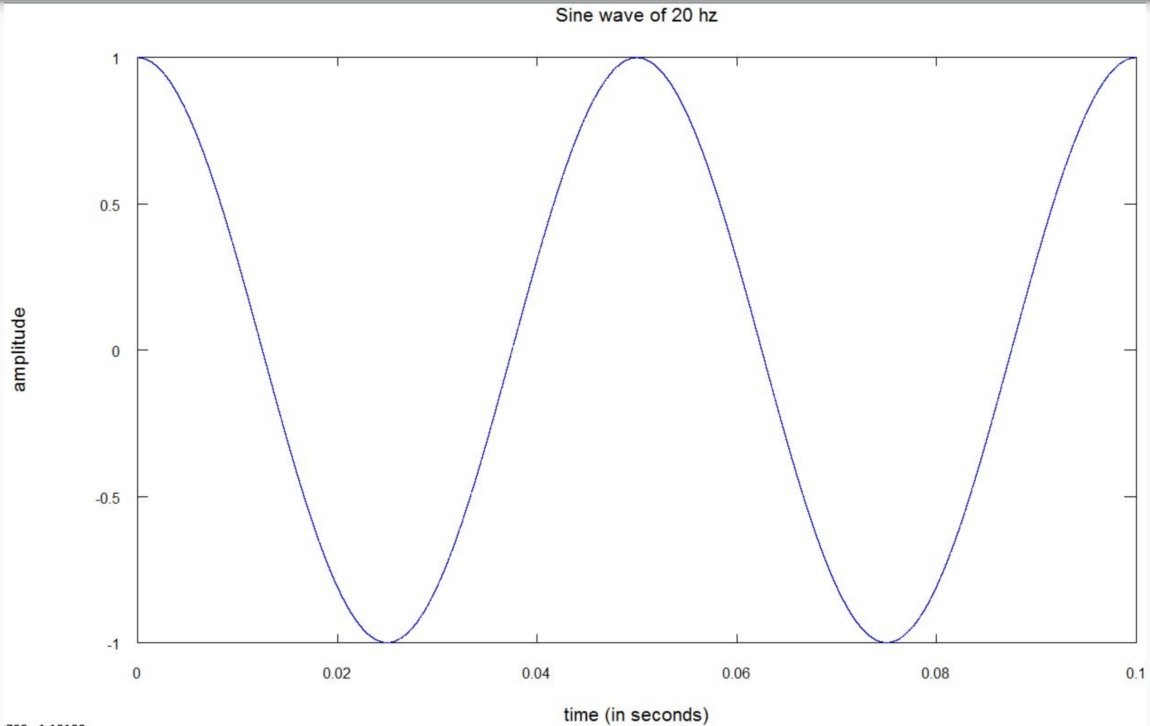
Shazam - Audio Fingerprinting

Arpit Mohan
Exotel

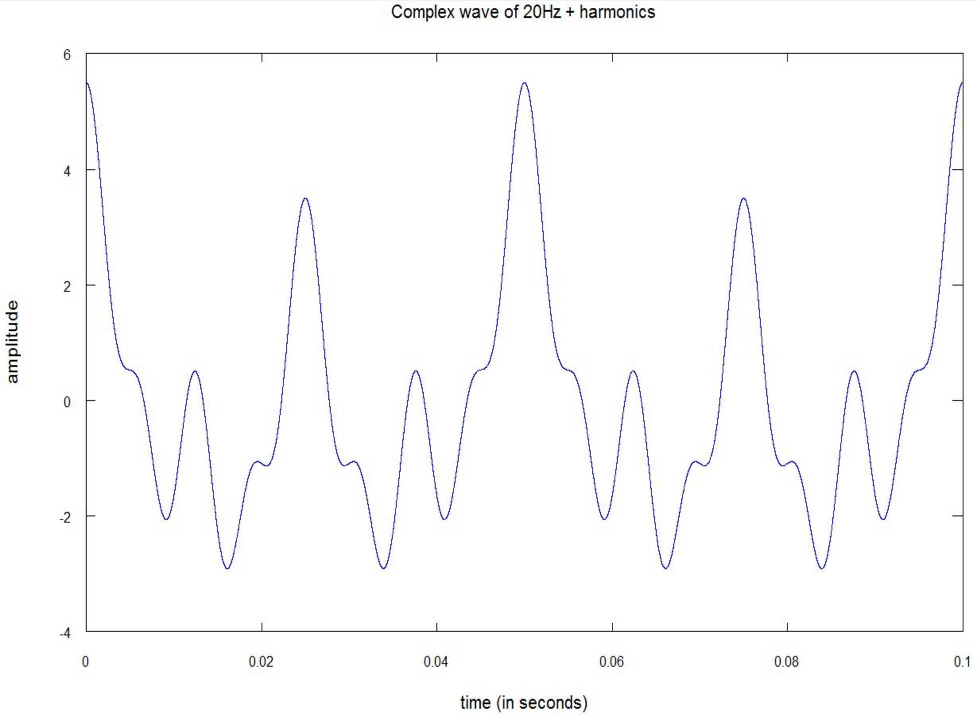
Agenda

- Basics of Audio
- Filtering
- Storing the Data
- Searching for the Data

Basics of Audio
Frequency & Amplitude

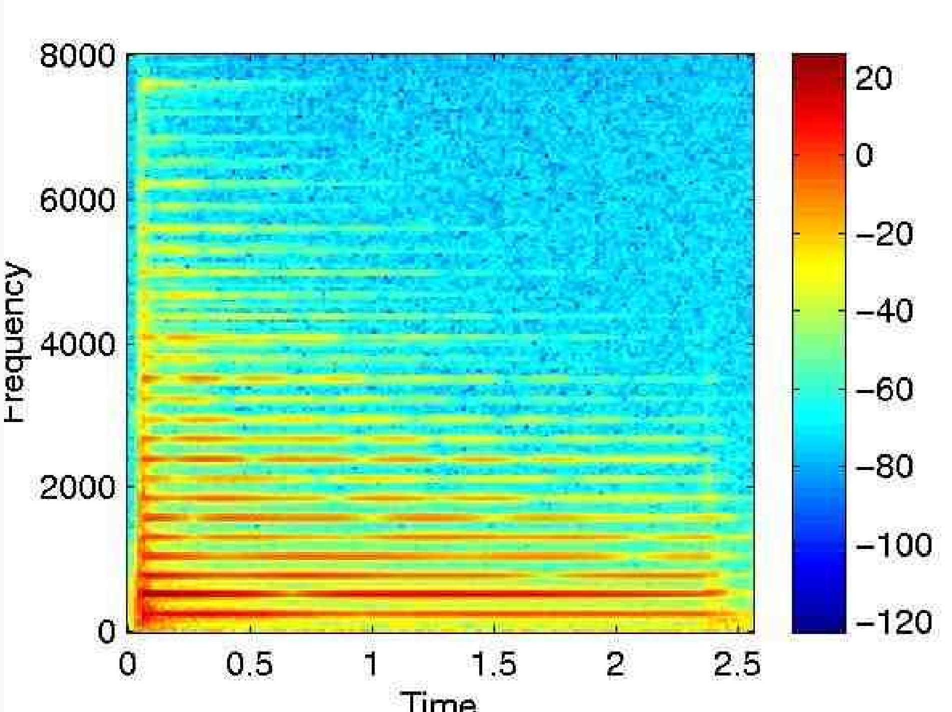


