CS 224S / LINGUIST 281
Speech Recognition, Synthesis, and Dialogue

Dan Jurafsky

Lecture 6: Waveform Synthesis (in Concatenative TTS)

IP Notice: many of these slides come directly from Richard Sproat's slides, and others (and some of Richard's) come from Alan Black's excellent TTS lecture notes. A couple also from Paul Taylor

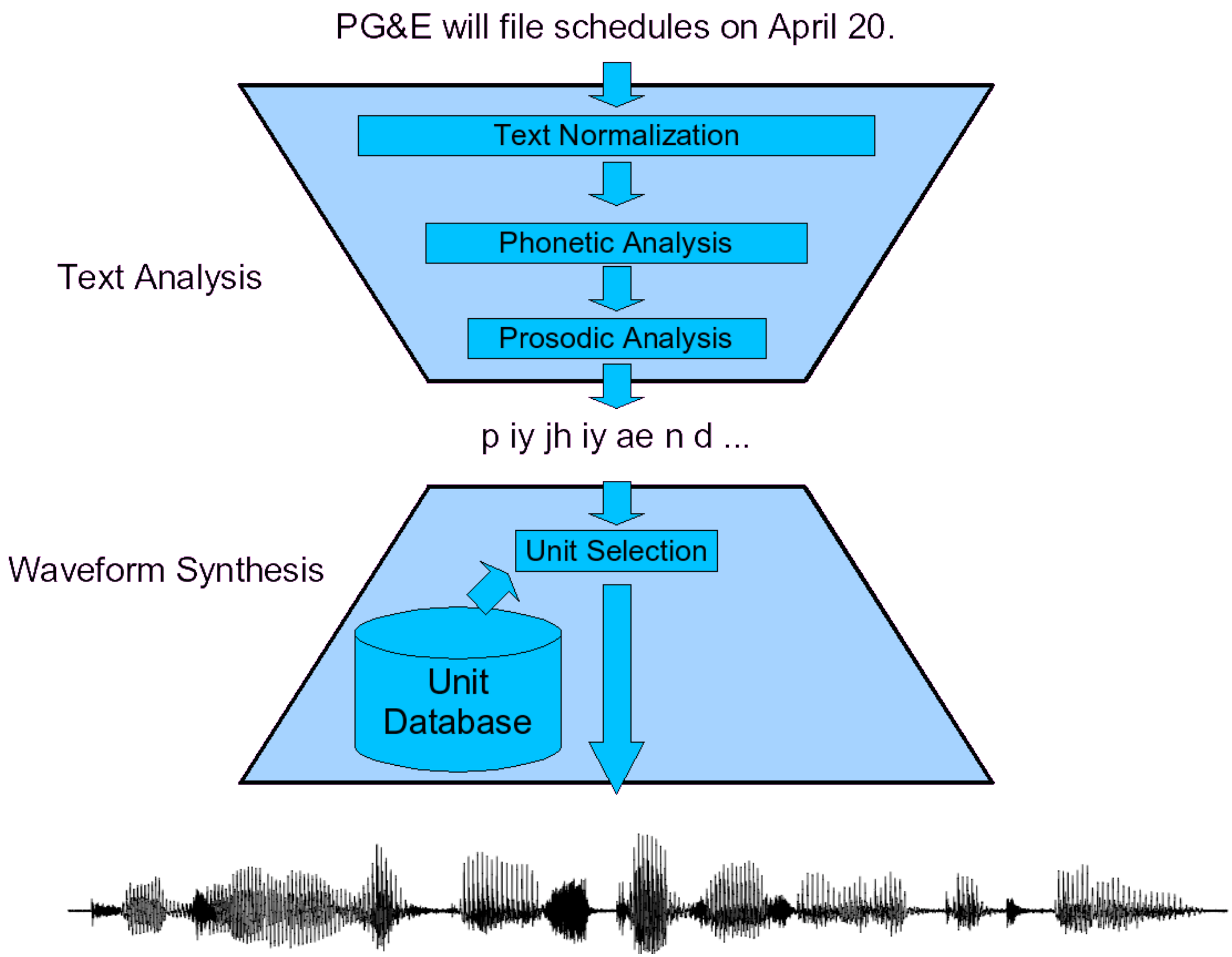
Goal of Today’s Lecture

- Given:
  - String of phones
  - Prosody
    - Desired F0 for entire utterance
    - Duration for each phone
    - Stress value for each phone, possibly accent value
- Generate:
  - Waveforms

