (DWT)

CMU SCS

DSP - Detailed outline

- DFT
  - what
  - why
  - how
  - Arithmetic examples
  - properties / observations
  - DCT
  - 2-d DFT
  - Fast Fourier Transform (FFT)

15-826 Copyright: C. Faloutsos (2017) 5

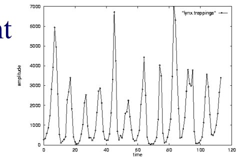
CMU SCS

Introduction

Goal: given a signal (eg., sales over time and/ or space)

Find: patterns and/or compress

count



lynx caught per year

year

15-826 Copyright: C. Faloutsos (2017) 6

CMU SCS

What does DFT do?

A: highlights the periodicities

15-826 Copyright: C. Faloutsos (2017) 7

CMU SCS

Why should we care?

A: several real sequences are periodic

Q: Such as?

15-826 Copyright: C. Faloutsos (2017) 8

CMU SCS

Why should we care?

A: several real sequences are periodic

Q: Such as?

A:

- sales patterns follow seasons;
- economy follows 50-year cycle
- temperature follows daily and yearly cycles

Many real signals follow (multiple) cycles

CMU SCS

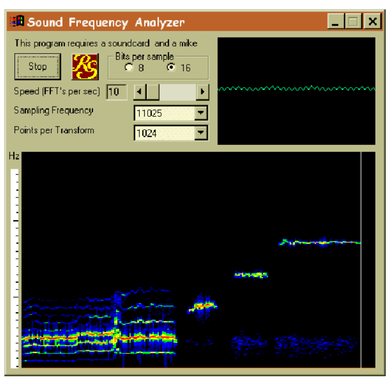
Why should we care?

For example: human voice!

- Frequency analyzer http://www.relisoft.com/freeware/freq.html
- speaker identification
- impulses/noise -> flat spectrum
- high pitch -> high frequency

CMU SCS

'Frequency Analyzer'

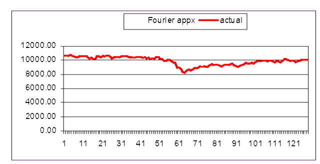


frequency

time

CMU SCS

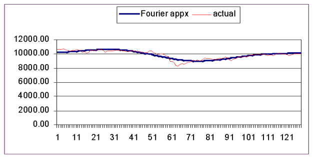
DFT and stocks



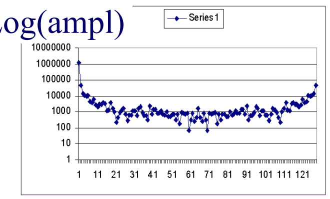
- Dow Jones Industrial index, 6/18/2001-12/21/2001

CMU SCS

DFT and stocks



Log(ampl)



- Dow Jones Industrial index, 6/18/2001-12/21/2001
- just 3 DFT coefficients give very good approximation

freq

CMU SCS

DFT: definition

- Discrete Fourier Transform (n-point):

$$X_f = 1/\sqrt{n} \sum_{t=0}^{n-1} x_t * \exp(-j 2\pi f t / n) \qquad f = 0,\ldots,n-1$$

$$(j = \sqrt{-1})$$

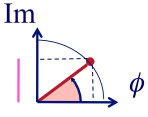
inverse DFT

$$x_t = 1/\sqrt{n} \sum_{f=0}^{n-1} X_f * \exp(+j 2\pi f t / n)$$

CMU SCS

(Reminder)

$$\exp(\phi * j) = \underline{\cos(\phi)} + j * \underline{\sin(\phi)}$$



(fun fact: the equation with the 5 most important numbers:

$$e^{j\pi} + 1 = 0$$

)

CMU SCS

DFT: alternative definition

- Discrete Fourier Transform (n-point):

$$a_f = 1/\sqrt{n} \sum_{t=0}^{n-1} x_t * \cos(-2\pi f t / n) \qquad f = 0,\ldots,n-1$$

$$b_f = 1/\sqrt{n} \sum_{t=0}^{n-1} x_t * \sin(-2\pi f t / n)$$

inverse DFT

$$x_t = 1/\sqrt{n} \sum_{f=0}^{n-1} [a_f * \cos(2\pi f t / n) + j * b_f * \sin(2\pi f t / n)]$$

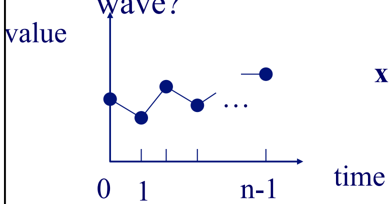
CMU SCS

details

How does it work?

Decomposes signal to a sum of sine (and cosine) waves.

Q:How to assess 'similarity' of **x** with a wave?



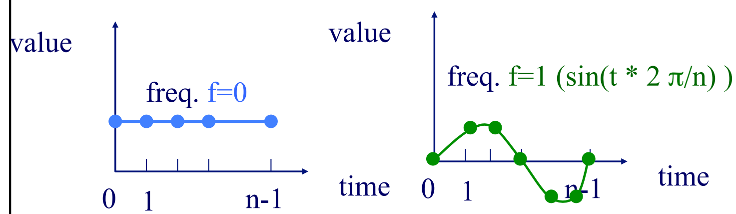
x $=\{x_0, x_1, \ldots x_{n-1}\}$

CMU SCS

details

How does it work?

A: consider the waves with frequency 0, 1, ...; use the inner-product (~cosine similarity)

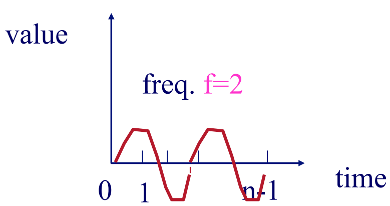


CMU SCS

details

How does it work?

A: consider the waves with frequency 0, 1, ...; use the inner-product (~cosine similarity)

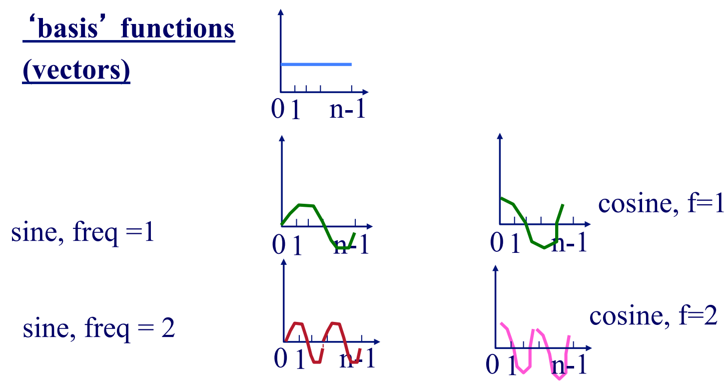


CMU SCS

details

How does it work?

'basis' functions (vectors)



CMU SCS

details

How does it work?

- Basis functions are actually n-dim vectors, **orthogonal** to each other
- 'similarity' of **x** with each of them: inner product
- DFT: ~ all the similarities of **x** with the basis functions

CMU SCS

DFT: definition

- **Good** news: Available in **all** symbolic math packages, eg., in 'mathematica'

 x = [1,2,1,2];

 X = Fourier[x];

 Plot[Abs[X]];

CMU SCS

DFT: definition

(variations:

- 1/n instead of 1/sqrt(n)
- exp(-...) instead of exp(+...)