Basics of Audio
Frequency & Amplitude

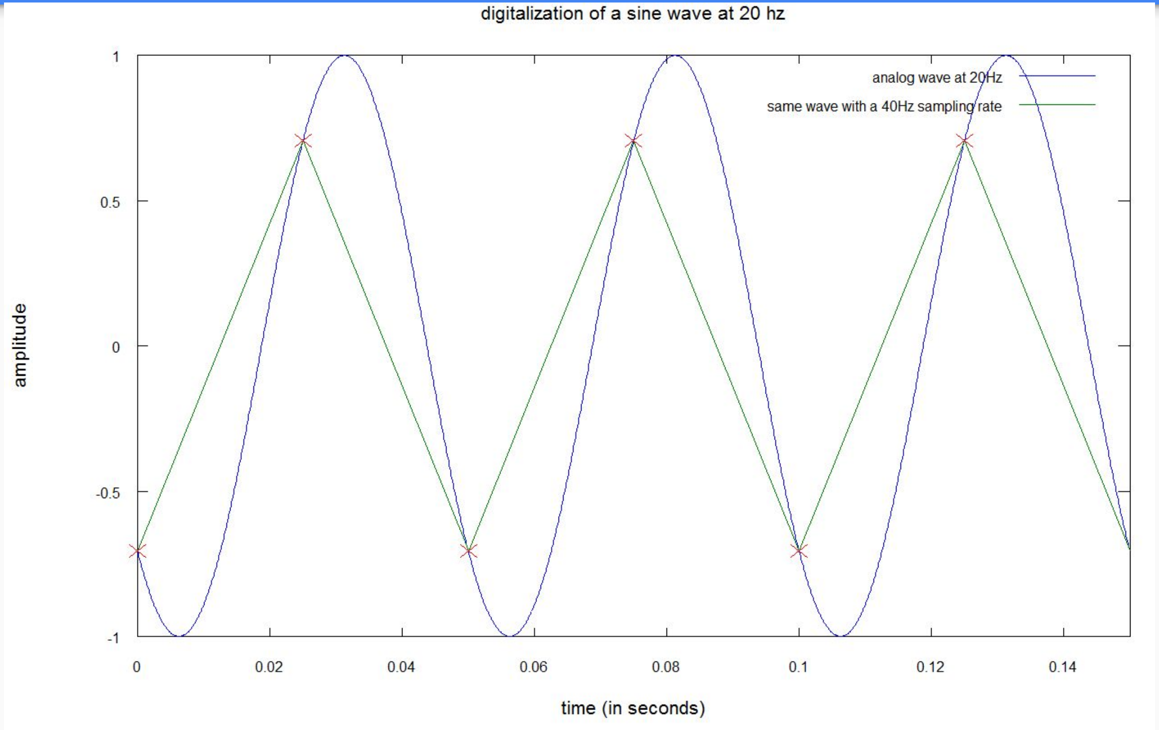


- a pure sinewave of frequency 20hz and amplitude 1
- a pure sinewave of frequency 40hz and amplitude 2
- a pure sinewave of frequency 80hz and amplitude 1.5
- a pure sinewave of frequency 160hz and amplitude 1