Outline: Waveform Synthesis in Concatenative TTS

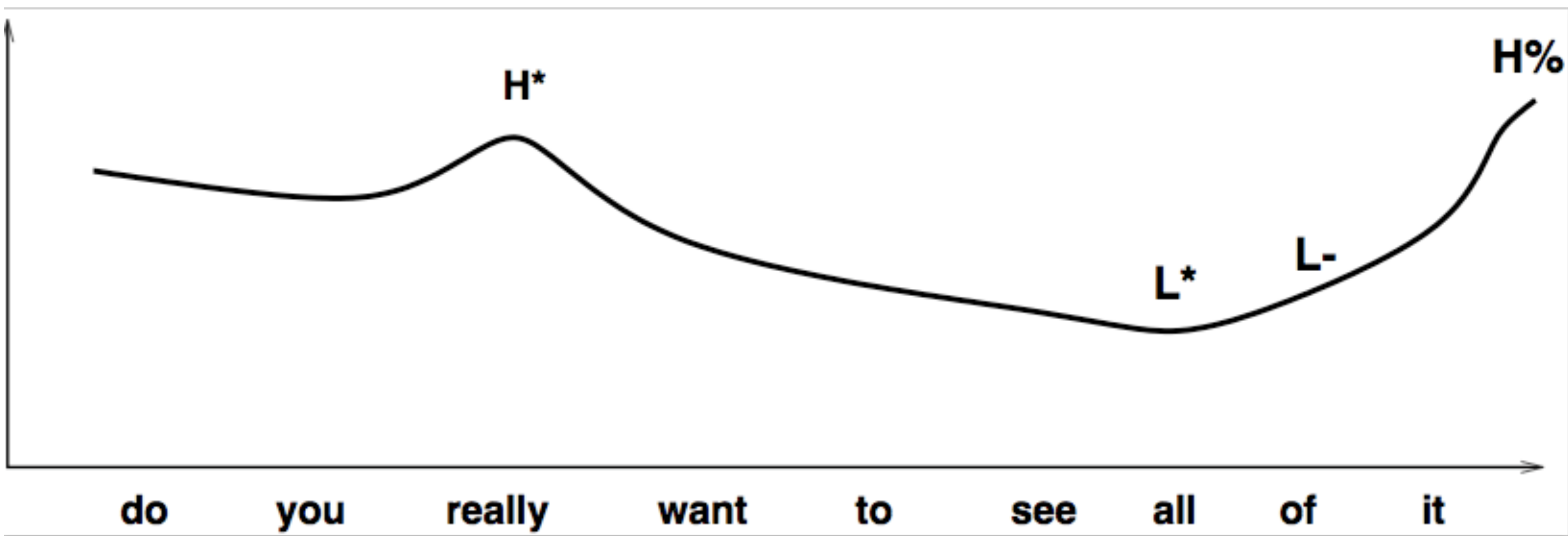
- Diphone Synthesis
- Break: Final Projects
- Unit Selection Synthesis
  - Target cost
  - Unit cost
- Joining
  - Dumb
  - PSOLA

The hourglass architecture



Internal Representation: Input to Waveform Wynthesis

				H*												L*		L- H%			
do		you		really				want				to		see		all		of		it	
d	uw	y	uw	r	ih	l	iy	w	aa	n	t	t	ax	s	iy	ao	l	ah	v	ih	t
110	110	50	50	75	64	57	82	57	50	72	41	43	47	54	130	76	90	44	62	46	220

