

# Notes

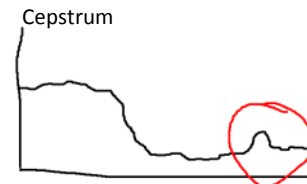
Friday, March 13, 2009  
11:02 AM

Friday, March 13, 2009  
3/16/2009, 10:55 AM

## CH 4

### Properties of Fourier Transform

- a. High frequencies will give you jagged edges
- b. Spectral leakage
  - i. The non-original frequencies showing up in the frequency
- c. Soften out the leakage
  - i. You use different windows to soften
  - ii. The wave will be more gradual
  - iii. You will start at 0 and end at 0
- d. Frequency resolution
  - i. 100Hz wave form
  - ii. Sampling rate/samples per block = frequency resolution
  - iii.  $8000/512 = 15.625$  frequency resolution
  - iv.  $8000/4096 = 1.9531$  frequency resolution
    - 1) Good frequency resolution
  - v.  $8000/128 = 62.5$  frequency resolution
    - 1) Bad frequency resolution
  - vi. Then you can look at the energy at each of the frequencies
- e. Samples per block
  - i. This is suppose to be the Hz of the wave form
  - ii. If you give it an incorrect number
    - 1) You'll get most of the energy in the two surrounding blocks around the actual Hz of the wave form
- f. Pitch Tracking
  - i. Is rather difficult
  - ii. Time domain pitch tracking
    - 1) Look at one peak then the next peak and find the distance between those
    - 2) Sometimes harmonics mess with the fundamental frequency
  - iii. Frequency domain pitch tracking
    - 1) Use the Fourier transform and look at its spectrum
    - 2) Multiple peaks will show up
    - 3) Most likely the first peak is the fundamental frequency
  - iv. Cepstral domain
    - 1) The cepstrum is Fourier transform of the log power spectrum
    - 2) In your cepstrum you will get amplitude on y and quefrequency on x
    - 3) The fundamental frequency will appear as a blip in the lower part of the amp
  - v. Auto Correlation
    - 1) Autocorr.py
    - 2) Not always perfect either
      - a) You get more peaks at  $2 * \text{the period}$ ,  $3 * \text{the period}$ , etc
    - 3) You multiply each sample by its neighbor from a distance
      - a) Then you sum up these values
    - 4) If you get them exactly at a period apart there will be a big peak
      - a) This helps you find the period



### Speech Analysis

- a. Spectrogram
  - i. Amplitude: Darker bands are louder
  - ii. Frequency on vertical scale
  - iii. Time as it goes
  - iv. For noise characters there is not banding
    - 1) Example: sound of t
  - v. Banding shows up better if you have good frequency resolution
  - vi. Time resolution is better for smaller blocks

3/18/2009, 11:01 AM

## CH 4

### Aliased Frequencies due to under sampling

- a. Cannot represent above the Nyquist frequency
- b. Mirror around Nyquist frequency until you're at sampling rate
- c. Over sampling rate just repeats on lower ones
  - i. 9000 Hz is like 1000 Hz
- d. Aliased frequencies are not discernable from valid frequencies
- e. We have a pre filter that prevents aliased frequencies from under sampling
  - i. Pre filter is the Nyquist frequency or right below that
- f. Telephone system
  - i. Samples at 8kHz
  - ii. Pre filter cuts out everything above 3.3kHz
- g. Example
  - i. 8000 sampling rate
    - 1) Mirror around 4000 (nyquist frequency)
  - ii. Create a tone at 1000 Hz for 1 second
  - iii. Create a tone at 7000 Hz for 1 second
  - iv. Play each, they sound the same
    - 1) 1000 Hz aliases to 7000 Hz for 8000 sampling rate
  - v. Create a tone at 6000 Hz for 1 second
  - vi. Create a tone at 2000 Hz for 1 second
  - vii. Play each, they sound the same again
    - 1) 2000 Hz aliases to 6000 Hz for 8000 sampling rate
  - viii. At 8000 Hz and 4000 Hz
    - 1) Come out as 0
  - ix. 9000 Hz is like 1000 Hz

### Dithering

- a. Interactive tutorial
- b. Adds noise to the signal to make signal sound better

### Noise shaping

- a. There will be some noise in the signal
- b. Shape the noise to a higher frequency where we hear less
- c. It uses a feedback loop during quantization to do this
- d. Example in book
- e. Example
  - i. POW-R
    - 1) Spreads error to 9 samples

### Dynamic range compression

- a. Popular music dynamics are compressed
- b. Makes the softs not as soft and the louds not as loud

### Non-linear quantization

- a. Our ears are not sensitive to the amount of noise but the amount of noise compared to signal
- b. Normal quantization all levels are spaced equally
- c. Non-linear quantization uses more levels for lower amplitude signals
- d. Usually logarithmic
- e. Non-linear Companding
  - i. Mu-law
    - 1) Used in US and Japan
    - 2)  $\text{Sign}(x) \cdot (\ln(1 + \mu |x|)) / (\ln(1 + \mu))$
    - 3) Telephone system
      - a) 8kHz 8 bits/sample mu-law
      - b) Roughly same quality as 12 bit linear
    - 4) CD doesn't use it, they say they don't need it they have enough levels
  - ii. A-Law
    - 1) Used in Europe
- f. Book has examples

3/20/2009, 10:53 AM

### Generate a sine wave

- a. sinegen.py
- b. Uses sine
- c. Needed for comparison for homework exercise 2

- d. Scale factor changes the dynamic range
- e. 440 concert A
  - S440 = makeSine(440, 8000, 1000, 32767)
  - S440.play()
  - MediaTools->Examine Sound