CMU SCS

DFT: definition

Observations:

- X_f : are complex numbers except
  - X_0 , who is real
- Im (X_f): ~ amplitude of sine wave of frequency *f*
- Re (X_f): ~ amplitude of cosine wave of frequency *f*
- **x:** is the sum of the above sine/cosine waves

CMU SCS

details

DFT: definition

Observation - SYMMETRY property:

$$\boxed{X_f = (X_{n-f})^*}$$

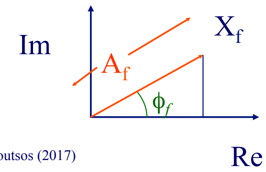
("*": complex conjugate*: (a + b j)* = a - b j)*

CMU SCS

DFT: definition

Definitions

- $A_f = |X_f|$: amplitude of frequency f
- $|X_f|^2 = Re(X_f)^2 + Im(X_f)^2$ = energy of frequency f
- phase ϕ_f at frequency f



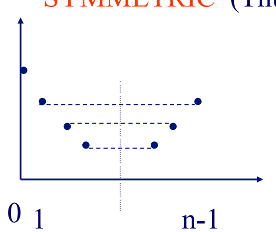
CMU SCS

details

DFT: definition

Amplitude spectrum: $| X_f |$ vs f $(f=0, 1, \ldots n-1)$

SYMMETRIC (Thus, we plot the first half only)



CMU SCS

details

DFT: definition

Phase spectrum $| \phi_f |$ vs f $(f=0, 1, \ldots n-1)$: Anti-symmetric

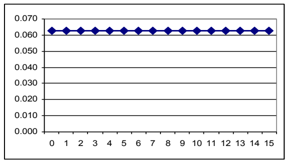
(Rarely used)

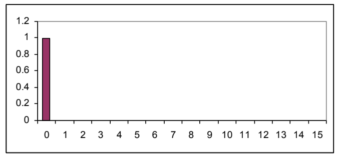
CMU SCS

DFT: examples

flat

Amplitude





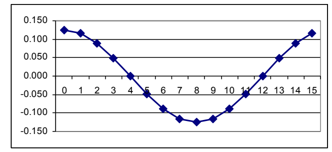
time freq

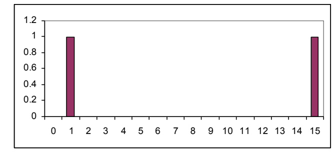
15-826 Copyright: C. Faloutsos (2017) 29

CMU SCS

DFT: examples

Low frequency sinusoid





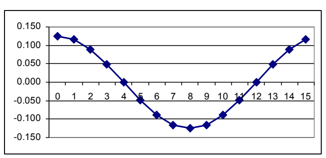
time freq

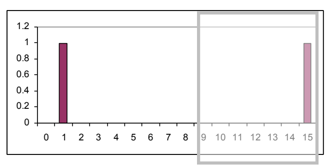
15-826 Copyright: C. Faloutsos (2017) 30

CMU SCS

DFT: examples

- Sinusoid - symmetry property: $X_f = X^*_{n-f}$





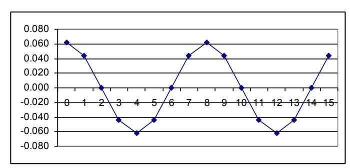
time freq

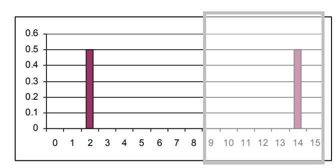
15-826 Copyright: C. Faloutsos (2017) 31

CMU SCS

DFT: examples

- Higher freq. sinusoid





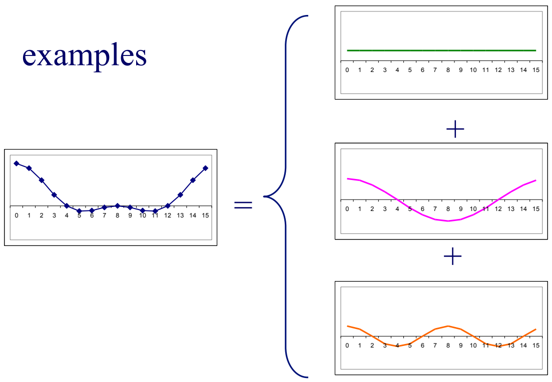
time freq

15-826 Copyright: C. Faloutsos (2017) 32

CMU SCS

DFT: examples

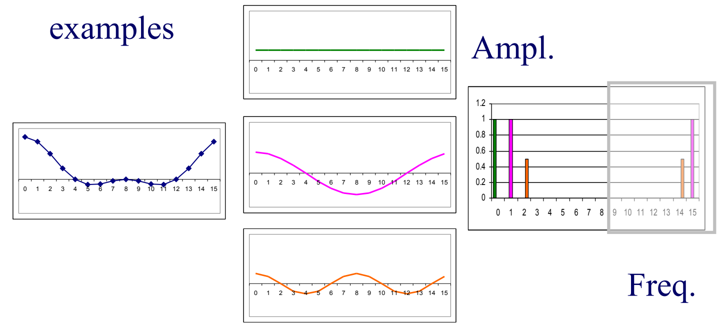
examples



CMU SCS

DFT: examples

examples

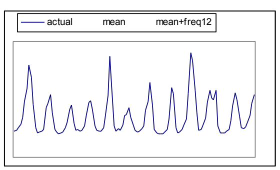


CMU SCS

DFT: Amplitude spectrum

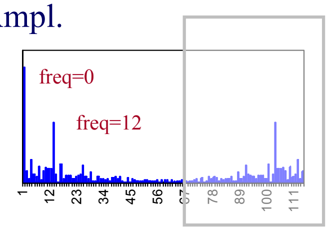
Amplitude: $A_f^2 = \mathrm{Re}^2(X_f) + \mathrm{Im}^2(X_f)$

count



year

Ampl.

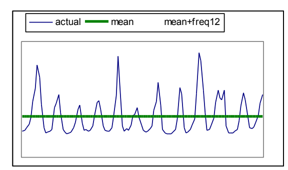


Freq.

CMU SCS

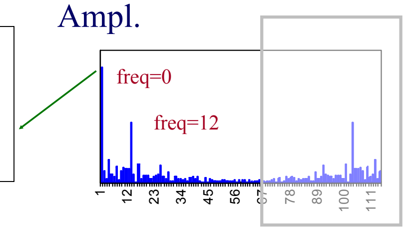
DFT: Amplitude spectrum

count



year

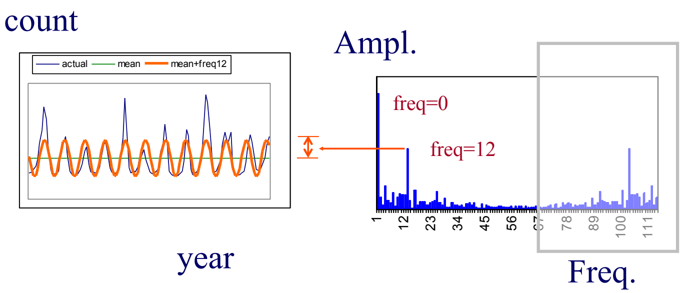
Ampl.



Freq.

CMU SCS

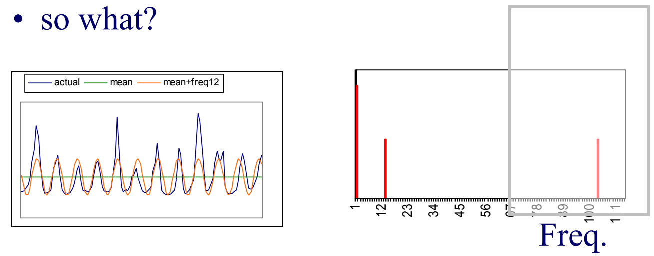
DFT: Amplitude spectrum



CMU SCS

DFT: Amplitude spectrum

- excellent approximation, with only 2 frequencies!
- so what?



CMU SCS

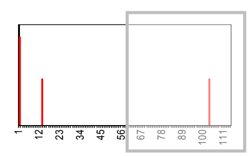
DFT: Amplitude spectrum

- excellent approximation, with only 2 frequencies!
- so what?
- A1: compression
- A2: pattern discovery
- (A3: forecasting)

CMU SCS

DFT: Amplitude spectrum

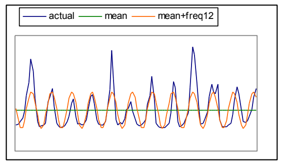
- excellent approximation, with only 2 frequencies!
- so what?
- A1: **(lossy) compression**
- A2: pattern discovery



CMU SCS

DFT: Amplitude spectrum

- excellent approximation, with only 2 frequencies!
- so what?
- A1: (lossy) compression
- A2: **pattern discovery**



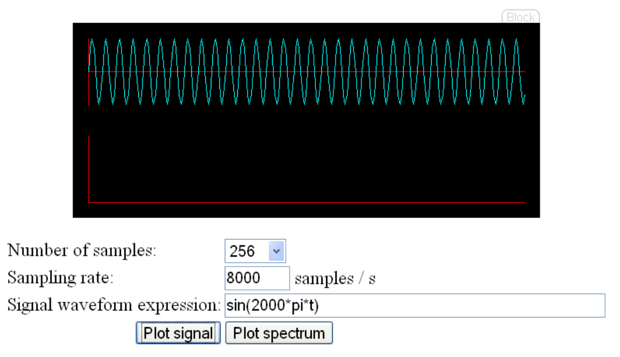
CMU SCS

DFT: Amplitude spectrum

- Let's see it in action (defunct now…)
- (http://www.dsptutor.freeuk.com/jsanalyser/FFTSpectrumAnalyser.html)
- plain sine
- phase shift
- two sine waves
- the 'chirp' function
- http://ion.researchsystems.com/