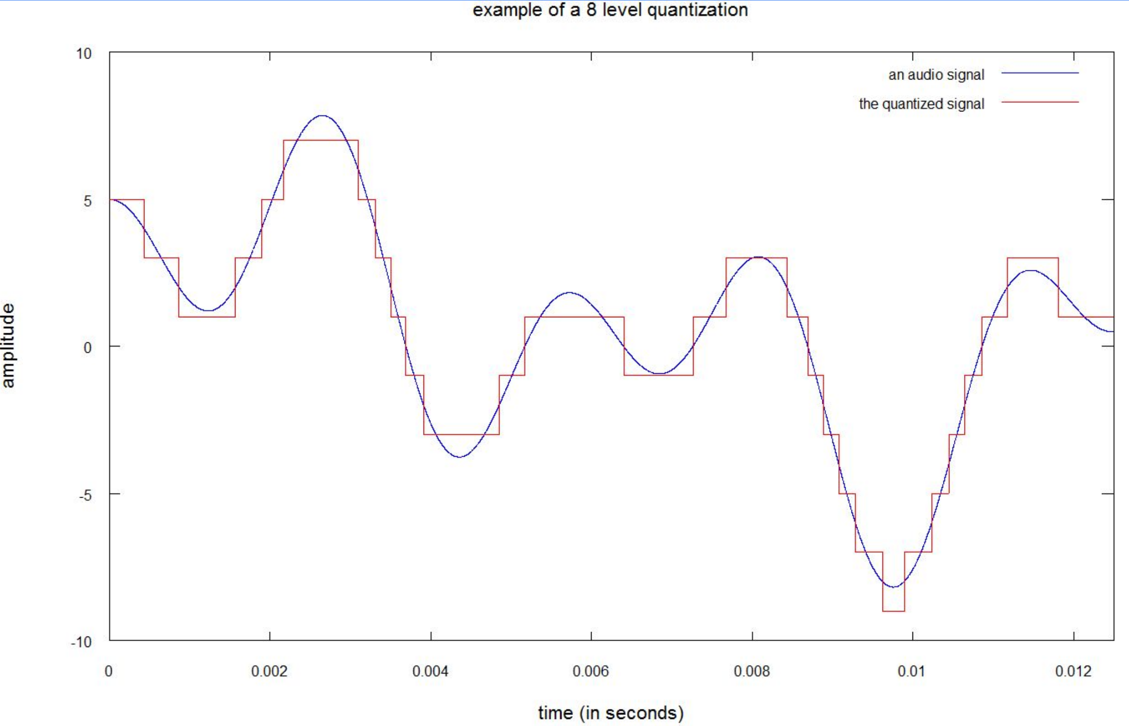
Basics of Audio
Spectrogram



Basics of Audio Digitization (Sampling)



Basics of Audio Digitization (Quantization)



- Loudness is the difference of amplitude b/w the lowest & highest point
- 8 level quantization requires 3 bits. 2^3 = 8
- 16 bits means 65535 levels of quantization

Basics of Audio
Fourier Transform

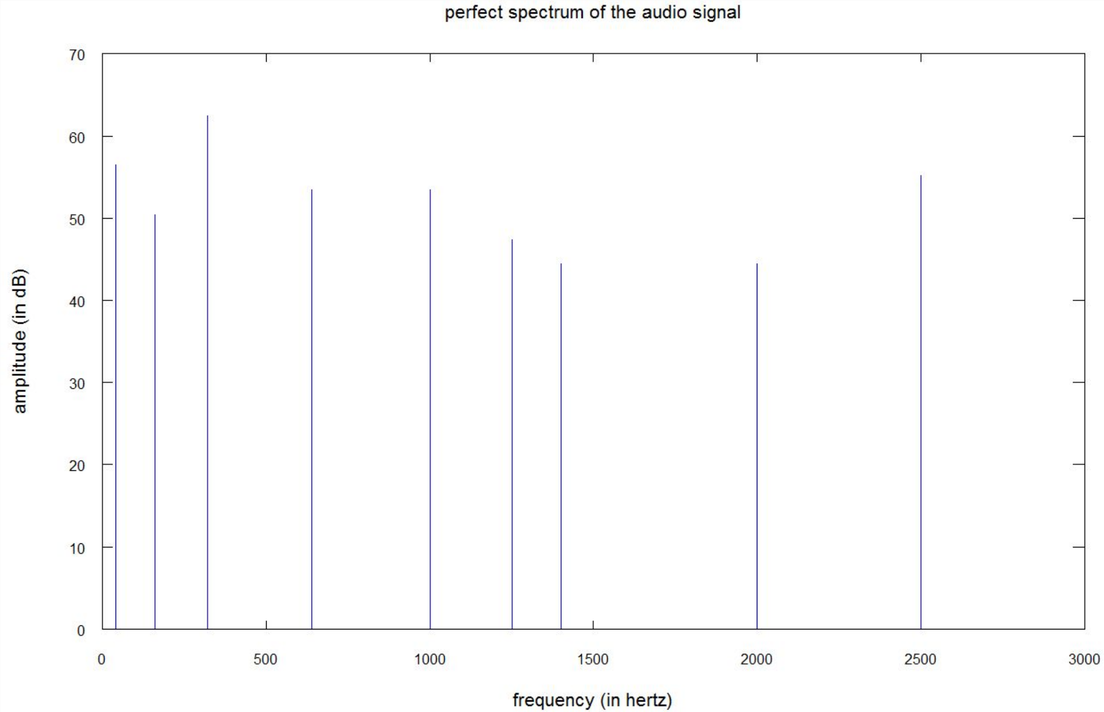
- Allows us to break the complex audio signal into it’s base frequencies.

$$X(n) = \sum_{k=0}^{N-1} x[k] e^{-j(2\pi kn/N)}$$

  - N: Size of the window.
  - X(n): nth bin of frequencies
  - x[k]: kth sample of audio signal

  - Discrete Fourier Transform: Compute Intensive. O(N^2)
  - Fast Fourier Transform: O(N*log(N))
  - Size of the frequency bin is called “Frequency Resolution”

Basics of Audio
Fourier Transform

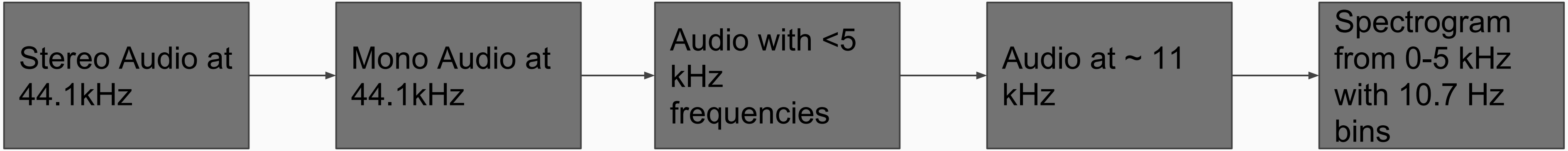


Shazam

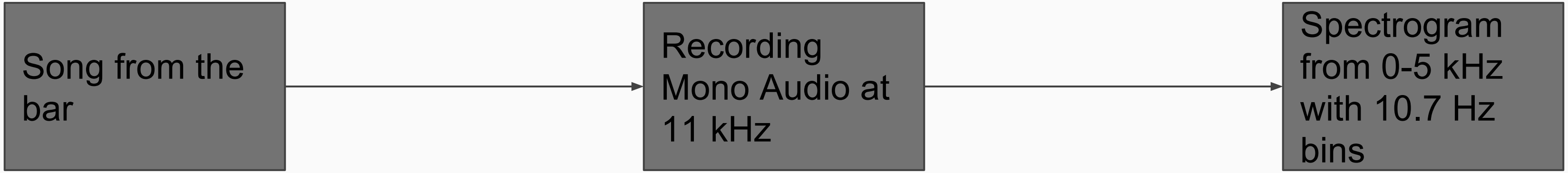
- Properties of the Fingerprinting Algorithm
  - Noise / Fault Tolerant
  - Time Invariant
  - Fast
  - Fewer false positives