#### MIDI

- a. Standard, developed in 1983
- b. Protocol
  - i. Serial protocol @ 31.25 kb/sec
  - ii. Used for communication
  - iii. Specifies musical events
    - 1) Musical event
      - a) Note: must specify pitch, duration (note on, note off), volume
- c. Channel / Track System
  - i. Each instrument needs a channel if they're going to play things at the same time
- d. General MIDI 1
  - i. Specifies instruments so you have the same instruments on the same patches no matter where you are
  - ii. MIDI 2 added character info and some more stuff
- e. Notes
  - i. Each note has a number
  - ii. Middle C (C4) is 60
  - iii. C5 is 72
  - iv. Up 12 for each octave!
- f. Music
  - i. If you have 440 and want to go up an octave you get 880
  - ii. 8 scale tones between an octave
  - iii. 12 notes/semitones in an octave
  - iv. Major Scale
    - 1) Whole semitone whole whole semitone
  - v. Just intonation
    - 1) 2/1 is an octave
    - 2) 5/4 major third
    - 3) 3/2 fifth
    - 4) If you go outside your key it messes up
  - vi. Equal temperament system
    - 1) Slightly out of tune
    - 2) All scales and key will be equally out of tune
    - 3) To get this even spacing we use
      - a)  $440 * 12\text{th root of } 2$ , 12 times you'll get 880
      - b)  $220 * 12\text{th root of } 2$ , 12 times is 440
      - c) 12th root of 2 is 1.059
- g. Organization
  - i. Uses Messages
    - 1) Each message has a status byte and sometimes data bytes
    - 2) Most significant bit is set in status byte
      - a) Most Significant Bits of is the message
      - b) Least Significant Bits tell you what channel to apply this to
      - c) This allows 16 channels
      - d) Lowest: 128
      - e) Highest: 255
      - f) Running status
        - i) One status bytes with multiple data bytes
    - 3) Most significant bit is not in the data bytes
      - a) Lowest: 0
      - b) Highest: 127
  - ii. Channel Messages
    - 1) Chords
    - 2) Notes
    - 3) etc
  - iii. System Messages
    - 1) Related to timing

#### Distinguishing audio formats

- a. How are the samples encoded
  - i. Linear or log quantization
  - ii. Signed or unsigned
- b. File format
- c. Byte order
  - i. Little Indian or big Indian
    - 1) Big Indian MSB first then LSB
    - 2) Little Indian LSB first then MSB
    - 3) Wave uses little Indian
- d. Are the channels interleaved?
  - i. Ch1 sample, ch2 sample, ch1 sample, ch2 sample, etc
  - ii. Interleaving keeps the samples for each time period together
  - iii. Non interleaved keeps all the samples for each channel together
- e. File constraints
  - i. Sampling rate constraints
  - ii. Bit depth constraints
  - iii. Number of channels
  - iv. Meta info in the file
- f. Compressed or uncompressed data
  - i. Lossy or lossless compression if applicable
- g. Open or proprietary
- h. Table of different audio formats in table 5.1

Wave file dissection <http://ccrma.stanford.edu/courses/422/projects/WaveFormat/>

#### IFF (Interchange File Format)

- a. EA created in 1985
- b. Represents data in chunks

#### AIFF

- a. Apple created in 1988
- b. Uses big Indian byte order

#### RIFF

- a. Microsoft created in 1991
- b. Same as AIFF but uses little Indian byte order

#### Speech

- a. Source-filter model
  - i. Source (vocal cords, noise generators: lips, etc)->filter->output
    - 1) Glottis is the space between your vocal source
  - ii. Vocal tract
    - 1) From larynx up to lips
    - 2) Your jaw, mouth act as a filter
  - iii. When leaving the vocal tract to air
    - 1) There is a impedance mismatch
    - 2) Size of air is much greater than size of mouth
    - 3) Creates a 6 dB/octave filter
      - a) Causing a 6 dB/octave roll off
    - 4) Difference the samples to counter act this
      - a) Done for most speech applications

4/1/2009, 9:49 AM

I/O functions in JES are listed in JES Functions

```
n = requestInteger("please type a number")
requestNumber is a float
```

#### Audio Coding

- a. Must consider if it causes a delay
- b. Differentiation is a high pass filter
- c. Delta modulation
  - i. Very simple
  - ii. 1 bit per sample
  - iii. Fixed set size
  - iv. Each bit tells it to go up or down
  - v. Some slopes will cause slope overload and distortion

- vi. Granular noise occurs during silence, since you cant specify no change only up and down
- vii. Less granular noise, smaller steps
- viii. Better slope overload resistance, larger steps
- ix. Good for just intelligible speech
- d. ADM (adaptive delta modulation)
  - i. Same as above but adaptive changes the step size
  - ii. You would also want a min and max
  - iii. Example
    - 1)  $M = \{z, e(n) = e(n-1)$   
 $\{y, e(n) \neq e(n-1)$
- e. (CVSD) Continuously Variable Slope Delta Modulation
  - i. Commonly used
  - ii.  $s(n)$  is the step size
  - iii.  $s(n) = \begin{cases} ks(n-1)+P, & e(n)=e(n-1)=e(n-2) \\ ks(n-1)+Q, & \text{otherwise} \end{cases}$   
 $P$  is a large constant to allow for overloads  
 $K$  allows or slows the response of the system to change in amplitude
  - iv. 40kbps = log PCM 8kHz 8 bits/sample, 64 kbps
  - v. 32kbps communication quality at 16kbps
- f. Linear or Log PCM (Pulse Code Modulation)
  - i. 16 bits per sample for high quality
  - ii. 12 bits per sample linear 8 bits log for pretty good quality
  - iii. Works fine for real time systems
- g. ACPM (Syllabic Companding)
  - i. Adaptively changes the quantization of the audio based on the energy in a block of audio
  - ii. Finding the energy of the current block causes a problem for real time communication
    - 1) We use the previous blocks energy for the current to get around this
- h. DPCM (Digital pulse code modulation)
  - i. You encode the differences between the samples
- i. ADPCM (2 types: adaptive digital code modulation and adaptive prediction)
  - i. Used very often
  - ii. Adaptive prediction
    - 1) Explicitly says that the samples are correlated closely
    - 2) Predicts one sample ahead
    - 3) Uses  $e(n)=x(n)-ax(n-1)$  for error signal
      - a)  $A$  is adaptive, calculated every 10 to 20 ms
      - b) Error is also calculated every 10 to 20 ms

4/3/2009, 10:06 AM

Diff.py is an example of dpcm

- a. Left first sound to get back to original sound
- b. Diffsnd differentiates the sounds
- c. Sounds are more hollow kinda like telephone speech, also lowers the volume, seems to increase noise as well