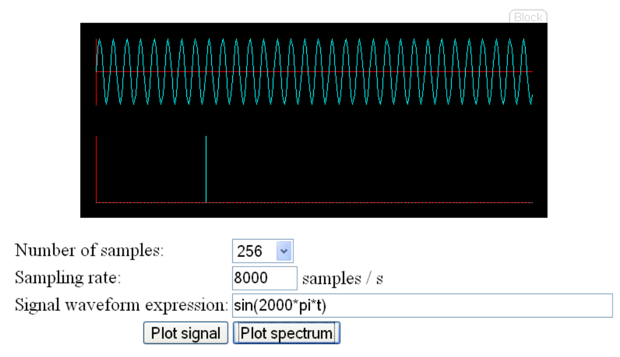
CMU SCS

Plain sine



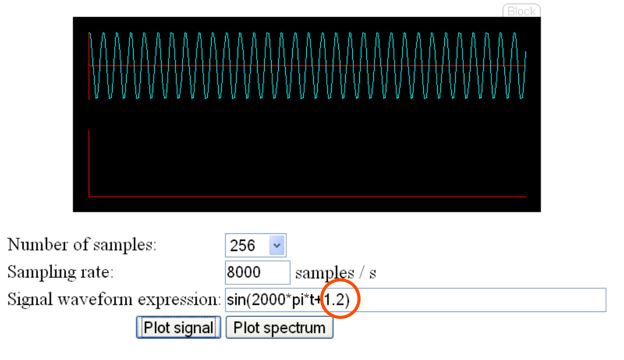
CMU SCS

Plain sine



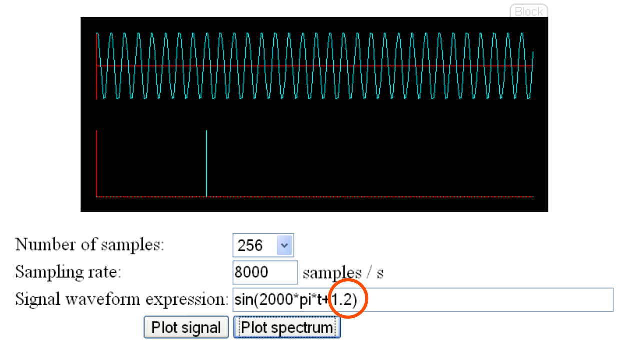
CMU SCS

Plain sine – phase shift



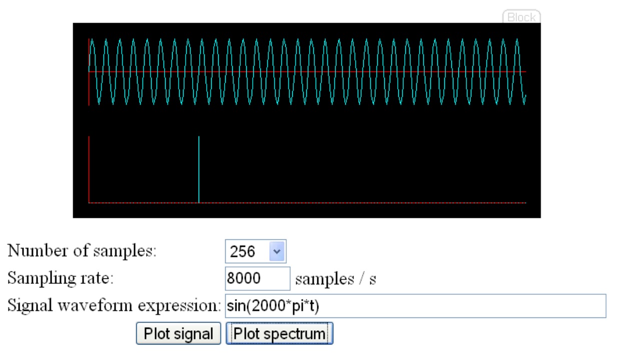
CMU SCS

Plain sine – phase shift



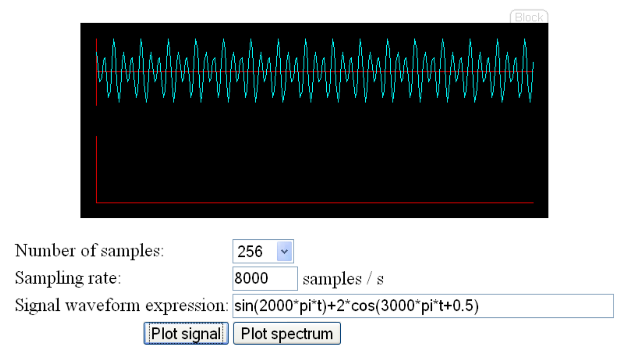
CMU SCS

Plain sine



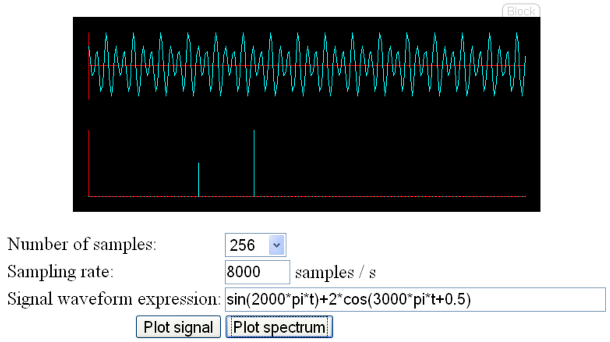
CMU SCS

Two sines



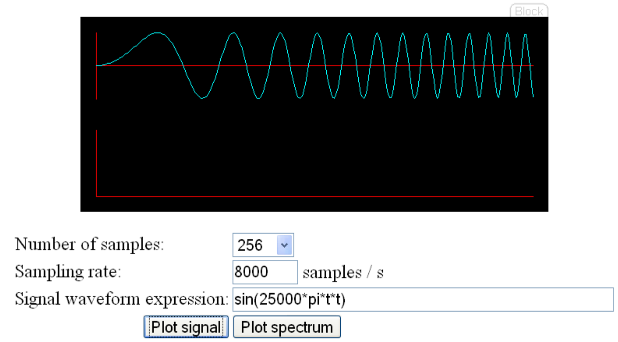
CMU SCS

Two sines



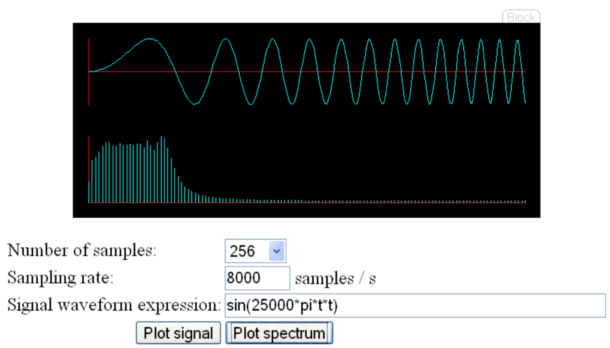
CMU SCS

Chirp



CMU SCS

Chirp



CMU SCS

Another applet

http://www.falstad.com/fourier/

(seems virus-free – but scan, before you install)

Local copy, INTERNAL @ CMU:

www.cs.cmu.edu/~christos/courses/826-resources/DEMOS/FFT_applet/

Start DEMO

CMU SCS

Properties

- Time shift sounds the same
  - Changes only phase, not amplitudes
- Sawtooth has almost all frequencies
  - With decreasing amplitude
- Spike has all frequencies

15-826 53

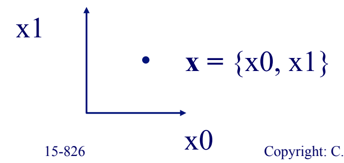
CMU SCS

DFT: Parseval's theorem

$$\text{sum}(\, x_t^{\,2}\,) = \text{sum}\,(\, |\, X_f\, |^{\,2}\,)$$

Ie., DFT preserves the 'energy'

or, alternatively: it does an axis rotation:



15-826 54

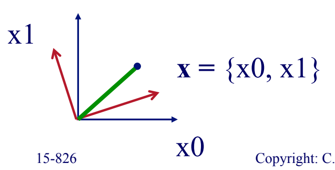
CMU SCS

DFT: Parseval's theorem

$$\text{sum}(\, x_t^{\,2}\,) = \text{sum}\,(\, |\, X_f\, |^{\,2}\,)$$

Ie., DFT preserves the 'energy'

or, alternatively: it does an axis rotation:



15-826 55

CMU SCS

DSP - Detailed outline

- DFT
  - what
  - why
  - how
  - Arithmetic examples
  - properties / observations
  - DCT
  - 2-d DFT
  - Fast Fourier Transform (FFT)

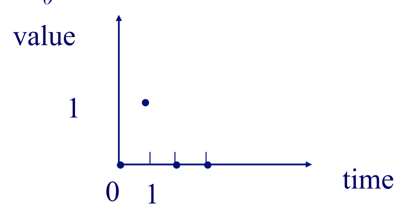
15-826 56

CMU SCS

details

Arithmetic examples

- Impulse function: **x** = { 0, 1, 0, 0} (n = 4)
- X_0=?