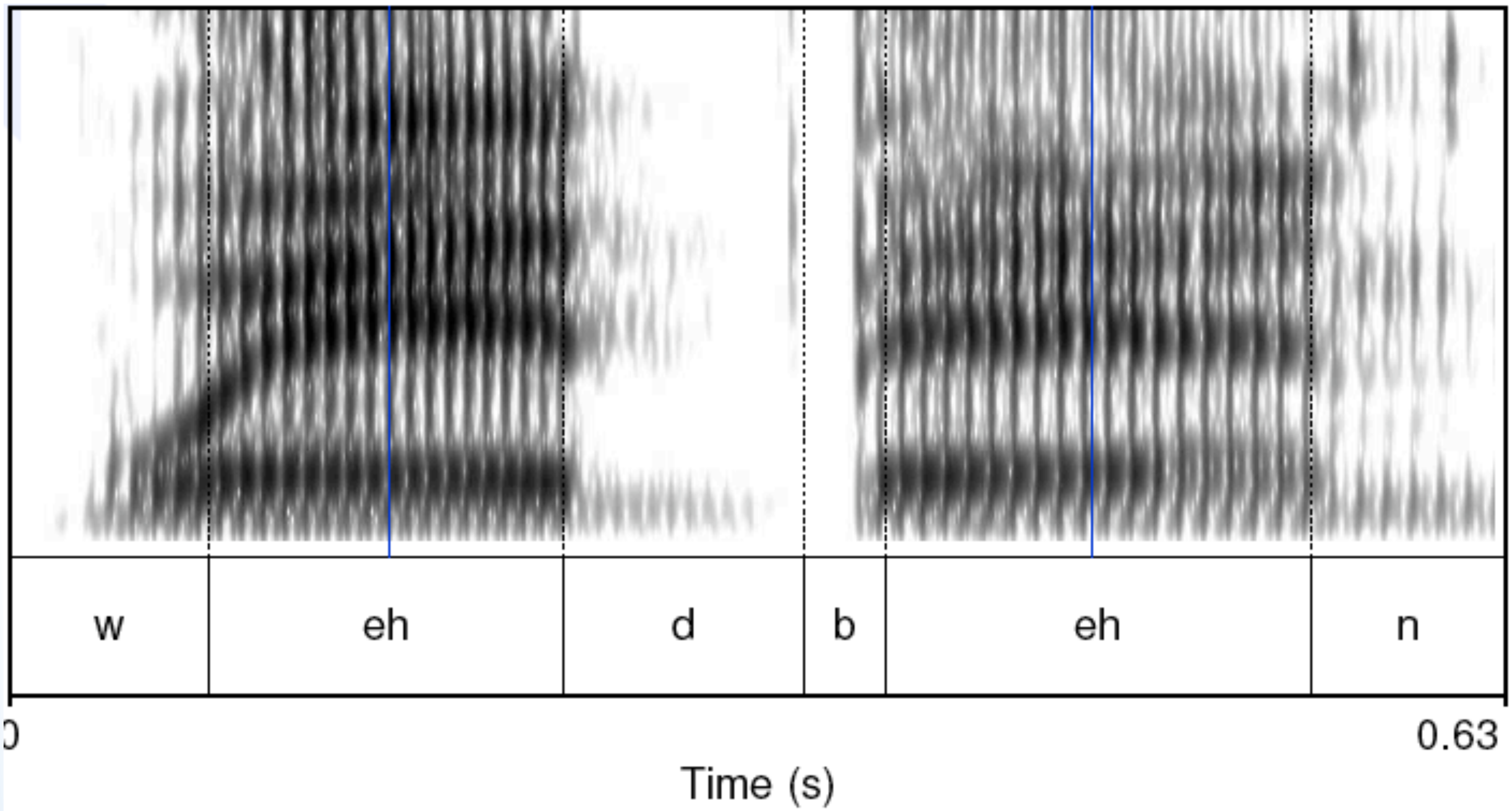


Diphone TTS architecture

- Training:
  - Choose units (kinds of diphones)
  - Record 1 speaker saying 1 example of each diphone
  - Mark the boundaries of each diphones,
    - cut each diphone out and create a diphone database
- Synthesizing an utterance,
  - grab relevant sequence of diphones from database
  - Concatenate the diphones, doing slight signal processing at boundaries
  - use signal processing to change the prosody (F0, energy, duration) of selected sequence of diphones

Diphones

- Mid-phone is more stable than edge:



Diphones

- mid-phone is more stable than edge
- Need O(phone^2) number of units
  - Some combinations don't exist (hopefully)
  - ATT (Olive et al. 1998) system had 43 phones
    - 1849 possible diphones
    - Phonotactics ([h] only occurs before vowels), don't need to keep diphones across silence
    - Only 1172 actual diphones
  - May include stress, consonant clusters
    - So could have more
  - Lots of phonetic knowledge in design
- Database relatively small (by today's standards)
  - Around 8 megabytes for English (16 KHz 16 bit)

Slide from Richard Sproat

Voice

- Speaker
  - Called a **voice talent**
- Diphone database
  - Called a **voice**

Designing a diphone inventory: Nonsense words

- Build set of carrier words:
  - pau t aa b aa b aa pau
  - pau t aa m aa m aa pau
  - pau t aa m iy m aa pau
  - pau t aa m iy m aa pau
  - pau t aa m ih m aa pau
- Advantages:
  - Easy to get all diphones
  - Likely to be pronounced consistently
    - No lexical interference
- Disadvantages:
  - (possibly) bigger database
  - Speaker becomes bored

Slide from Richard Sproat

Designing a diphone inventory: Natural words

- Greedily select sentences/words:
  - Quebecois arguments
  - Brouhaha abstractions
  - Arkansas arranging
- Advantages:
  - Will be pronounced naturally
  - Easier for speaker to pronounce
  - Smaller database? (505 pairs vs. 1345 words)
- Disadvantages:
  - May not be pronounced correctly

Slide from Richard Sproat

Making recordings consistent:

- Diiphone should come from mid-word
  - Help ensure full articulation
- Performed consistently
  - Constant pitch (monotone), power, duration
- Use (synthesized) prompts:
  - Helps avoid pronunciation problems
  - Keeps speaker consistent
  - Used for alignment in labeling