Filters

- a. Low pass
  - i. Averages samples
  - ii. Smoothing
  - iii. Muffles with sound
- b. High pass
  - i. Differentiates samples
  - ii. Can increase noise

Infinite peak clipping

- a. If the sample is positive, set to 10,000
- b. If the sample is negative, set it to -10,000

Simple Encryption

- a. Frequency inversion
- b. Invert the sign of every other sample

Example in folder

4/6/2009, 10:11 AM

Filter

- a. Uses
  - i. Modify the frequency ( spectrum ) of a sound
  - ii. Alter phase response
    - 1) All-pass filter
  - iii. Both
- b. Order
  - i. How many samples you are using
- c. Types
  - i. Low pass
  - ii. High pass
  - iii. Band pass
  - iv. Notch
  - v. Causal filter
    - 1) Depends on present and past input and past output
  - vi. Non-causal filter
    - 1) Can look ahead
    - 2) Cannot operate in real time
- d. Filter Equation
  - i.  $y(n) = \text{summation from } i=0 \text{ to } m(a_i \cdot x_{n-i}) - \text{summation from } i=1 \text{ to } N(b_i \cdot y_{n-i})$
- e. Stable vs. unstable
  - i. Stable filter: output goes to 0 after input goes to 0
  - ii. Unstable filter: output may grow without bound
- f. Impulse response
  - i. Response of a filter to a single pulse of sound
  - ii. Example:  $1/8x(n)+7/8y(n-1)$

| N time | x(n) | y(n-1)              | y(n)                        |
|--------|------|---------------------|-----------------------------|
| -10    | 0    | 0                   | 0                           |
| -1     | 0    | 0                   | 0                           |
| 0      | 1    | 0                   | 1/8                         |
| 1      | 0    | 1/8                 | 7/8*<br>1/8                 |
| 2      | 0    | 7/8*<br>1/8         | 7/8*<br>7/8*<br>1/8         |
| 3      | 0    | 7/8*<br>7/8*<br>1/8 | 7/8*<br>7/8*<br>7/8*<br>1/8 |

And so on...

- iii. The above is an IIR (infinite impulse response),
  - 1) It will never go to zero
  - 2) If 7/8s was greater than one, we would get an unstable filter
  - 3) IIR gets equivalent or better results than FIR with fewer coefficients
  - 4) Any analog filter can be implemented as IIR
  - 5) Butterworth filter
    - a) Maximum flat frequency response at low frequencies
  - 6) Bessel filter
    - a) Maximum linear phase response at low frequencies
  - 7) Chebyshev filter
    - a) Equal error above and below pass band
    - b) Equiripple criterion
  - 8) Elliptic filter
- iv. FIR (finite impulse response)
  - 1) If depend only on present and past input will go to zero
  - 2) All b coefficients are zero
  - 3) Can have linear phase response
  - 4) Cannot be unstable

|                     | IIR    | FIR      |
|---------------------|--------|----------|
| No. of coefficients | Low    | High     |
| Sensitivity         | Can be | Low (16) |

|                                     |                                |                  |
|-------------------------------------|--------------------------------|------------------|
| ity to<br>coeff<br>quantiz<br>ation | high<br>(24<br>bits/co<br>eff) | bits/co<br>eff)  |
| Probabi<br>lity of<br>overflo<br>w  | Can be<br>high                 | None             |
| Stabilit<br>y                       | Must<br>be<br>designe<br>d in  | Always<br>stable |

4/8/2009, 10:00 AM

Information Theory

- a. Developed by Claude Shannon
- b. Tells us the most efficient way to encode signals
- c. Encoding groups of symbols is always better than individual symbols
  - i. This uses the idea of entropy
  - ii. If it is completely random, it cannot be compressed
- d. Vector quantization
  - i. We can vector quantize almost ANYTHING
  - ii. Tree based vector quantization
    - 1) Eliminates stuff you go down the tree
  - iii. Multistage vector quantization
    - 1) 2 or more code books
    - 2) Original result of the going through the first codebook is encoded by the second codebook
    - 3) You must send two numbers then, the first codebook value and second codebook value
  - iv. Available Gain/Shape vector quantization
    - 1) Use different code books based on the RMS
    - 2) Easy to use for different or voice sounds
  - v. Bit rate
    - 1)  $R = \log_2 N/K$
    - 2)  $N$  = number of vectors
    - 3)  $K$  = number of samples in the vector
  - vi. Adaptive Code book filtering
    - 1) Making a code book
    - a) Create
      - i) <http://falstad.com/dfilter/>
      - ii) Create random divisions
      - iii) Create one average vector of training data
        - o Then split the vector
    - b) Training sample
      - i) Encode training sample
    - c) Measure distortion
      - i) f distance
        - o Requires more computation
      - ii) City Block: summation of absolute value of differences
    - d) If distortion is less than some threshold we decide we stop
    - e) Else we replace each codebook vector with the centroid of the input vectors that matched it
  - vii. Encoder uses codebook to match to the closest vector
    - 1) We must decide size of codebook and how long each vector is
    - 2) We will use 128 bit codebook 6 samples per vector
  - viii. Decoder needs codebook too
    - 1) Takes the received value looks it up in codebook and assigns the sample value
  - ix. Example:
    - 1) Sample (64) -> Encoder Codebook (8) -> Decoder Codebook (8) -> Sample (64)
- e. Wave form coder
  - i. Just tries to represent this sequence
- f. Source coder
  - i. Knows how the signal was produced
  - ii. Uses the above knowledge to help represent the data
  - iii. ADPCM
    - 1) Look at past samples to predict future samples
      - a)  $x(n) = \text{summation from } k=1 \text{ to } k(a_k * x[n-k] + e(n))$
      - b) Error signal is e
  - iv. Linear predictive coding