15-826 Copyright: C. Faloutsos (2017) 57

CMU SCS

details

Arithmetic examples

- Impulse function: **x** = { 0, 1, 0, 0} (n = 4)
- X_0=?
- A: X_0 = 1/sqrt(4) * 1* exp(-j 2 π 0 / n) = 1/2
- X_1=?
- X_2=?
- X_3=?

15-826 Copyright: C. Faloutsos (2017) 58

CMU SCS

details

Arithmetic examples

- Impulse function: **x** = { 0, 1, 0, 0} (n = 4)
- X_0=?
- A: X_0 = 1/sqrt(4) * 1* exp(-j 2 π 0 / n) = 1/2
- X_1= -1/2 j
- X_2= - 1/2
- X_3= + 1/2 j
- Q: does the 'symmetry' property hold?

15-826 Copyright: C. Faloutsos (2017) 59

CMU SCS

details

Arithmetic examples

- Impulse function: **x** = { 0, 1, 0, 0} (n = 4)
- X_0=?
- A: X_0 = 1/sqrt(4) * 1* exp(-j 2 π 0 / n) = 1/2
- X_1= -1/2 j
- X_2= - 1/2
- X_3= + 1/2 j
- Q: does the 'symmetry' property hold?
- A: Yes (of course)

15-826 Copyright: C. Faloutsos (2017) 60

details

Arithmetic examples

- Impulse function: **x** = { 0, 1, 0, 0} (n = 4)
- X_0=?
- A: X_0 = 1/sqrt(4) * 1* exp(-j 2 π 0 / n) = 1/2
- X_1= -1/2 j
- X_2= - 1/2
- X_3= + 1/2 j
- Q: check Parseval's theorem

details

Arithmetic examples

- Impulse function: **x** = { 0, 1, 0, 0} (n = 4)
- X_0=?
- A: X_0 = 1/sqrt(4) * 1* exp(-j 2 π 0 / n) = 1/2
- X_1= -1/2 j
- X_2= - 1/2
- X_3= + 1/2 j
- Q: (Amplitude) spectrum?

Arithmetic examples

- Impulse function: **x** = { 0, 1, 0, 0} (n = 4)
- X_0=?
- A: X_0 = 1/sqrt(4) * 1* exp(-j 2 π 0 / n) = 1/2
- X_1= -1/2 j
- X_2= - 1/2
- X_3= + 1/2 j
- Q: (Amplitude) spectrum?
- A: FLAT!

Arithmetic examples

- Q: What does this mean?

CMU SCS

Arithmetic examples

- Q: What does this mean?
- A: All frequencies are equally important ->
  - we need *n* numbers in the frequency domain to represent just one non-zero number in the time domain!
  - "*frequency leak*"

CMU SCS

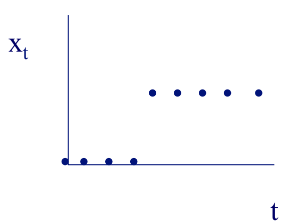
DSP - Detailed outline

- DFT
  - what
  - why
  - how
  - Arithmetic examples
  - properties / observations
  - DCT
  - 2-d DFT
  - Fast Fourier Transform (FFT)

CMU SCS

Observations

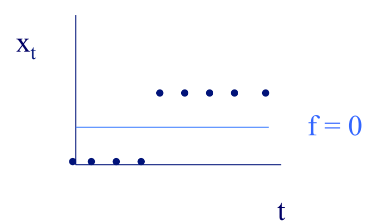
- DFT of 'step' function:
  x = { 0, 0, ..., 0, 1, 1, ... 1}



CMU SCS

Observations

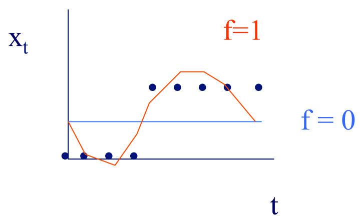
- DFT of 'step' function:
  x = { 0, 0, ..., 0, 1, 1, ... 1}



CMU SCS

Observations

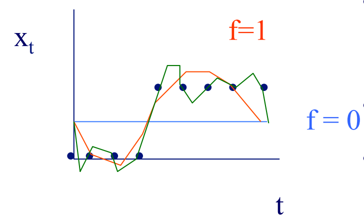
- DFT of 'step' function:
  x = { 0, 0, ..., 0, 1, 1, ... 1}



CMU SCS

Observations

- DFT of 'step' function:
  x = { 0, 0, ..., 0, 1, 1, ... 1}

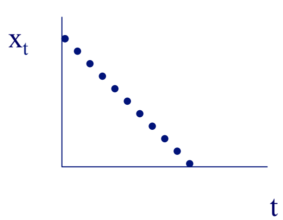


- the more frequencies, the better the approx.
- 'ringing' becomes worse
- reason: discontinuities; trends

CMU SCS

Observations

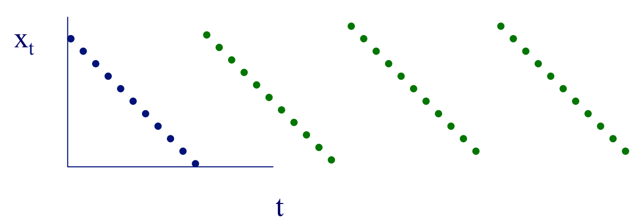
- Ringing for trends: because DFT 'sub-consciously' replicates the signal



CMU SCS

Observations

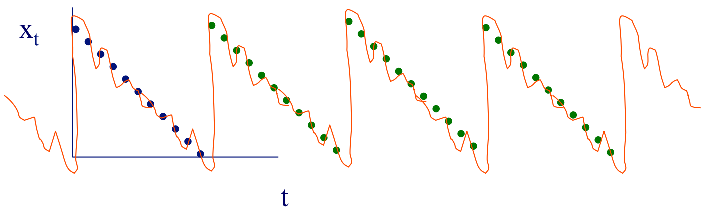
- Ringing for trends: because DFT 'sub-consciously' replicates the signal



CMU SCS

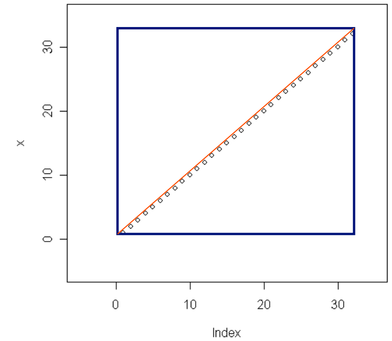
Observations

- Ringing for trends: because DFT ‘sub-consciously’ replicates the signal



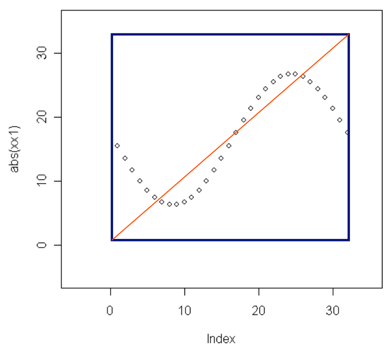
CMU SCS

original



CMU SCS

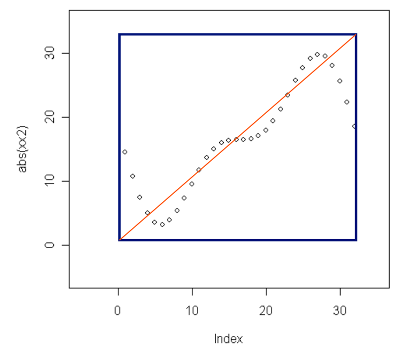
DC and 1st



CMU SCS

DC and 1st

And 2nd

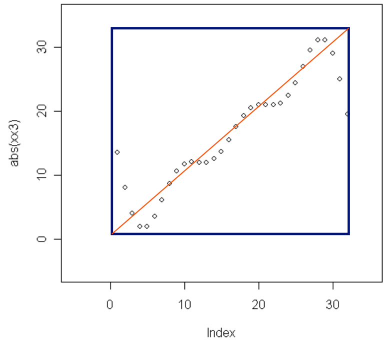


CMU SCS

DC and 1st

And 2nd

And 3rd



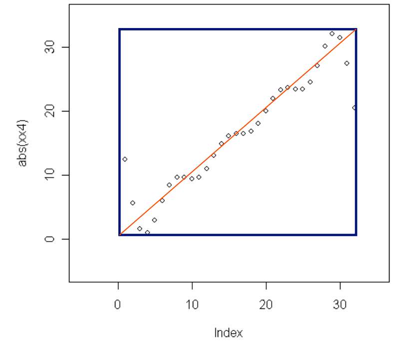
CMU SCS

DC and 1st

And 2nd

And 3rd

And 4th



CMU SCS

Observations

- Q: DFT of a sinusoid, eg.