Shazam

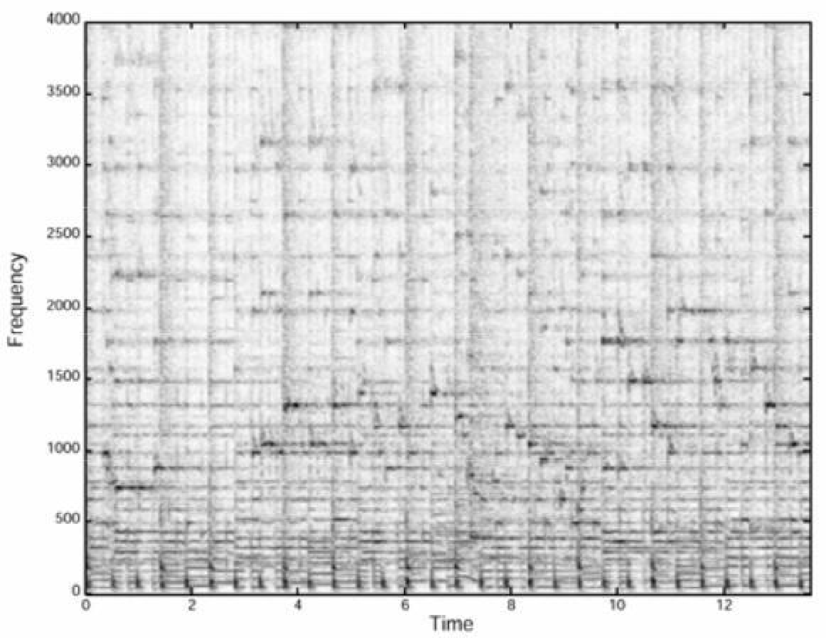
Server Side



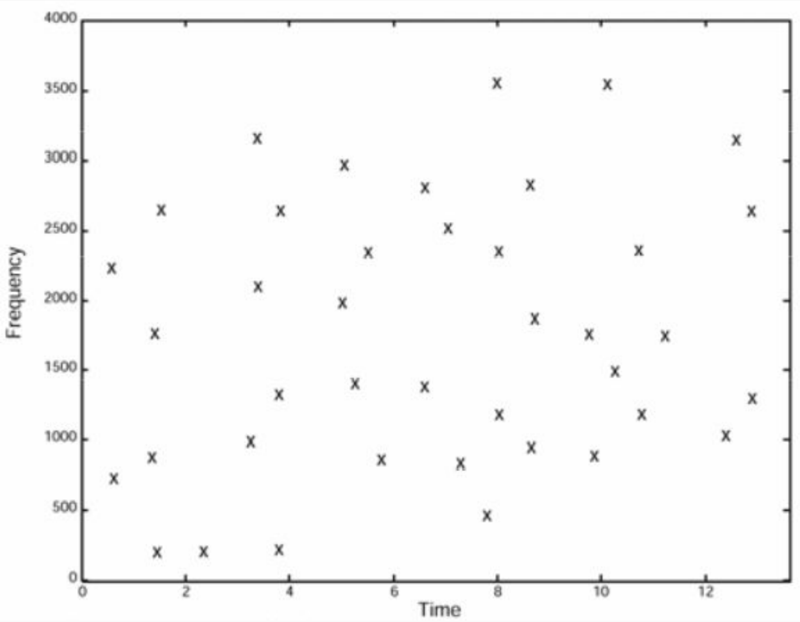
Client Side



Shazam



Frequency
Time
4000
3500
3000
2500
2000
1500
1000
500
0
0
2
4
6
8
10
12



Frequency
Time
4000
3500
3000
2500
2000
1500
1000
500
0
0
2
4
6
8
10
12

Shazam
Hash Calculation

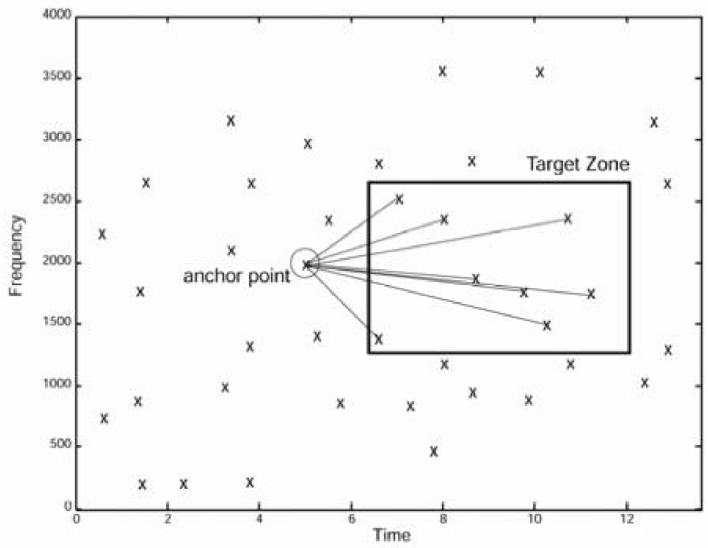


Fig. 1C - Combinatorial Hash Generation