Slide from Richard Sproat

Building diphone schemata

- Find list of phones in language:
  - Plus interesting allophones
  - Stress, tons, clusters, onset/coda, etc
  - Foreign (rare) phones.
- Build carriers for:
  - Consonant-vowel, vowel-consonant
  - Vowel-vowel, consonant-consonant
  - Silence-phone, phone-silence
  - Other special cases
- Check the output:
  - List *all* diphones and justify missing ones
  - Every *diphone* list has mistakes

Slide from Richard Sproat

Recording conditions

- Ideal:
  - Anechoic chamber
  - Studio quality recording
  - EGG signal
- More likely:
  - Quiet room
  - Cheap microphone/sound blaster
  - No EGG
  - Headmounted microphone
- What we can do:
  - Repeatable conditions
  - Careful setting on audio levels

Slide from Richard Sproat

Labeling Diphones

- Run a speech recognizer in **forced** alignment mode
  - Forced alignment:
    - A trained ASR system
    - A wavefile
    - A word transcription of the wavefile
    - Returns an alignment of the phones in the words to the wavefile.
- *Much* easier than phonetic labeling:
  - The words are defined
  - The phone sequence is generally defined
  - They are clearly articulated
  - But sometimes speaker still pronounces wrong, so need to check.
- Phone boundaries less important
  - +- 10 ms is okay
- Midphone boundaries important
  - Where is the stable part
  - Can it be automatically found?

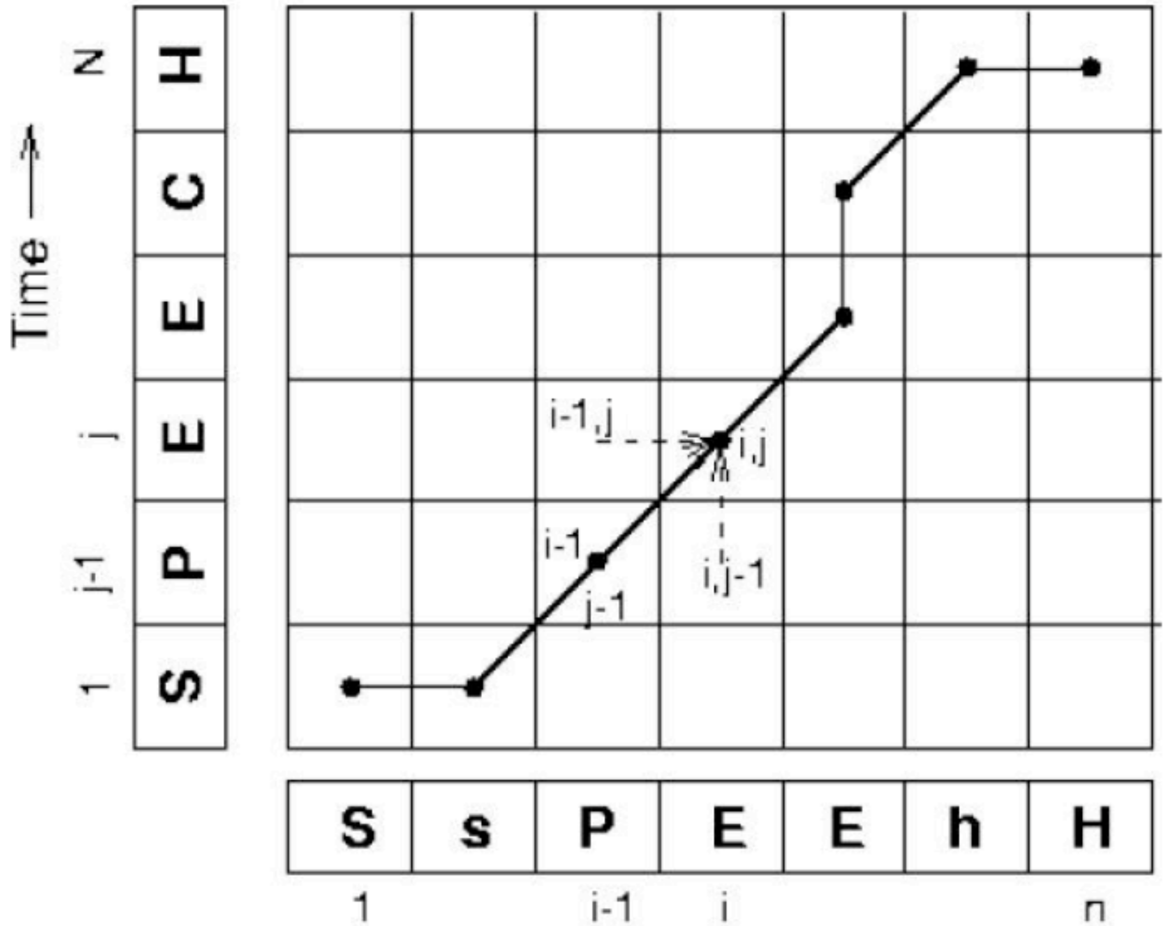
Slide from Richard Sproat

Diphone auto-alignment

- Given
  - synthesized prompts
  - Human speech of same prompts
- Do a dynamic time warping alignment of the two
  - Using Euclidean distance
- Works very well 95%+
  - Errors are typically large (easy to fix)
  - Maybe even automatically detected
- Malfrere and Dutoit (1997)