- 1) Much like ADPCM
  - 2)  $y(n) = \text{summation from } k=1 \text{ to } k(a_k * k[n-k])$
  - 3) No error codes, not exactly encoding the signal
  - 4) Difference between sample we predicted and actual sample
    - a) Residual
    - b) Distortion
    - c) Error
  - 5) Models a vocal tract filter function
    - a) Because our vocal tract cant move instantaneously, we can predict sameness below that time
    - b) We can calculate once and use that value once for a certain length of time
  - 6) Transmitter
    - a) Sends
      - i) Coefficients
      - ii) Frequency
      - iii) Voiced/unvoiced
      - iv) Gain
    - b) Determine voiced or unvoiced
      - i) Amplitude
      - ii) Zero crossings: where the signal goes to 0
        - o Count these
        - o Unvoiced will have a lot
        - o Voiced will have a few
      - iii) Spectral energy
        - o Some consonants have high frequency
    - c) Tracking frequency
      - i) Time domain
      - ii) Look for peaks in residual
    - d) Gain
      - i) Just check the last 10 to 25ms
    - e) Another thing we could do is
      - i) Send the residual through the filter
- v. RELP
- 1) Residual excited linear prediction
  - 2) Down to 9600 bps
- vi. KELP
- 1) Uses a code book
  - 2) Down to 2400 bps
  - 3) 10-12 predictors is good for speech
  - 4) Used by
    - a) Speex
    - b) LPC-10: Uses 10 predictors

4/13/2009, 9:58 AM

#### Source coders

- a. Uses knowledge about the process that created the waveform

#### Waveform coders

- a. Purely encode the waveform

#### Vocoders (voice coders)

- a. Sub-band coding
  - i. Takes the observation that any complex wave can be modeled as a summation of sine waves
  - ii. Breaks it up into frequency bands
  - iii. Uses less bits for lower frequency and more for higher frequency
  - iv. Input -> filterbank (16-32 filters) -> band pass filters [1, 2, 3, ... n] -> decimate -> ADPCM -> receiver -> decodes ADPCM -> back through band pass filters [1, 2, 3, ... n] -> combine into output
- b. Perceptual coding
  - i. Uses psychoacoustics
    - 1) Study of how people perceive sound
    - 2) Involvement of ear and brain
  - ii. Exploits how people hear acoustic elements
  - iii. What do we know?
    - 1) Hearing is non linear
      - a) In amplitude (dB) algorithmic
      - b) In frequency (octave)

- 2) Most acute between 1kHz-5kHz range
  - a) 100 Hz and 1kHz sound
    - i) More power must be on 100 Hz sound to make it sound the same as 1kHz
    - ii) Sone is based off this idea
- 3) Our hearing operates in terms of critical bands
  - a) If two sounds are in the same critical band, the difference is pretty much the same
  - b) Linear up to 500 Hz
  - c) About 100Hz wide
  - d) They get wider after 500 Hz
  - e) If we have a 400Hz and 450Hz tone and one is louder than the other, we will only hear the louder one
  - f) So if a tone is masked, we can throw it away
- iv. If we can't hear it, throw it away
- v. Perceptual coding is used by: MP3, OggVorbis, AC3 Dolby

#### Mpeg-1

- a. 1991
- b. 3 audio layers
- c. Mp3 is the third layer of mpeg1
  - i. Cd quality sound
  - ii. Uses a modified discrete cosine transform

#### Mpeg-2

- a. 1994
- b. Surround sound
  - i. Multichannel surround sound

#### Mpeg-4

- a. 1998
- b. Supports mixed audio
  - i. Voice
  - ii. Midi
- c. Very high quality sound

#### Mpeg-7

- a. 2003
- b. More general multimedia framework
- c. Supports searching and filtering of multimedia data

#### MP3 and perceptual encoding

- a. Doesn't specify encoding/compressing algorithm
- b. Specifies how it must look to the decoder
- c. Near cd quality is 11:1 compression
- d. We can specify mono or stereo
- e. Steps
  - i. Divide audio into frames
    - 1) 384, 576, 1152 samples
  - ii. Put through filterbank
    - 1) 32 bands/filters
    - 2) Layers 1 and 2 use equal sized bands
    - 3) Layer 3 has variable bands
      - a) These bands more or less match critical bands
  - iii. Do an FFT on each band
    - 1) 512 or 1024 samples
  - iv. Calculate masking curve for each band
    - 1) Help decide what to throw away
    - 2) Finds floor for quantization noise
  - v. Analyze tonal and non tonal elements in each band
  - vi. Determine interactions with neighboring bands
    - 1) Masking could occur between bands
  - vii. Determine bit depth needed for each band
  - viii. Quantize the fft outputs
    - 1) Possibly apply Huffman encoding

#### Mozer encoding

- a. We know our ear doesn't hear the shape, more the spectrum
- b. For every pitch period, it finds another wave with the same spectrum, and if the waveform repeats it can just say repeat and inverse

Speeding up a sound w/o messing with the pitch

- a. Cut out samples in blocks

4/15/2009, 9:56 AM

Synthesis

- a. You must decide:
  - i. Voice response
  - ii. Copy synthesis
    - 1) Recorded words of syllables
    - 2) Takes recorded speech and replays it strung together
  - iii. Text to speech
    - 1) Articulatory synthesis
      - a) Model speech tract  
Throat, tongue, etc
      - b) Send it pulse and the model will effect the sound
    - 2) Formant synthesis
      - a) Models the areas resonances
        - i) Example:  
One. /i/ (ee) 270Hz 2200Hz 3000Hz  
Two. /a/ (ah) 900Hz 1100Hz
      - b) 2-3 formants are usually used
      - c) Resonances are a little higher for women usually
      - d) Each person has slightly different formants
      - e) To recognize specific voices you might want to use 4-5 formants
  - iv. Parallel Synthesizer
    - 1) Good for voiced and unvoiced
    - 2) Bad for Zs and such
    - 3) Problems with phase relationships (can be cancelled)
  - v. Cascade Synthesizer
    - 1) Good vowel sounds
    - 2) Bad for frickitive and explosives
      - a) Bad for things made above lyrnx
  - vi. Stored speech-copy synthesis
    - 1) Relatively limited no. of words/phrases
    - 2) Word insertion/concatenations
    - 3) Example: 411 telling you a phone number
      - a) They also record based on position
        - i) Record 0-9 in beginning position, ending position, and internal position
  - vii. Languages
    - 1) English
      - a) Non tonal
      - b) Phonemes
        - i) 42 in English
        - ii) A vowel or consonant sound
    - 2) Chinese
      - a) Tonal
      - b) 12 tones
    - 3) Mandarin
      - a) Tonal
      - b) 4 tones
  - viii. Problems with synthesizers
    - 1) Abbreviations
    - 2) Digit Strings
    - 3) Proper Names
    - 4) Language irregularities
      - a) Cam from dutch
      - b) Came from old english/old norse
      - c) Cambridge from roman bridge over river Cam
      - d) Cameo from Italian
      - e) Magic e e rule
      - f) High frequency words are often exceptions
        - i) The
        - ii) There
        - iii) This vs thick
        - iv) Thin thimble