Get an anchor point

Get a target zone

Create a hash using the delta(t) and the frequencies

Shazam
Hash Calculation

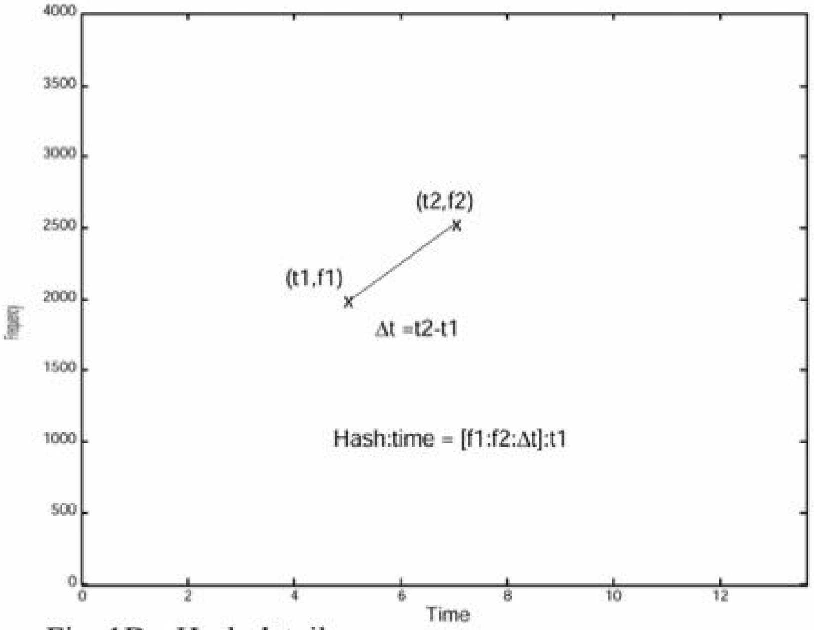


Fig. 1D - Hash details

Get an anchor point

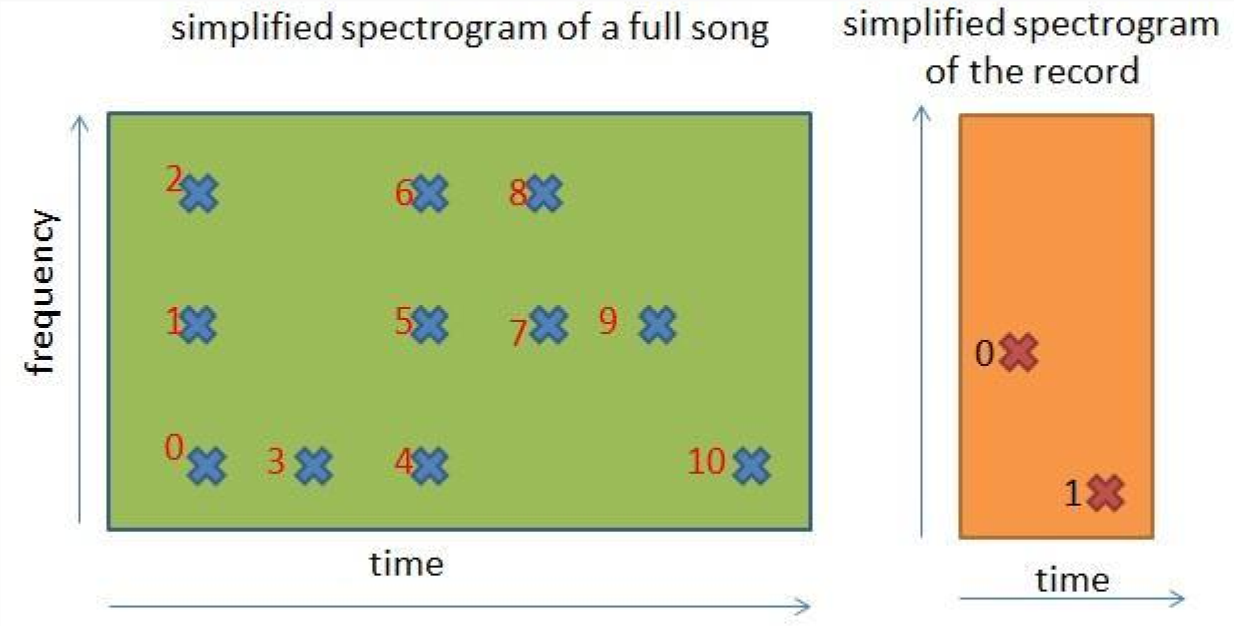
Get a target zone

Create a hash using the delta(t) and the frequencies

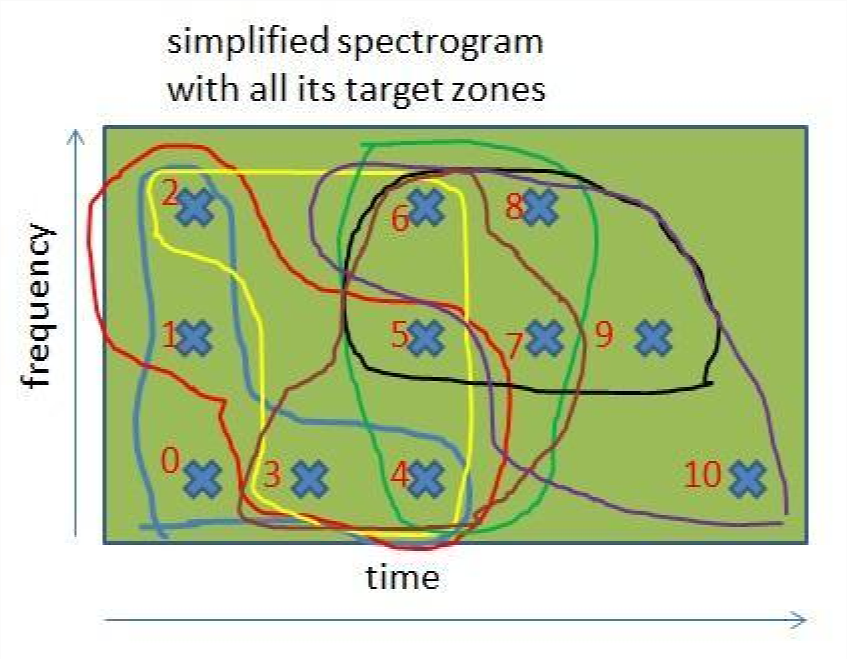
Hash h = f1:f2:d(t)