Slide from Richard Sproat

Dynamic Time Warping



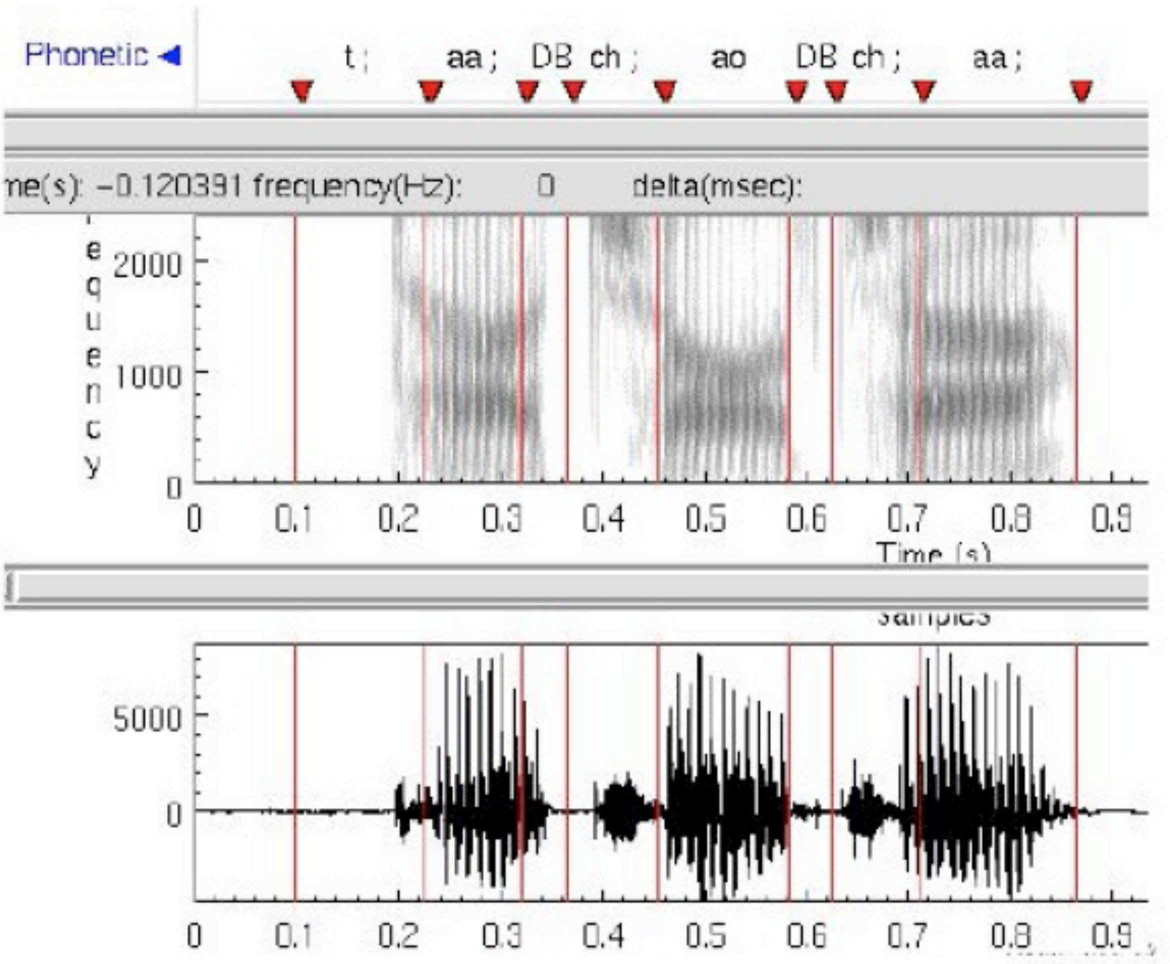
Slide from Richard Sproat

Finding diphone boundaries

- Stable part in phones
  - For stops: one third in
  - For phone-silence: one quarter in
  - For other diphones: 50% in
- In time alignment case:
  - Given explicit known diphone boundaries in prompt in the label file
  - Use dynamic time warping to find same stable point in new speech
- Optimal coupling
  - Taylor and Isard 1991, Conkie and Isard 1996
  - Instead of precutting the diphones
    - Wait until we are about to concatenate the diphones together
    - Then take the 2 complete (uncut diphones)
    - Find optimal join points by measuring cepstral distance at potential join points, pick best