  $x_t = 3 \sin(2 \pi / 4\ t)$

(t = 0, ... , 3)

- Q: X_0 = ?
- Q: X_1 = ?
- Q: X_2 = ?
- Q: X_3 = ?

CMU SCS

Observations

- Q: DFT of a sinusoid, eg.

  $x_t = 3 \sin(2 \pi / 4\ t)$

(t = 0, ... , 3)

- Q: $X_0 = 0$
- Q: $X_1 = -3 j$
- Q: $X_2 = 0$
- Q: $X_3 = 3j$

•check 'symmetry'
•check Parseval

CMU SCS

Observations

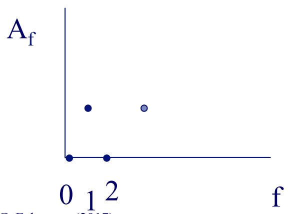
- Q: DFT of a sinusoid, eg.

  $x_t = 3 \sin(2 \pi / 4\ t)$

(t = 0, ... , 3)

- Q: $X_0 = 0$
- Q: $X_1 = -3\ j$
- Q: $X_2 = 0$
- Q: $X_3 = 3j$

•Does this make sense?



15-826 Copyright: C. Faloutsos (2017) 81

CMU SCS

Property

- Shifting **x** in time does NOT change the amplitude spectrum
- eg., **x** = { 0 0 0 1} and **x'** = { 0 1 0 0 }: same (flat) amplitude spectrum
- (only the phase spectrum changes)
- Useful property when we search for patterns that may 'slide'

15-826 Copyright: C. Faloutsos (2017) 82

CMU SCS

Summary of properties

- Spike in time: -> all frequencies
- Step/Trend: -> ringing (~ all frequencies)
- Single/dominant sinusoid: -> spike in spectrum
- Time shift -> same amplitude spectrum

15-826 Copyright: C. Faloutsos (2017) 83

CMU SCS

DSP - Detailed outline

- DFT
  - what
  - why
  - how
  - Arithmetic examples
  - properties / observations
  - DCT
  - 2-d DFT
  - Fast Fourier Transform (FFT)

15-826 Copyright: C. Faloutsos (2017) 84

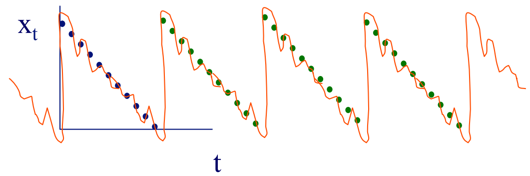
CMU SCS

details

DCT

Discrete Cosine Transform

- motivation#1: DFT gives complex numbers
- motivation#2: how to avoid the 'frequency leak' of DFT on trends?

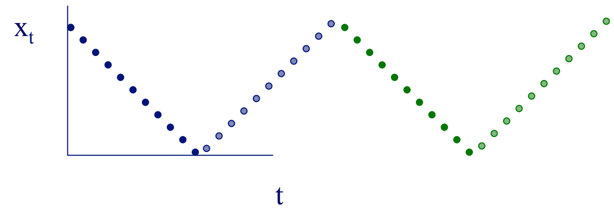


CMU SCS

details

DCT

- brilliant solution to both problems: mirror the sequence, do DFT, and drop the redundant entries!



CMU SCS

details

DCT

- (see Numerical Recipes for exact formulas)

CMU SCS

DCT - properties

- it gives real numbers as the result
- it has no problems with trends
- it is very good when x_t and $x_{(t+1)}$ are correlated

(thus, is used in JPEG, for image compression)

CMU SCS

DSP - Detailed outline

- DFT
  - what
  - why
  - how
  - Arithmetic examples
  - properties / observations
  - DCT
  - 2-d DFT
  - Fast Fourier Transform (FFT)

CMU SCS

2-d DFT

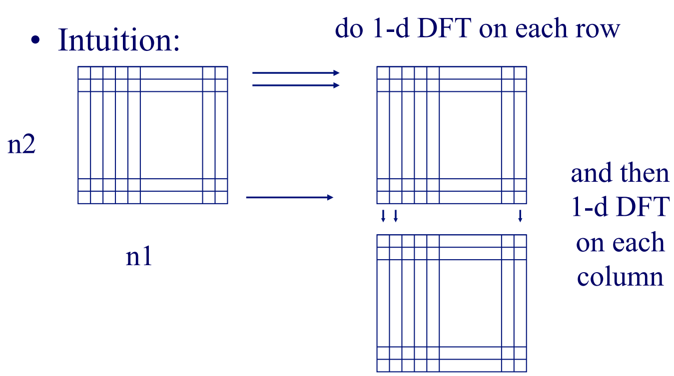
- Definition:

$$X_{f_1,f_2} = \frac{1}{\sqrt{n_1}} \frac{1}{\sqrt{n_2}} \sum_{i_1=0}^{n_1-1} \sum_{i_1=0}^{n_2-1} x_{i_1,i_2} \exp(-2\pi j i_1 f_1 / n_1) \exp(-2\pi j i_2 f_2 / n_2)$$

CMU SCS

2-d DFT

- Intuition:



CMU SCS

2-d DFT

- Quiz: how do the basis functions look like?
- for f1= f2 =0
- for f1=1, f2=0
- for f1=1, f2=1

CMU SCS

2-d DFT

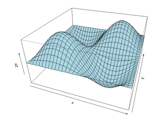
- Quiz: how do the basis functions look like?
- for f1= f2 =0 flat
- for f1=1, f2=0 wave on x; flat on y
- for f1=1, f2=1 ~ egg-carton

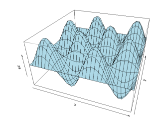
15-826 Copyright: C. Faloutsos (2017) 93

CMU SCS

2-d DFT

- Quiz: how do the basis functions look like?
- for f1= f2 =0 flat
- for f1=1, f2=0 wave on x; flat on y
- for f1=1, f2=1 ~ egg-carton





15-826 Copyright: C. Faloutsos (2017) 94

CMU SCS

DSP - Detailed outline

- DFT
  - what
  - why
  - how
  - Arithmetic examples
  - properties / observations
  - DCT
  - 2-d DFT
  - Fast Fourier Transform (FFT)

15-826 Copyright: C. Faloutsos (2017) 95

CMU SCS

details

FFT

- What is the complexity of DFT?

$$X_f = 1/\sqrt{n} \sum_{t=0}^{n-1} x_t * \exp(-j 2\pi\, tf / n)$$

15-826 Copyright: C. Faloutsos (2017) 96

CMU SCS

details

FFT

- What is the complexity of DFT?

$$X_f = 1/\sqrt{n} \sum_{t=0}^{n-1} x_t * \exp(-j 2\pi t f / n)$$

- A: Naively, O(n^2)

15-826 Copyright: C. Faloutsos (2017) 97

CMU SCS

details

FFT

- However, if n is a power of 2 (or a number with many divisors), we can make it

  O(n log n)

Main idea: if we know the DFT of the odd time-ticks, and of the even time-ticks, we can quickly compute the whole DFT

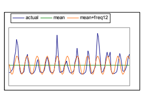
Details: in Num. Recipes

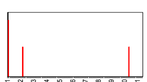
15-826 Copyright: C. Faloutsos (2017) 98

CMU SCS

DFT - Conclusions

- It spots periodicities (with the '**amplitude spectrum**')
- can be quickly computed (O(n log n)), thanks to the FFT algorithm.
- **standard** tool in signal processing (speech, image etc signals)





Freq.