Inverted Index: h -> [abs(t), song_id]

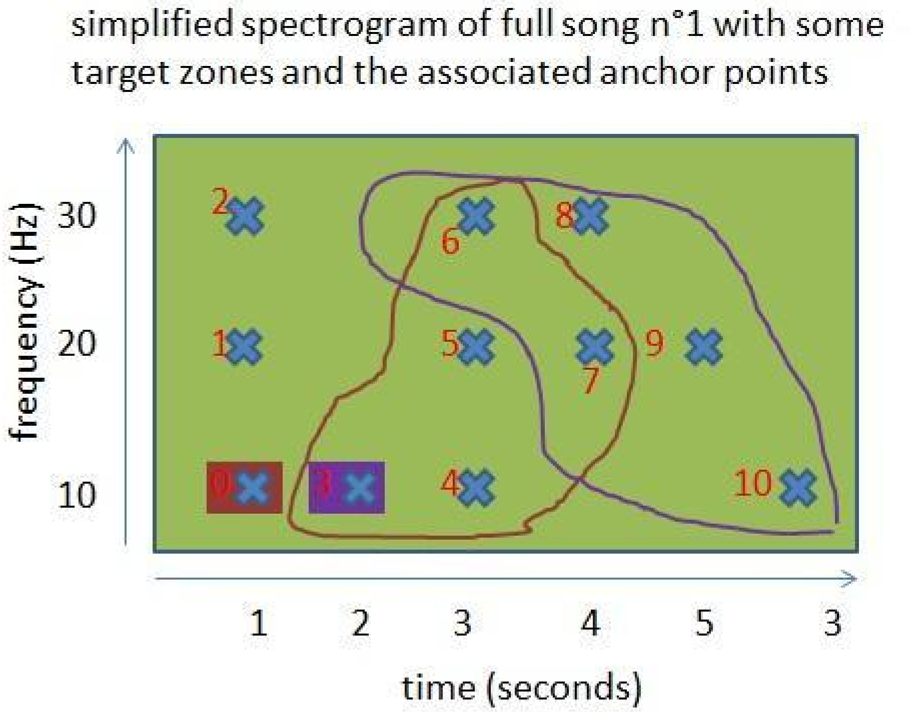
Shazam (Another look)



Shazam (Another look)



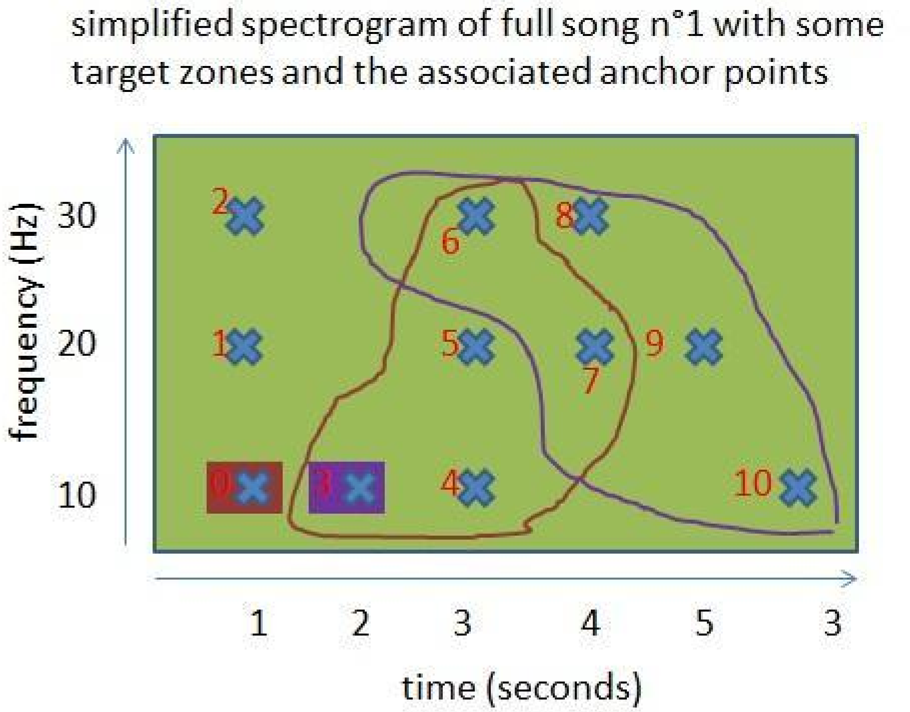
Shazam
Address Generation



Anchor point is 3 points before the targett zone

It can be anywhere as long as the algo is the same on server & client side

Shazam
Address Generation



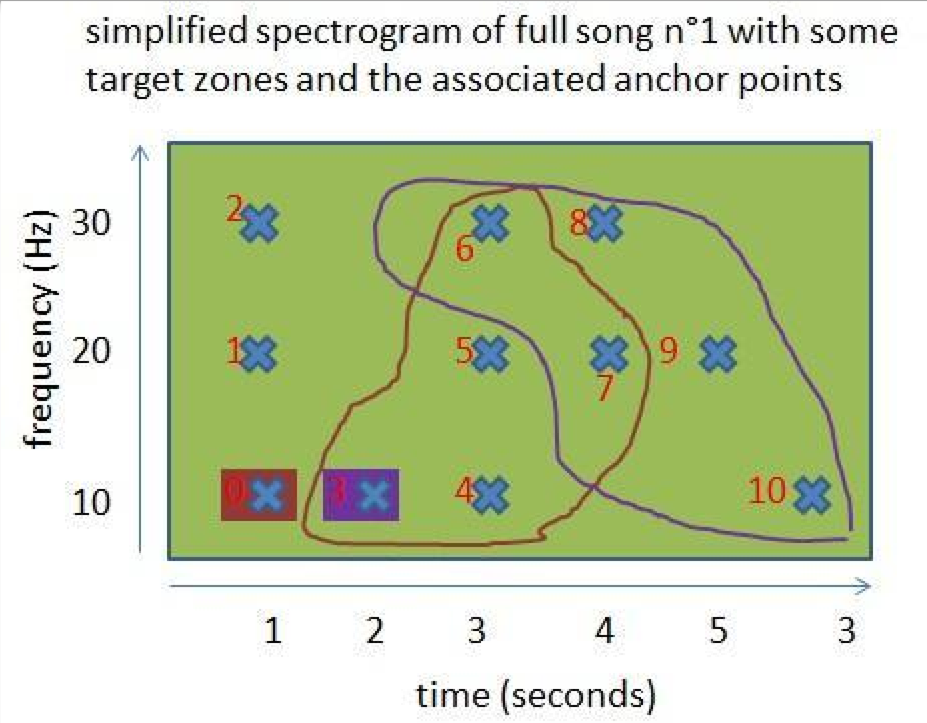
For purple target zone:

Address of point 6: [Freq of 3, Freq of 6, delta(time)] -> [abs(t), song_id]

Addr(6): [10,30,1] -> [2, 1]

Addr(7): [10,20,2] -> [2, 1]

Shazam
Address Generation



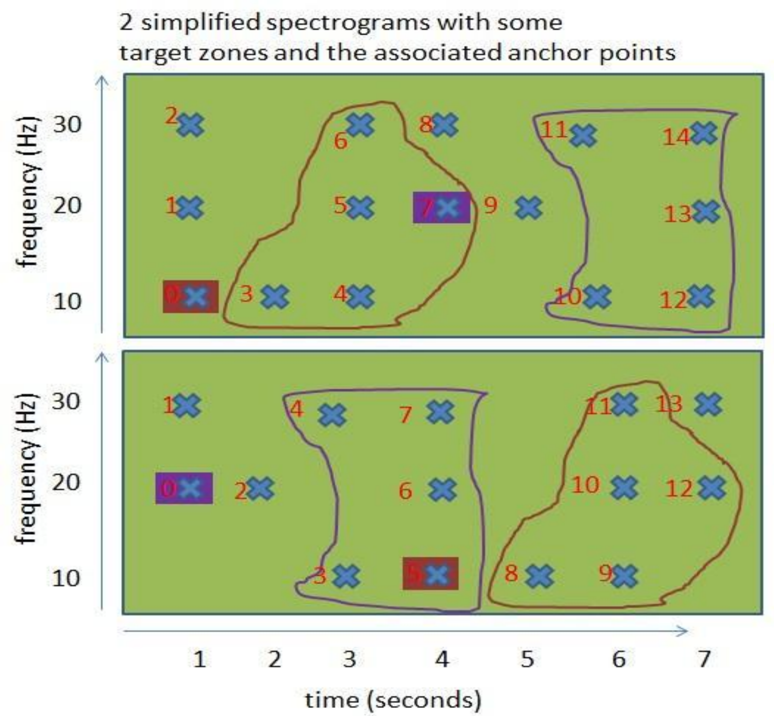
For red target zone:

Address of point 6: [Freq of 0, Freq of 6, delta(time)] -> [abs(t), song_id]

Addr(6): [10,30,2] -> [1, 1]

Addr(7): [10,20,3] -> [1, 1]

Shazam
Time Ordering



For each address, compute delta of time from recording & original song.

delta(t) = abs(time in song) - abs(time in record)

Add this to list of deltas and count the distinct(deltas)

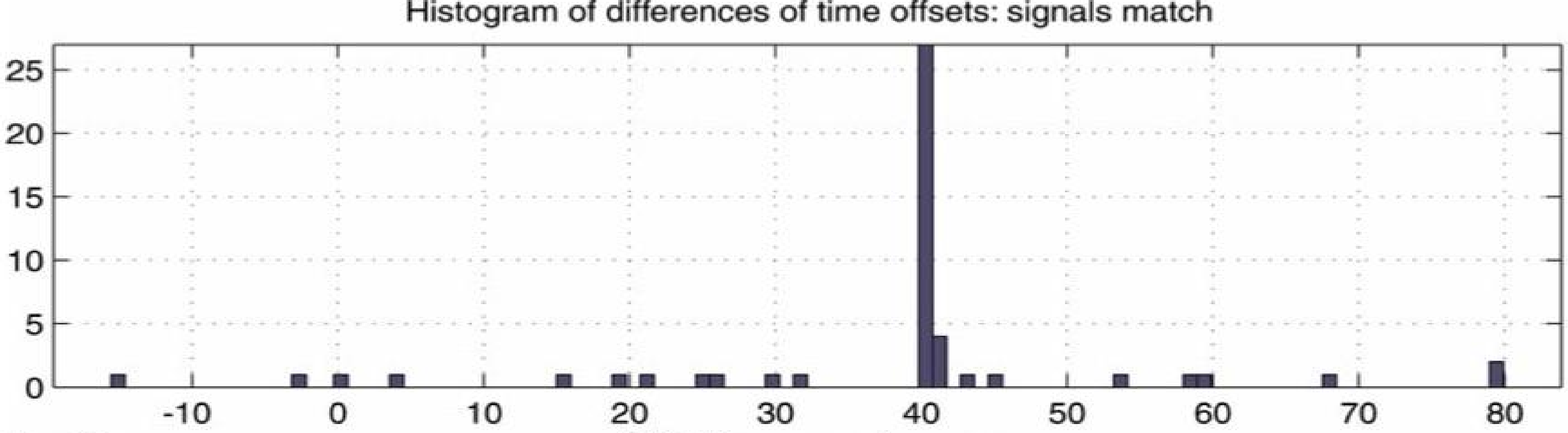

Histogram of differences of time offsets: signals match
25
20
15
10
5
0
-10
0
10
20
30
40
50
60
70
80
Offset $t_{database}$-t_{sample}