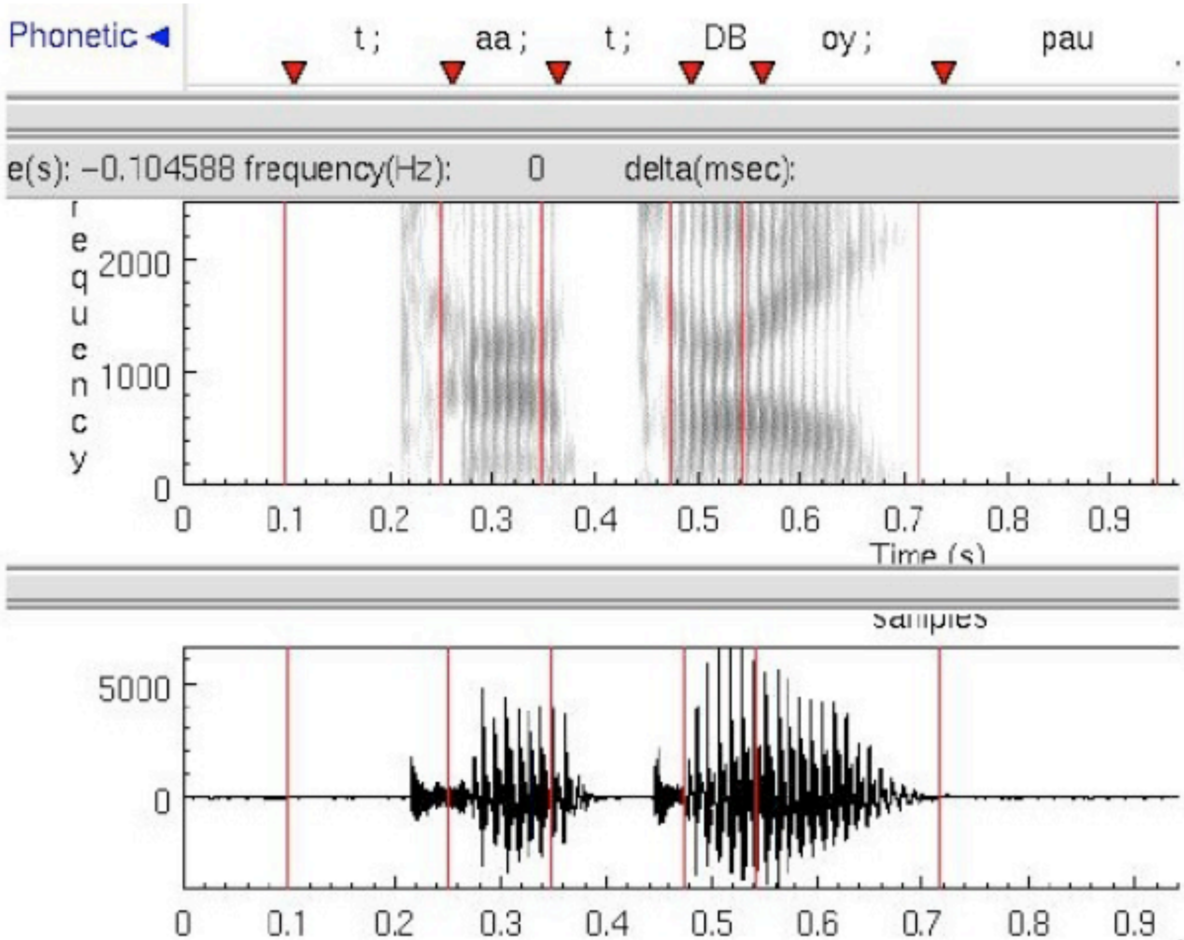
Slide modified from Richard Sproat

Diphone boundaries in stops



Slide from Richard Sproat

Diphone boundaries in end phones



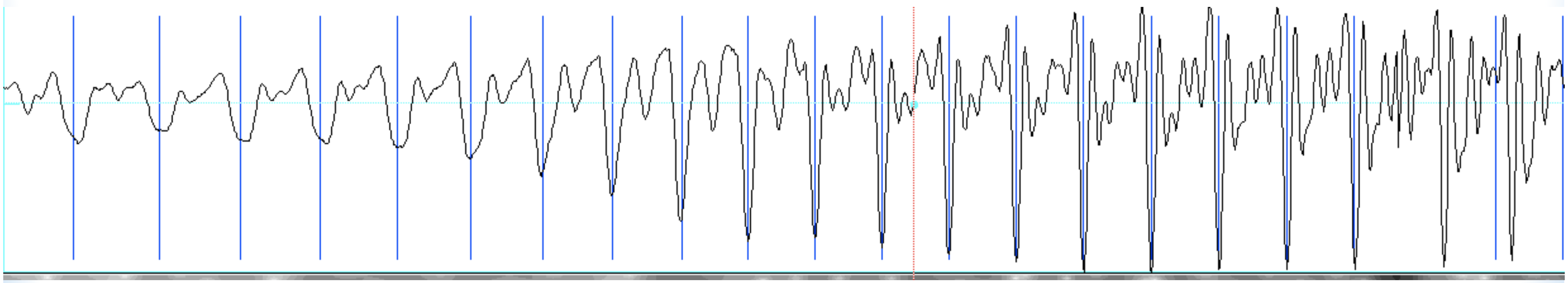
Slide from Richard Sproat

Concatenating diphones: junctures

- If waveforms are very different, will perceive a click at the junctures
  - So need to window them
- Also if both diphones are voiced
  - Need to join them **pitch-synchronously**
- That means we need to know where each pitch period begins, so we can paste at the same place in each pitch period.
  - **Pitch marking** or **epoch detection**: mark where each **pitch pulse** or **epoch** occurs
    - Finding the Instant of Glottal Closure (IGC)
  - (note difference from **pitch tracking**)