15-826 Copyright: C. Faloutsos (2017) 99

CMU SCS

Detailed outline

- primary key indexing
- ..
- multimedia
  - Digital Signal Processing (DSP) tools
    - Discrete Fourier Transform (DFT)
    - Discrete Wavelet Transform (DWT)

15-826 Copyright: C. Faloutsos (2017) 100

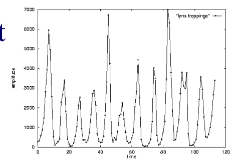
CMU SCS

Reminder: Problem:

Goal: given a signal (eg., #packets over time)

Find: patterns, periodicities, and/or **compress**

count



lynx caught per year
(packets per day;
virus infections per month)

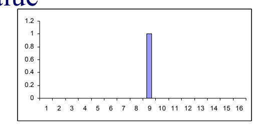
year

CMU SCS

Wavelets - DWT

- DFT is great - but, how about compressing a spike?

value



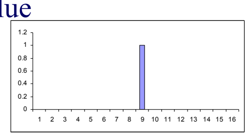
time

CMU SCS

Wavelets - DWT

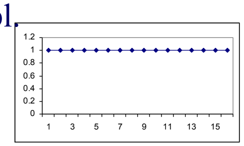
- DFT is great - but, how about compressing a spike?
- A: Terrible - all DFT coefficients needed!

value



time

Ampl.



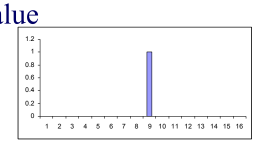
Freq.

CMU SCS

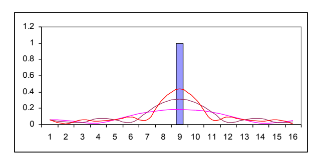
Wavelets - DWT

- DFT is great - but, how about compressing a spike?
- A: Terrible - all DFT coefficients needed!

value



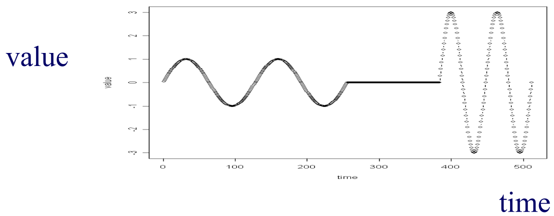
time



CMU SCS

Wavelets - DWT

- Similarly, DFT suffers on short-duration waves (eg., baritone, silence, soprano)

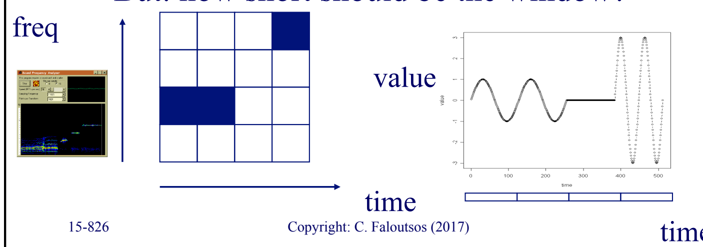


15-826 Copyright: C. Faloutsos (2017) 105

CMU SCS

Wavelets - DWT

- Solution#1: Short window Fourier transform (SWFT)
- But: how short should be the window?

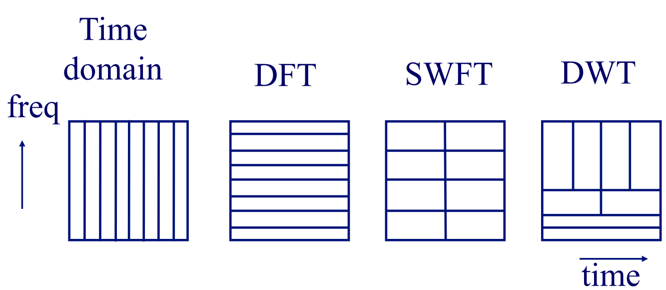


15-826 Copyright: C. Faloutsos (2017)

CMU SCS

Wavelets - DWT

- Answer: **multiple** window sizes! -> DWT

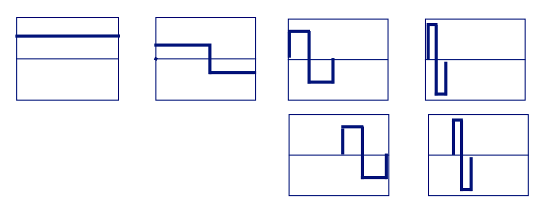


15-826 Copyright: C. Faloutsos (2017) 107

CMU SCS

Haar Wavelets

- subtract sum of left half from right half
- repeat recursively for quarters, eight-ths, ...



15-826 Copyright: C. Faloutsos (2017) 108

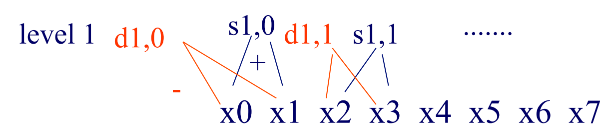
CMU SCS

Wavelets - construction

x0 x1 x2 x3 x4 x5 x6 x7

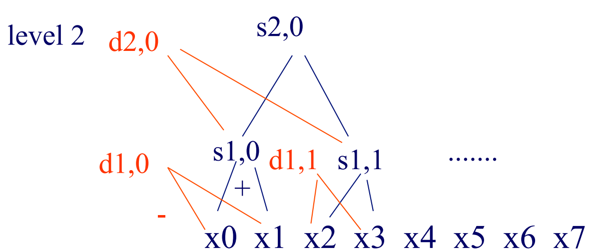
CMU SCS

Wavelets - construction



CMU SCS

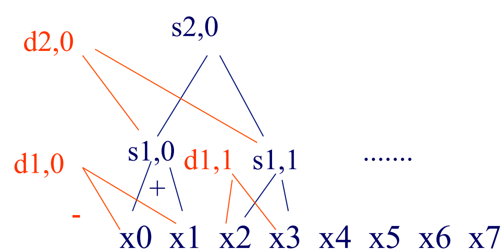
Wavelets - construction



CMU SCS

Wavelets - construction

etc ...

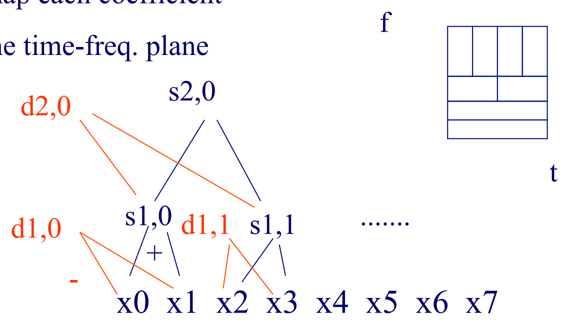


CMU SCS

Wavelets - construction

Q: map each coefficient

on the time-freq. plane



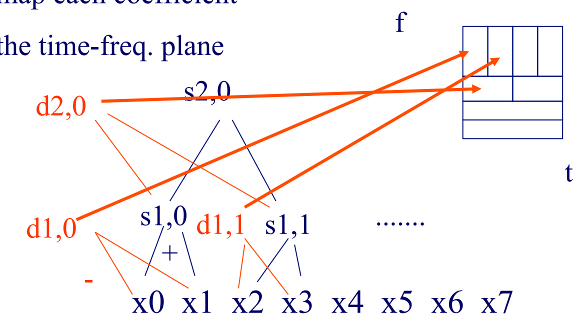
15-826 Copyright: C. Faloutsos (2017) 113

CMU SCS

Wavelets - construction

Q: map each coefficient

on the time-freq. plane



15-826 Copyright: C. Faloutsos (2017) 114

CMU SCS

Haar wavelets - code

```
#!/usr/bin/perl5
# expects a file with numbers
# and prints the dwt transform
# The number of time-ticks should be a power of 2
# USAGE
#    haar.pl <fname>

my @vals=();
my @smooth; # the smooth component of the signal
my @diff;   # the high-freq. component

# collect the values into the array @val
while(<>){
        @vals = ( @vals ,  split );
}
```

```
my $len = scalar(@vals);
my $half = int($len/2);
while($half >= 1 ){
  for(my $i=0; $i< $half; $i++){
        $diff [$i] = ($vals[2*$i] - $vals[2*$i + 1] )/ sqrt(2);
        print "\t", $diff[$i];
        $smooth [$i] = ($vals[2*$i] + $vals[2*$i + 1] )/ sqrt(2);
  }
  print "\n";
  @vals = @smooth;
  $half = int($half/2);
}
print "\t", $vals[0], "\n" ;     # the final, smooth component
```

Also at: www.cs.cmu.edu/~christos/SRC/DWT-Haar-all.tar

15-826 Copyright: C. Faloutsos (2017) 115

CMU SCS

Wavelets - construction

Observation1:

- '+' can be some weighted addition
- '-' is the corresponding weighted difference ('Quadrature mirror filters')

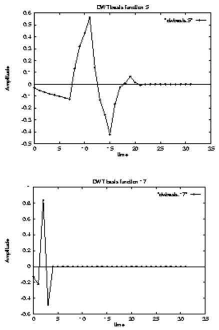
Observation2: unlike DFT/DCT,

- there are *many* wavelet bases: Haar, Daubechies-4, Daubechies-6, Coifman, Morlet, Gabor, ...