Fig. 3B

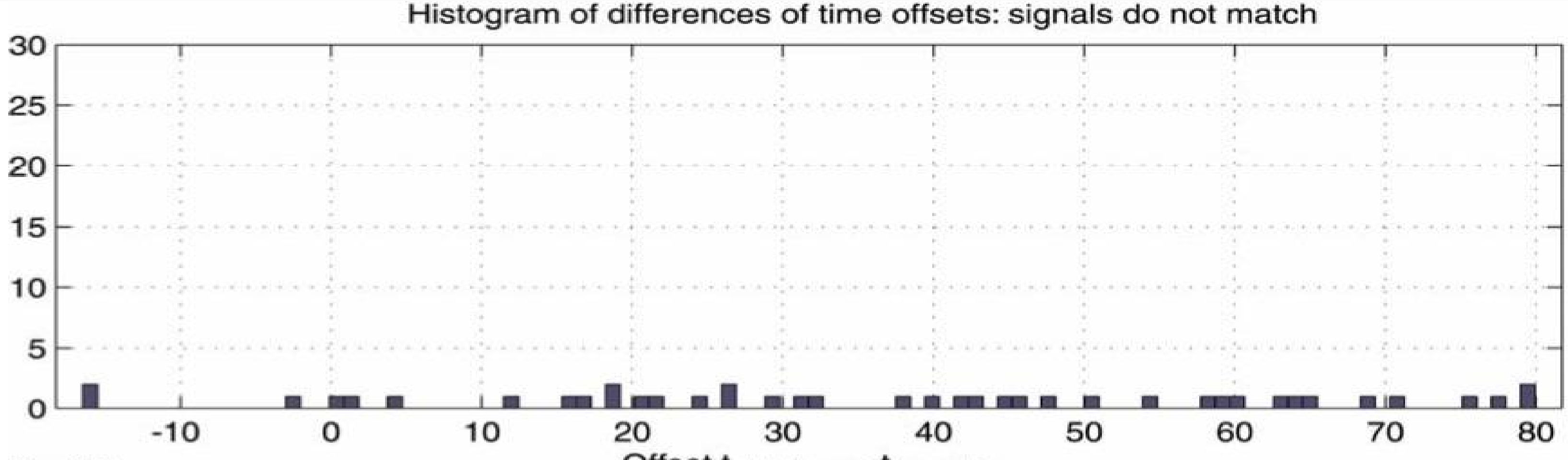

Histogram of differences of time offsets: signals do not match
30
25
20
15
10
5
0
-10
0
10
20
30
40
50
60
70
80
Offset $t_{database}$-t_{sample}

Fig. 2B

Shazam
Time Ordering (Another Look)

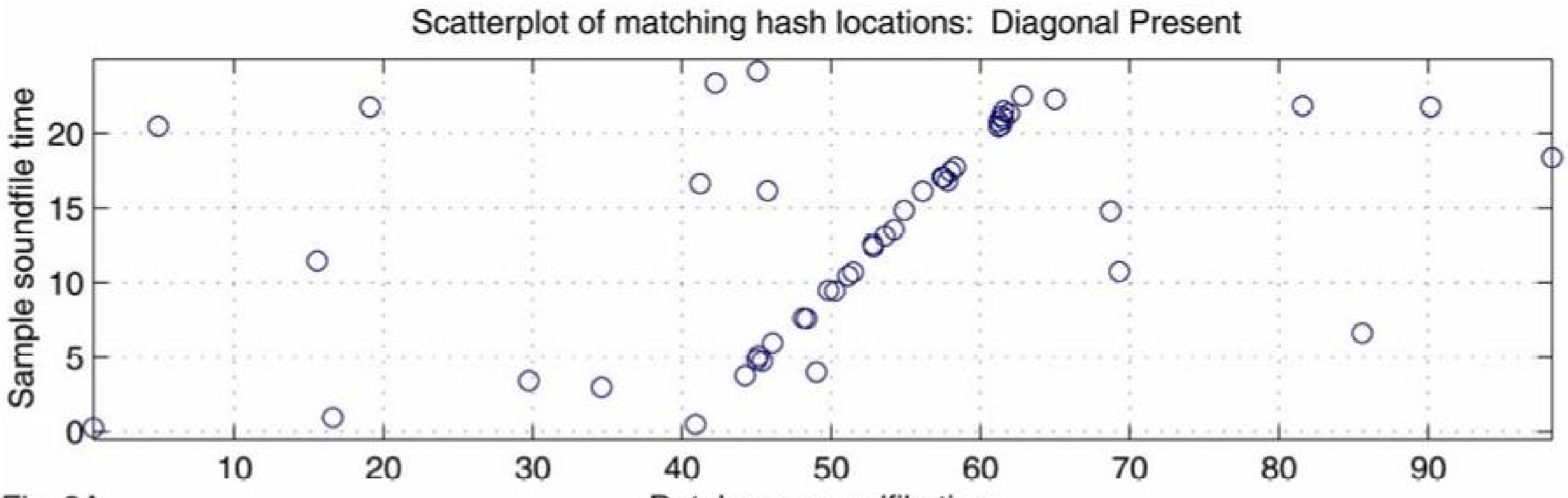


Fig. 3A

Shazam Complexity

- Assumptions:
  - FFT with 1024 samples. So there are 512 possible frequencies
  - On avg, a song contains 30 peak frequencies / second
  - So a 10 s recording contains 300 peaks
  - S it total # of songs in the database
  - Size is target zone = 5
  - Assume that the delta(t) b/w anchor point and target point is either 0 or 10 ms
  - Assume uniform distribution of frequencies

Shazam Complexity

- Space Complexity of Data:
  - 512 frequencies can be coded in 9 bits (2^9 = 512)
  - 9 bits for anchor point freq + 9 bits for target point freq
  - Assuming a 16 sec recording, we have 14 bits for the time field (in ms): 2 ^ 14 = 16384
  - Total 32 bits for the key & 64 bits for the absolute time + song ID

Possible Improvements?

- Distributed datasets with sharding
- Use 3 anchor points and encode the result in a 64 bit integer