Epoch-labeling

- An example of epoch-labeling useing "SHOW PULSES" in Praat:



Epoch-labeling: Electroglottograph (EGG)

- Also called **laryngograph** or **Lx**
  - Device that straps on speaker's neck near the larynx
  - Sends small high frequency current through adam's apple
  - Human tissue conducts well; air not as well
  - Transducer detects how open the glottis is (I.e. amount of air between folds) by measuring impedence.

Picture from UCLA Phonetics Lab

Less invasive way to do epoch-labeling

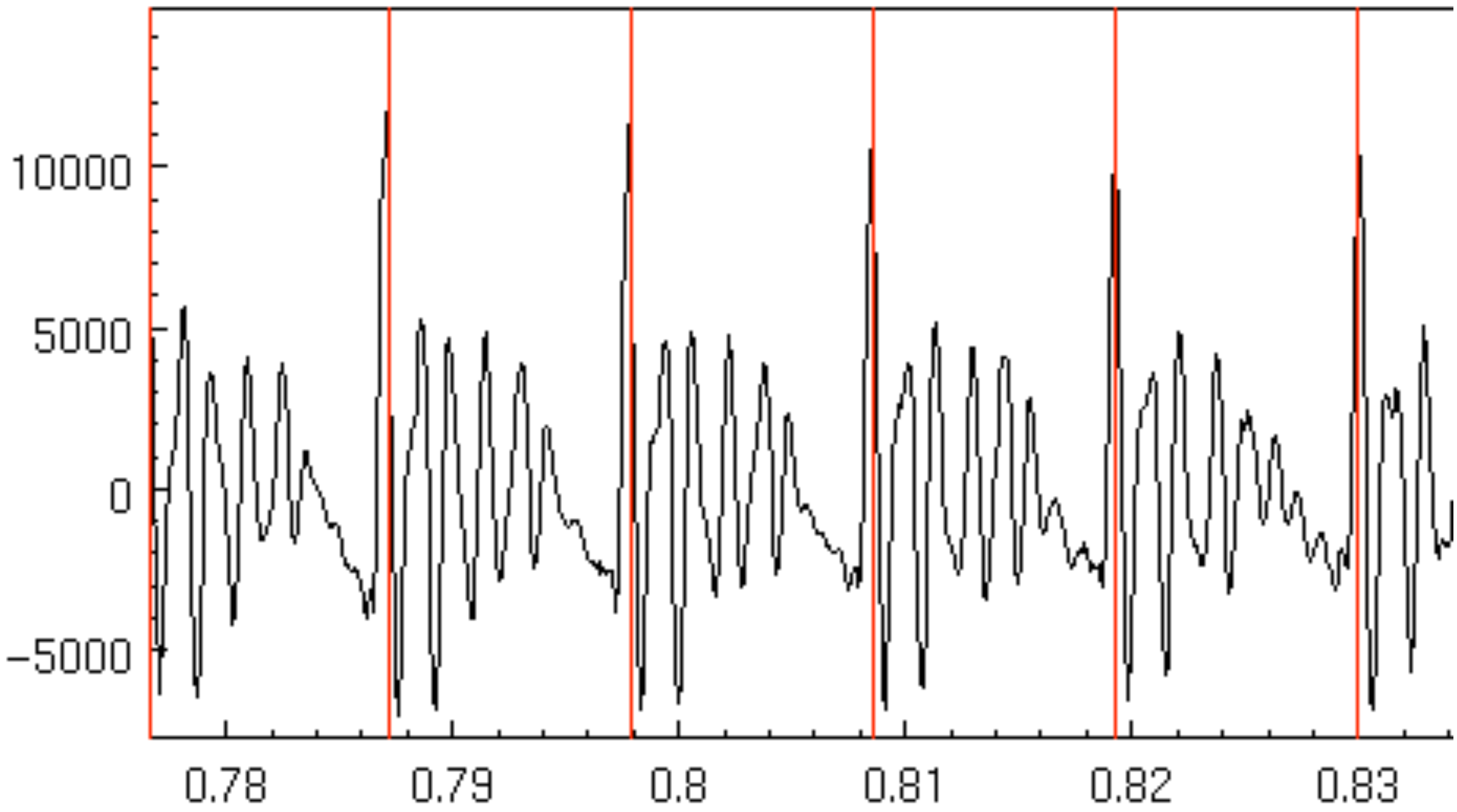
- **Signal processing**
  - E.g.:
  - BROOKES, D. M., AND LOKE, H. P. 1999. Modelling energy flow in the vocal tract with applications to glottal closure and opening detection. In *ICASSP 1999.*