15-826 Copyright: C. Faloutsos (2017) 116

CMU SCS

Wavelets - how do they look like?

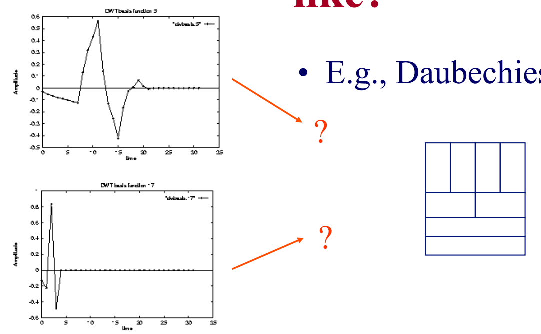


- E.g., Daubechies-4

CMU SCS

Wavelets - how do they look like?

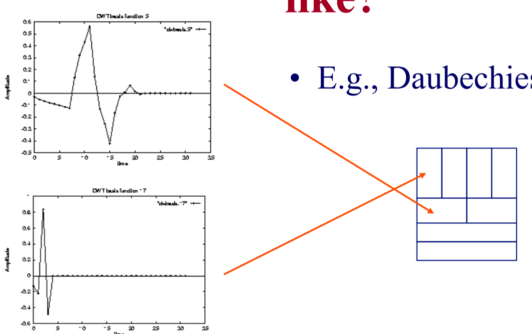
- E.g., Daubechies-4



CMU SCS

Wavelets - how do they look like?

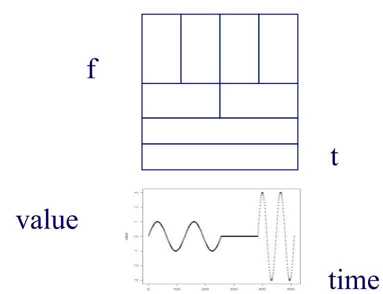
- E.g., Daubechies-4



CMU SCS

Wavelets - Drill#1:

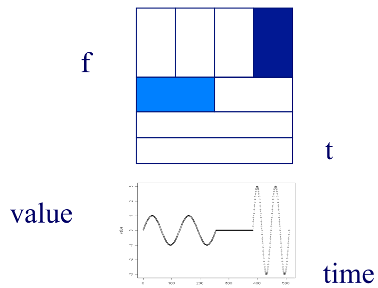
- Q: baritone/silence/soprano - DWT?



CMU SCS

Wavelets - Drill#1:

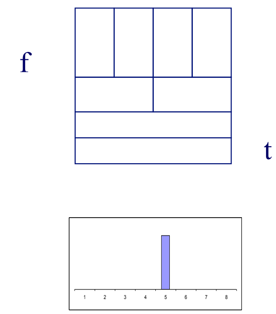
- Q: baritone/silence/soprano - DWT?



CMU SCS

Wavelets - Drill#2:

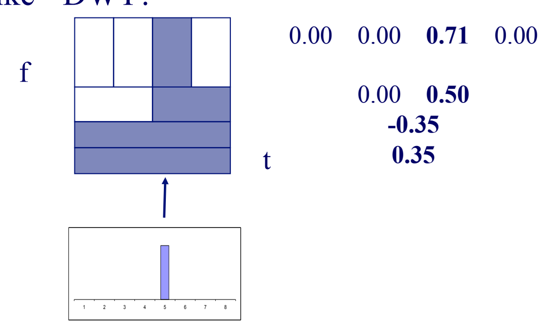
- Q: spike - DWT?



CMU SCS

Wavelets - Drill#2:

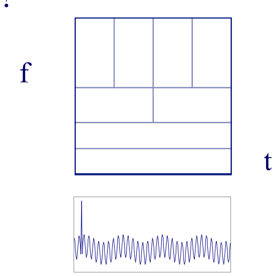
- Q: spike - DWT?



CMU SCS

Wavelets - Drill#3:

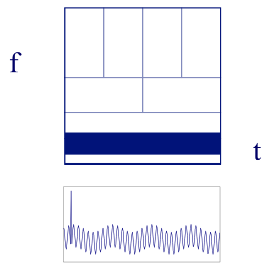
- Q: weekly + daily periodicity, + spike - DWT?



CMU SCS

Wavelets - Drill#3:

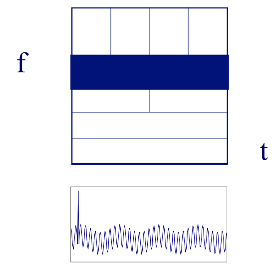
- Q: **weekly** + daily periodicity, + spike - DWT?



CMU SCS

Wavelets - Drill#3:

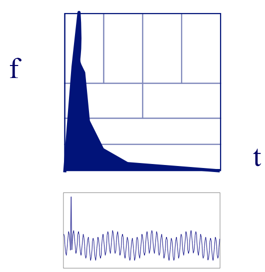
- Q: weekly + **daily** periodicity, + spike - DWT?



CMU SCS

Wavelets - Drill#3:

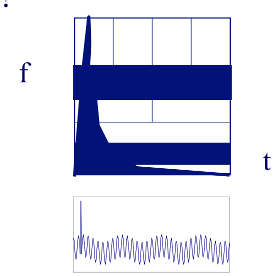
- Q: weekly + daily periodicity, + **spike** - DWT?



CMU SCS

Wavelets - Drill#3:

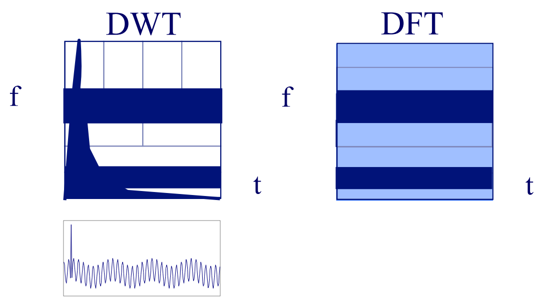
- Q: weekly + daily periodicity, + spike - DWT?



CMU SCS

Wavelets - Drill#3:

- Q: DFT?



CMU SCS

Wavelets - Drill:

Let’s see it live:

http://dsp.rice.edu/software/dsp-teaching-tools

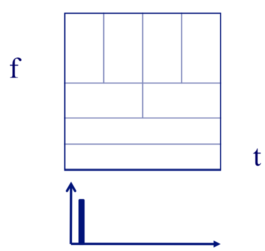
delta; cosine; cosine2; chirp

- Haar vs Daubechies-4, -6, etc

CMU SCS

Delta?

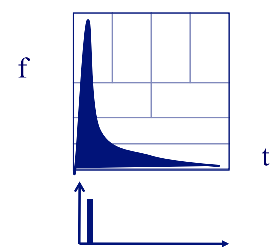
x(0)=1; x(t)= 0 elsewhere



CMU SCS

Delta?

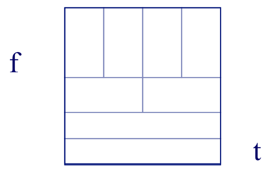
x(0)=1; x(t)= 0 elsewhere



CMU SCS

2 cosines?

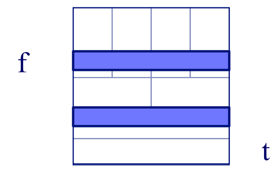
x(t) = cos(2 * pi * 4 * t / 1024) +
5 * cos(2 * pi * 8 * t / 1024)



CMU SCS

2 cosines?

x(t) = cos(2 * pi * 4 * t / 1024) +
5 * cos(2 * pi * 8 * t / 1024)

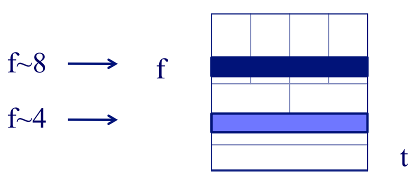


Which one
is for freq.=4?