- v) Have vs shave
- vi) Share vs behave
- 5) Emphasis
  - a) Louder
  - b) Longer
  - c) Higher pitch
  - d) Suprasegmental
    - i) Travels over multiple words
- 6) Prosody
  - a) Harmony of speech
  - b) Hard to make with synthesis
- 7) Co articulation
  - a) Blending of sounds
  - b) Huge factor
  - c) Happens because our focal tract doesn't move instantaneously

#### Text to Speech

- a. Convert abbreviations, digit sequences, and special symbols to words
  - i. \$3.40
  - ii. St. Mark St.
  - iii. 1984
  - iv. 15%
- b. Identify morphemes and phrase and sentence boundaries, mark these boundaries
  - i. Morphological analysis
    - 1) Part of a word
    - 2) Ing is a morpheme
    - 3) Sing|ing
      - a) Have to not do s|ing|ing
    - 4) Looking for prefixes, roots, and suffixes
    - 5) New words always being created, new prefixes, suffixes not often
    - 6) Provides pronunciations information
      - a) Hot|house
      - b) Thesis no boundaries
    - 7) When we create word rules we don't apply those across morpheme boundaries
- c. Assign a duration and fundamental frequency (F0) to each phoneme
  - i. Use punctuation and syntax to estimate the appropriate intonation
- d. Generate a wave form
- e. A dictionary will be used for irregularities
- f. MITalk
  - i. Developed at MIT
  - ii. Breakthrough in synthesis
  - iii. Formant synthesizer
  - iv. Uses context window of 5 segments with 20 parameters
  - v. It's process:
    - 1) Symbols -> words
    - 2) Words -> phonemes
      - a) Dictionary of 12,00 morphemes
        - i) Spelling
        - ii) Part of speech
        - iii) Pronunciation
      - b) 95% are converted to morphemes
      - c) 5% handled by letter -> phonetic segment rules
    - 3) Determine Lexical stress
      - a) Stress on each morphemes
      - b) Considers effect of suffixes, compound words, sets stress marks, catches vowel changes
    - 4) Phonological recoding
      - a) Based on the phrase
      - b) Catches the
    - 5) Phrase level parsing
      - a) Assign a part of speech to each word
        - i) Aids prosody assignment
    - 6) Does Symantec analysis
      - a) Aids prosody assignment
      - b) Tries to find emphatic stress
    - 7) Timing analysis
      - a) Assigns durations for every segment
      - b) Prepausal lengthings

- 8) Pitch contour assignment
- 9) Assign phonetic targets
- 10) Smooths the target values
  - a) Parameters every 5 ms
- 11) Converts parameters to synthesizer coefficients
- 12) Generates waveform using sampling rate of 10kHz

4/17/2009, 10:04 AM

Speech Synthesizers history

<http://www.cs.indiana.edu/rhythmsp/ASA/highlights.html>

Speech Recognition

Power Point in Lectures folder

- a. Representation
  - i. Frames
  - ii. Amp vs. time
  - iii. Count zero crossings
  - iv. Spectral balance
    - 1) High vs. low frequency energy (or high-mid-low, etc)
  - v. Spectral detail
    - 1) FFT/filter bank coefficients
    - 2) LPC coefficients
    - 3) Auditory model
      - a) See what sounds are masked
      - b) See what we hear most
    - 4) Cepstral coefficients
      - a) Most common modern day
      - b) Fourier transform of Fourier transform
- b. Problems
  - i. Different rates of speaking
  - ii. Coarticulation
    - 1) Words effecting other words
    - 2) Even the same words from the same speaker could be different if used in different orders
  - iii. Spectral differences due to
    - 1) Different speakers
      - a) People speak very differently
    - 2) Accent
    - 3) Room acoustics
    - 4) Microphone characteristics
      - a) Each microphone is unique
    - 5) Background noise
  - iv. How can we overcome these?
  - v. How accurate do we have to be?
    - 1) If it is a telephone recognizer that is 90% accurate
    - 2) 10 digits in phone number
      - a) Probability of 1 digits being correct = .9
      - b) Probability of 2 digits being correct = .81
      - c) Probability of 3 digits being correct = .729
      - d) Probability of 4 digits being correct = .6561
      - e) Probability of 5 digits being correct = .59
      - f) Probability of 6 digits being correct = .53
      - g) Probability of 7 digits being correct = .47
      - h) Already less then 50%
  - vi. Where does the speech start and end?
    - 1) Endpoints
      - a) Some words have silence in the word, ie six, Kansas City
    - 2) Probably want some slack in the frequency threshold
      - a) So a start threshold
      - b) Then a stop threshold
    - 3) It might come back up so also keep looking for a certain period after activation
- c. Recognition Units
  - i. Words
    - 1) 300,000 in English; 50,000 common

- 2) 12,000 morphemes
- ii. Demisyllables
  - 1) Middle of the syllable to middle of the next syllable
  - 2) 2000 (1500 in initial position of word - 3000 in final position of word, some overlap)
- iii. Syllable
  - 1) Vowel nucleus + surrounding consonants
  - 2) 4400 (1370 covers 93% of English)
- iv. Phonemes
  - 1) Smallest unit that makes a difference in a words sound sematically
  - 2) ~40 in English
  - 3) One: oo ah n
  - 4) It is an abstraction and have different sounds
- v. Allophones
  - 1) Acoustic instances of a phoneme
    - a) Includes stresses of phonemes like hard t and soft t, pit, piT
  - 2) Thousands, but we can get by with ~250
- vi. Diphone
  - 1) Middle of one phoneme to the middle of the next
  - 2) ~1200
  - 3) Street
    - a) As diphone #s-st-tr-ree-eet-t#
- d. Approaches
  - i. Template matching
    - 1) Nearest neighbor rule (KNN)
    - 2) Matching numbers
      - a) Have templates of one, two, three, etc
      - b) Get input and compare to templates
      - c) Match to closest one
    - 3) Pretty good
    - 4) Not as easy to use for large library
    - 5) Speech->front end processing->pattern matching (stored models)->decision rule-> recognized or unknown
      - a) Decision rule could return unknown if not a good enough match exists
  - ii. Statistical pattern recognition
    - 1) Uses Bayes Rule
  - iii. Works well for small vocabs and
    - 1) Hidden markov models HMMs
    - 2) Good for large vocab
    - 3) Probably what you will use if you're going to do large vocab recognition commercially
  - iv. Neural networks
    - 1) In development

4/20/2009, 10:03 AM

How do we recognize speech?

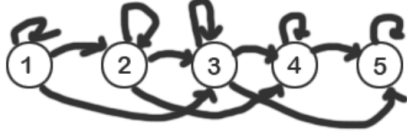
- a. Problem: people say things at different rates
- b. fil (dynamic time warping)
  - i. Form of dynamic programming
  - ii. Example
    - 1) Input
      - High 2 1 3 4 5
      - Mid 4 7 2 1 3
      - Low 5 11 4 5 8
    - 2) Recognizer
      - 3 0 1 3 4 5
      - 5 5 4 2 7 5
      - 2 9 4 4 1 5
    - 3) We will use abs value distance instead of Euclidian distance
      - Local distance matrix
      - 4 12 6 5 5
      - 9 13 9 10 12
      - 4 14 0 3 7
      - 2 10 4 7 9
      - 7 5 11 12 8
      - 5 13 5 8 10
      - Global distance matrix
      - Monotonic
      - Use neighbors to help recognition

Global Distance<sub>i,j</sub> = local distance<sub>i,j</sub> + min(D<sub>i-1,j</sub>, D<sub>i-1,j-1</sub>, D<sub>i,j-1</sub>)  
 31 39 29 28 29 <-- 29 is the actual distance between the utterances  
 27 31 23 24 29  
 18 28 14 17 24  
 14 20 14 21 30  
 12 10 21 33 39  
 5 18 23 31 41

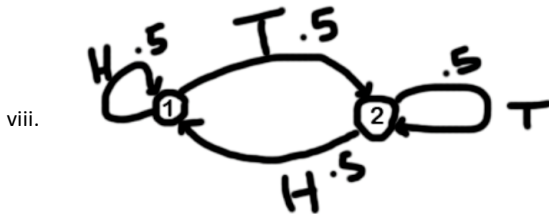
a) You do this for every frame then return the one that matches the best

c. HMM (hidden markov model)

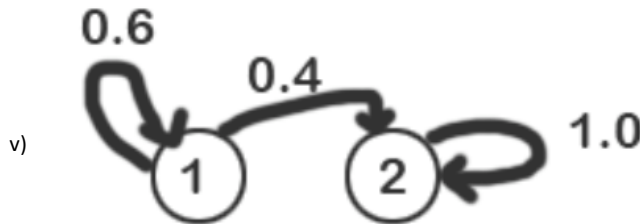
- i. Models speech pretty well
- ii. Finite state machine with probabilistic transitions
  - 1) 5 state model usually used



- iii. Set of states {s} states S<sub>i</sub>, S<sub>f</sub>
- iv. Set of Transitions {a<sub>i,j</sub>} a<sub>i,j</sub>=probability of transitioning for s<sub>i</sub> to s<sub>j</sub>
- v. {b<sub>i,j(k)</sub>} output probability matrix
- vi. This model makes two assumptions
  - 1) Probability of being in a particular state at time t+1 depends on the state at t only
  - 2) Output independence assumption
    - a) The probability of a particular symbol being outputted is independent of the past
- vii. Example



- ix. Three problems
  - 1) Evaluation
    - a) Given a model and an observed sequence what is the probability that the model generated that sequence
    - b) Isolated word recognition
    - c) Solved by Forward algorithm
      - i) Summing all paths of length t (time frame)
      - ii) Exponential algorithm
      - iii) Recursive
      - iv) Model:



vi) [A 0.8, B 0.2] [A 0.5, B 0.5] [A 0.3, B 0.7]

vii)

| t=0           | A           | t=1  | A            | t=2  | B            | t=3  |
|---------------|-------------|------|--------------|------|--------------|------|
| (1.0) State 1 | -(.8*.6)->  | 0.48 | -(0.6*0.8)-> | 0.23 | -(0.6*0.2)-> | 0.03 |
|               | \ (0.4*0.5) |      | \ (0.4*0.5)  |      | \ (0.4*0.5)  |      |
| (0.0) State 2 | ----->      | 0.2  | -(1.0*0.3)-> | 0.16 | -(1*0.7)->   | 0.16 |

P(M|y)=P(y|M)\*P(M)

2) Decoding

- a) Given a model and an observed sequence what is the most likely state sequence that observed sequence?
- b) continuous speech recognition
- c) Viterbi

- i) Looks back to see the most probably state for each time sequence
    - ii) At the end our most likely state was 2 (.16), before that most likely is state 1 (.23), before that the most likely is state 1 (.48)
    - iii) Therefore, went state1 (1.0) -> state 1 (.48) -> state 1 (.23) ->state 2 (.16)
- 3) Learning
  - a) Given a model and an observed sequence, what should the models parameters be so it has a high probability of generating that sequence?