CMU SCS

2 cosines?

x(t) = cos(2 * pi * 4 * t / 1024) +
5 * cos(2 * pi * 8 * t / 1024)

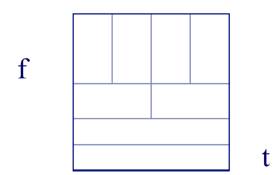


Which one
is for freq.=4?

CMU SCS

Chirp?

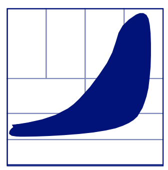
x(t) = cos(2 * pi * t * t / 1024)



CMU SCS

Chirp?

x(t) = cos(2 * pi * t * t / 1024)

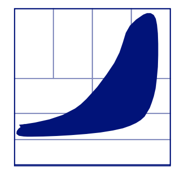
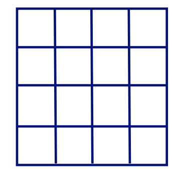
f

t

CMU SCS

Chirp?

x(t) = cos(2 * pi * t * t / 1024)

SWFT?

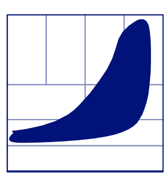
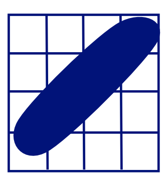
f

t


CMU SCS

Chirp?

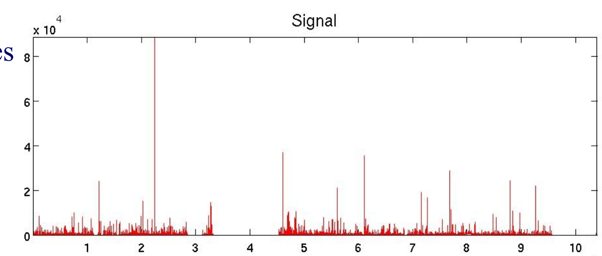
x(t) = cos(2 * pi * t * t / 1024)

SWFT

f

t


CMU SCS

More examples (BGP updates)

#updates


time

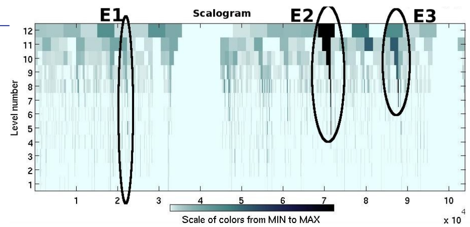
BGP-lens: Patterns and Anomalies in Internet Routing Updates B. Aditya Prakash et al, SIGKDD 2009

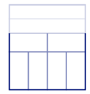
CMU SCS

More examples (BGP updates)

Low freq.:
omitted

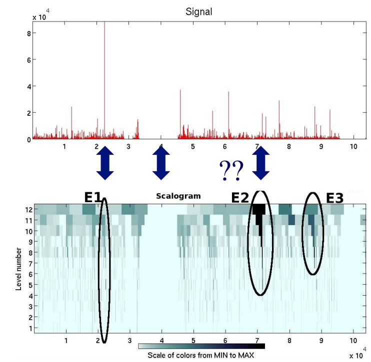
freq.





CMU SCS

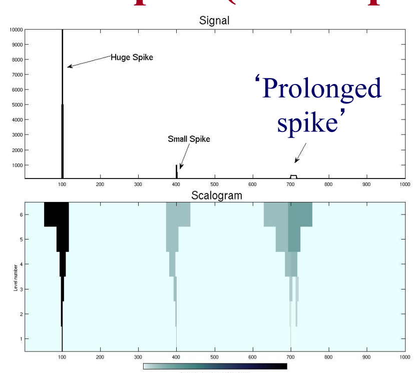
More examples (BGP updates)



freq.

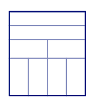
CMU SCS

More examples (BGP updates)



'Prolonged
spike'

freq.

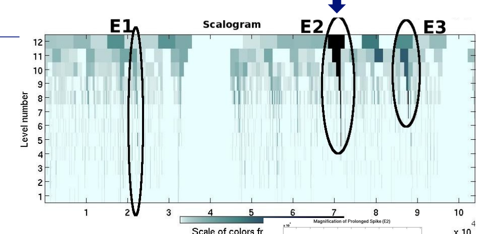


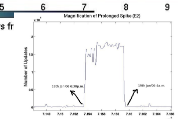
CMU SCS

More examples (BGP updates)

15K msgs, for several hours: 6pm-4am

freq.





Wavelets - Drill

- Or use 'R', 'octave' or 'matlab' – R:

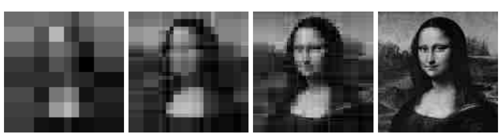
```
install.packages("wavelets")
library("wavelets")
X1<-c(1,2,3,4,5,6,7,8)
dwt(X1, n.levels=3, filter="d4")
mra(X1, n.levels=3, filter="d4")
```


Wavelets - k-dimensions?

- easily defined for any dimensionality (like DFT, DCT)

Wavelets - example

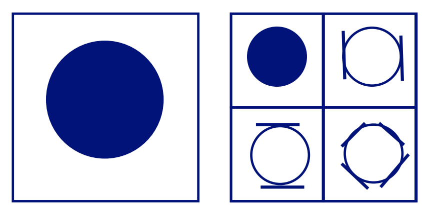
http://grail.cs.washington.edu/projects/query/
Wavelets achieve *great* compression:



20 100 400 16,000
coefficients

Wavelets - intuition

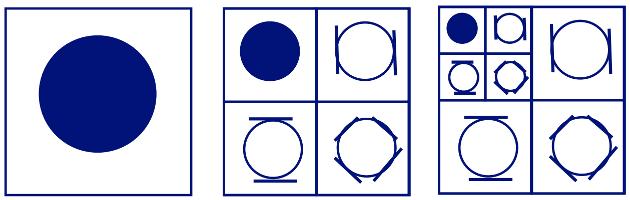
- Edges (horizontal; vertical; diagonal)



CMU SCS

Wavelets - intuition

- Edges (horizontal; vertical; diagonal)
- recurse

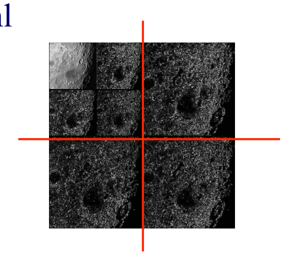


15-826 Copyright: C. Faloutsos (2017) 149

CMU SCS

Wavelets - intuition

- Edges (horizontal; vertical; diagonal)
- http://www331.jpl.nasa.gov/public/wave.html

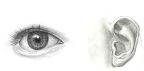


15-826 Copyright: C. Faloutsos (2017) 150

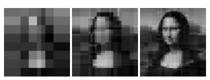
CMU SCS

Advantages of Wavelets

- Better compression (better RMSE with same number of coefficients)
- closely related to the processing of the mammalian eye and ear



- Good for progressive transmission



- handle spikes well
- usually, fast to compute (O(*n*)!)

15-826 Copyright: C. Faloutsos (2017) 151

CMU SCS

Overall Conclusions

- DFT, DCT spot periodicities
- DWT : multi-resolution - matches processing of mammalian ear/eye better
- All three: powerful tools for compression, pattern detection in real signals
- All three: included in math packages (matlab, R, mathematica, ...)

15-826 Copyright: C. Faloutsos (2017) 152

CMU SCS

Resources

- Numerical Recipes in C: great description, intuition and code for all three tools
- *xwpl:* open source wavelet package from Yale, with excellent GUI.

CMU SCS

Resources (cont'd)

- (defunct?) http://www.dsptutor.freeuk.com/jsanalyser/FFTSpectrumAnalyser.html : Nice java applets
- http://www.relisoft.com/freeware/freq.html : voice frequency analyzer (needs microphone – MSwindows only)

CMU SCS

Resources (cont'd)

- www-dsp.rice.edu/software/EDU/mra.shtml (wavelets and other demos)
- R ('install.packages("wavelets"))