#### Speech Recognition Evaluation

- a. Pretty easy evaluation
- b. Standard data sets
  - i. Speaker dependent
    - 1) TI-20: digits and control words
    - 2) TI-alpha: alphabet a to z
  - ii. Speaker independent
    - 1) TIMIT: Set of sentence that are intended for continuous speech
  - iii. Noise condition sets exist to
- c. Do a test
  - i. Results:
    - 1) % correct
    - 2) % wrong
    - 3) % reject

#### Synthesis/ transmission/ coding Evaluation

- a. 2 evaluation factors
  - i. Quality
  - ii. Intelligibility
- b. Objective subjective
  - i. Objective
    - 1) Speech recognizer
    - 2) SNR
    - 3) Doesn't really work that well
  - ii. Subjective
    - 1) A person
    - 2) Listener opinion test
    - 3) Predict degree of satisfaction for users
    - 4) Asses relative effects of different kinds of degradation
    - 5) MOS System
      - a) People listen to sentences
      - b) 1-5
        - i) 1 very annoying
        - ii) 5 excellent
      - c) Abs category rating
      - d) A few seconds for response
      - e) Not longer then an hour test
    - 6) D-MOS
      - a) Rates the degradation, you play unencoded then encoded
    - 7) Articulation (intelligibility test)
      - a) DRT
        - i) Uses 96 pairs of words, words have minimal differences
          - One. Word differ by only one phoneme
        - ii) Test consonants
        - iii) You ask which word they recognized of and play one of them
          - One. le play Daunt and the listener can select from daunt or taunt

**4/24/2009, 10:04 AM**

Group presentation describing a multimedia development library or language  
 CSound, JavaSound, Java Media PAI, Python Imaging Library, PureData, Nyquist)

3-4 group

10-15 minutes

Capabilities

Advantages/disadvantages

Sampling rates?

Easy to use?

Image and Sound?

2D and 3D?

Any built in filtering?

Persistence of Vision

- a. The eye sees an image it retains it
- b. Flicker fusion
  - i. If fps is faster than 40 the images are blurred together
  - ii. ~40 fps for flicker fusion
- c. Movies
  - i. They show each frame twice
  - ii.  $24 \text{ fps} \times 2 = 48 \text{ fps}$
  - iii. Film is celluloid tape
    - 1) Characterized by width
    - 2) 35mm film
    - 3) Images
    - 4) Perforations to advance film through camera
    - 5) Aspect ratio
  - iv. Silent movies
    - 1) 16mm film
    - 2) Aspect ratio 1.33:1 (Academy ratio/aperture)
  - v. 1930 - 1950
    - 1) 35mm film
    - 2) US
      - a) Aspect ratio 1.37:1 (Academy full screen)
    - 3) European
      - a) Aspect ratio 1.66:1
  - vi. 1950's Widescreen
    - 1) Vista Vision (1.85:1)
      - a) Rotated the film 90 degrees
      - b) This allowed for wide shots
      - c) Filmed with this
        - i) 10 Commandments
        - ii) White Christmas
    - 2) Anamorphic lens (2.39:1)
      - a) How to marry a millionaire
      - b) The Robe
      - c) 20 Leagues under the sea
    - 3) 70mm film (also called 65mm)
      - a) Easier to make widescreen
      - b) Standard aspect ratio is 2.2:1
        - i) Lawrence of Arabia
        - ii) IMAX
          - One. Turn it sideways
          - Two. Uses 1.43:1 because of the large image sizes
- d. TV
  - i. Uses lines
  - ii. Broadcasted a line at a time
  - iii. SDTV
    - 1) Radio waves 4:3
  - iv. HDTV
    - 1) 1981
    - 2) Came around in Japan
    - 3) 1125 lines/frame
    - 4) 30 fps
    - 5) 16:9
  - v. HDTV now
    - 1) 16:9
    - 2) 59.94 fps
    - 3) Surround Sound
    - 4) 1080i
      - a) 1920x1080
      - b) interlaced scanning
    - 5) 1080p
      - a) 1920x1080
      - b) Progressive scanning
    - 6) 720p
      - a) 1280x720
      - b) Progressive scanning

- vi. NTSC
  - 1) US, Japan, Taiwan, some Caribbean/South America
  - 2) 1941 Monochrome Standards
  - 3) 1954 Color Standards
- vii. PAL (Phase Alternating Line)
  - 1) UK, Western Europe, Australia, New Zealand, China
  - 2) 1967 Color Standards
- viii. SECAM
  - 1) France, Eastern Europe
  - 2) 1967 Standards

Video compression

- a. MPEG standards
- b. MPEG1
  - i. Medium bandwidth
  - ii.  $\leq 1.5$  Mbits/sec
  - iii. CD-ROM
  - iv. 1.25mb/sec
    - 1) Allocated to video
    - 2) 352x240
    - 3) 30 Hz
    - 4) Non-interlaced
  - v. 250 Kb/s audio
    - 1) 2 channels
    - 2) Mp3
- c. MPEG2
  - i. Higher bandwidth
  - ii.  $\leq 80$ Mbits/sec
  - iii. Up to 5 audio channels
    - 1) Supports surround sound
    - 2) AAC
  - iv. Wider range of frame rates
    - 1) Supports HDTV
  - v. Can be interlaced or non interlaced
- d. MPEG3
  - i. Intended for HDTV
  - ii. Up to frames of 1920x1080
  - iii. 30 Hz
  - iv. MPEG2 supports HDTV well, so MPEG3 hasn't been adopted
- e. MPEG4
  - i. Intended for very low bandwidth
  - ii.  $\leq 64$ Kb/s
  - iii. 176x155
  - iv. 10Hz
  - v. Optimized for video phones
  - vi. Extended to general multimedia coding standard
    - 1) For web and mobile application
    - 2) Supports
      - a) Midi
      - b) VRML 3D rendering
      - c) Digital rights management
- f. MPEG7
  - i. Metadata standard
    - 1) Metadata: data about data
  - ii. Objectives: provide fast and efficient
    - 1) Filtering
    - 2) Searching
    - 3) Content id
    - 4) Low level characteristics
    - 5) Data structure
    - 6) Models
    - 7) Index a wide range of data
  - iii. Independence between the information and description of the information
  - iv. Does audio, video, voice, images, graphs, 3D models
- g. MPEG21
  - i. Multimedia framework
  - ii. Uses Y-Cr-Cb
  - iii. 3 types of pictures

- 1) Intra pictures (I pictures)
  - a) Coded using only info in picture
  - b) Provide random access into the video sequence
  - c) You can always jump to an I picture
  - d) Uses transform coding
  - e) ~2bits per pixel compression
  - f) Encoding Process (like jpeg)
    - i) DCT in 8x8
    - ii) Quantize the DCT coefficients
    - iii) RLE
    - iv) Huffman
- 2) Predicted pictures (P pictures)
  - a) Coded as error from previous p picture or I picture
  - b) Forward prediction
  - c) Motion compensation
    - i) Uses 16x16 macro blocks
    - ii) Finds nearest 16x16 block so even if its moved you can still get it
  - d) Encoding
    - i) Motion vector
    - ii) DCT quantization
    - iii) RLE
    - iv) Huffman
- 3) Bidirectional pictures (B pictures)
  - a) Motion vector
  - b) Can reference previous or next I or P picture
- 4) Levels
 

|              |                |        |
|--------------|----------------|--------|
| a) Low       | 253x240 30Hz   | 4Mb/s  |
| b) Main      | 720x480 30Hz   | 15Mb/s |
| c) High 1440 | 1440x1152 30Hz | 60Mb/s |
| d) High      | 1920x1080 30Hz | 80Mb/s |
- 5) Profile
  - a) Simple
    - i) No b pictures
    - ii) Used 95% of time
    - iii) Cable TV, television etc
  - b) Main
    - i) Used 95% of the time
    - ii) Cable TV, television, etc
  - c) Main Plus
    - i) Adds scalability
    - ii) Resolution coding

h. AAC

- i. Channels
  - 1) Left
  - 2) Right
  - 3) Center
  - 4) 2 surround
- ii. Low frequency enhancements
- iii. Up to 7 commentary/multilingual channels
- iv. Supports half sampling rates
  - 1) 16kHz
  - 2) 22.05kHz
  - 3) 24kHz

Results of hw3

- a. If we put in a sine wave at 300 Hz twice, we get 600 Hz peak in spectrum
- b. If we put in a 300 Hz and 500 Hz, we get 100 Hz and 700 Hz peaks in spectrum
- c. If we put in 300 Hz and 500 Hz, we get 200 Hz and 801 Hz peaks in spectrum
- d. If we put in 300 Hz and 600 Hz, we get 301 Hz and 899 Hz peaks
- e. If we put in 400 Hz and 500 Hz, we get peaks at 100 Hz and 899 Hz
- f. If we put in 400 Hz and 600 Hz, we get a peak at 200 Hz and 1005 Hz
- g. We get the sum and difference tones

Content-based retrieval

- a. TreeQ Music Retrieval
  - i. Generate a profile for every song
  - ii. Find the distance between the passed song and all the profiles
  - iii. Closest one is probably the right song

iv. [www.rotorbrain.com/foote/musicr/nuts+bolts.html](http://www.rotorbrain.com/foote/musicr/nuts+bolts.html)

b. Example:

- i. Say you have a bunch of images, you want to be able to search for shade and it bring up all the pictures of shade without using any time of tag
- ii. Shazam for the iPhone

Final

Monday 5/100

11:00 - 1:00

Cheek 213

What a Fourier Transform gives you/does and assumptions it makes

Periodicity assumption

Expects one period of periodic data

Block based

Windowing

To minimize spectral leakage (frequencies leak into other bins)

Time / frequency resolution trade off

Long blocks, very good frequency resolution

Short blocks, very good time resolution

Highest bin

Half your sampling rate

Pitch tracking

How it is done

Time domain

-Find the peaks

Frequency domain

-Do FFT

-Look at spectrum

Cepstral domain

-Does FFT

-Does FFT on result

-Looks at spectrum

-In the quefrency there is a blip that is the frequency

LPC residual

-Error left over after predicting waveform

-As the waveform goes along within the pitch period we are modeling the samples pretty well then when a new pitch period starts the prediction messes up for a little while so there are peaks where this happens

Aliasing

What frequencies alias to

Aliasing mirrors around the Nyquist Frequency

8kHz sampling

| Sample at | Aliasing |
|-----------|----------|
| 5kHz      | 1kHz     |
|           |          |

Speech production

Source filter model

Source of sound that travels through a filter

Filter is your mouth, tongue, etc

Model of how we produce speech

Speech coding

Pulse Code Modulation

Basic sampling and quantization

Delta Pulse Code Modulation

Difference samples

Adaptive Pulse Code Modulation

Checks

Look these up

CVSD fixes these:

Granular noise

Very close to silence  
If you do speech encoding with delta modulation you will get noise  
Slope overload  
Step size not big enough to keep up with the signal

#### Filtering

High pass filter  
Low pass filter  
Band pass filter  
Notch filter  
Recognize filter by its equation  
Frequency response of filter  
Phase response of filter  
Impulse response with equation given  
If we difference or integrate samples know what it does for high pass or low pass  
Know what infinite peak clipping is

#### Frequency inversion

Simple method of encryption  
Did a lab on it

#### Vector quantization

Be able to find bit rate  
 $R = \log N / k$   
R:  
N:  
k:

#### LPC Linear Predictive Coding

Know Residual/distortion error/signal  
How its used in pitch tracking  
Know problems  
Basic LPC model  
Noise source (pulses/white noise)

#### RELPC residual excited linear prediction

Uses a segment of the residual for source, instead of pulses/white noise

#### CELP code excited linear prediction

Its RELPC with a vector quantization code book

#### MP3

How it works  
Mpeg-1 layer 3 audio  
Sub band coding

#### MPEG

Know what 1,2,4,7,21 are/their purpose  
Know how mpeg2 does video compression  
Ipicturesf  
Ppictures  
Bpictures

#### Speech synthesis

Process for text to speech  
Copy synthesis  
Why you need multiple recordings of the same word  
Prosody  
Tonal language  
Chinese  
Where pitch makes a phonetic difference

#### Speech recognition

DTW Dynamic Time Warping  
Be able to do it with two sets of numbers  
HMM Hidden Markov Model  
Know what it is  
If given a HMM and sequence, tell the probability that model gave that speech

Forward algorithm to do this