Prosodic Modification

- Modifying pitch and duration *independently*
- Changing sample rate modifies both:
  - Chipmunk speech
- Duration: duplicate/remove parts of the signal
- Pitch: resample to change pitch

Text from Alan Black

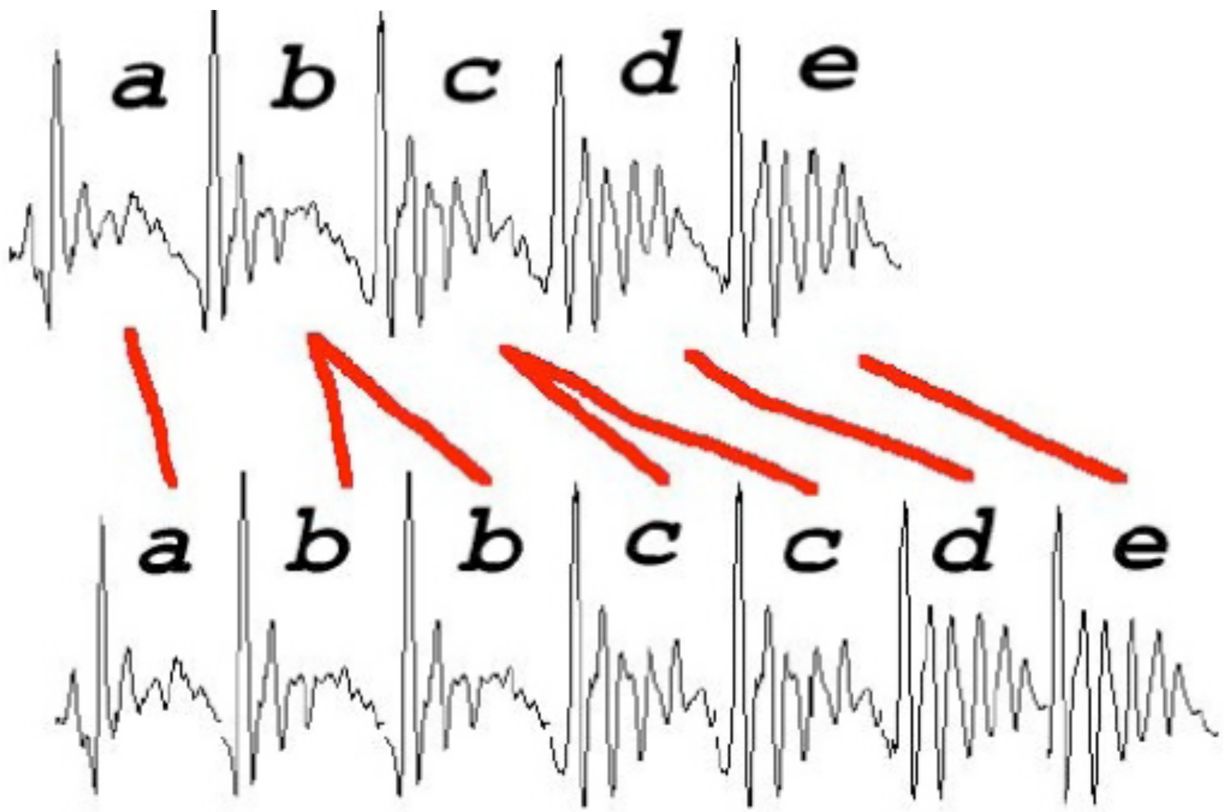
Speech as Short Term signals



Alan Black

Duration modification

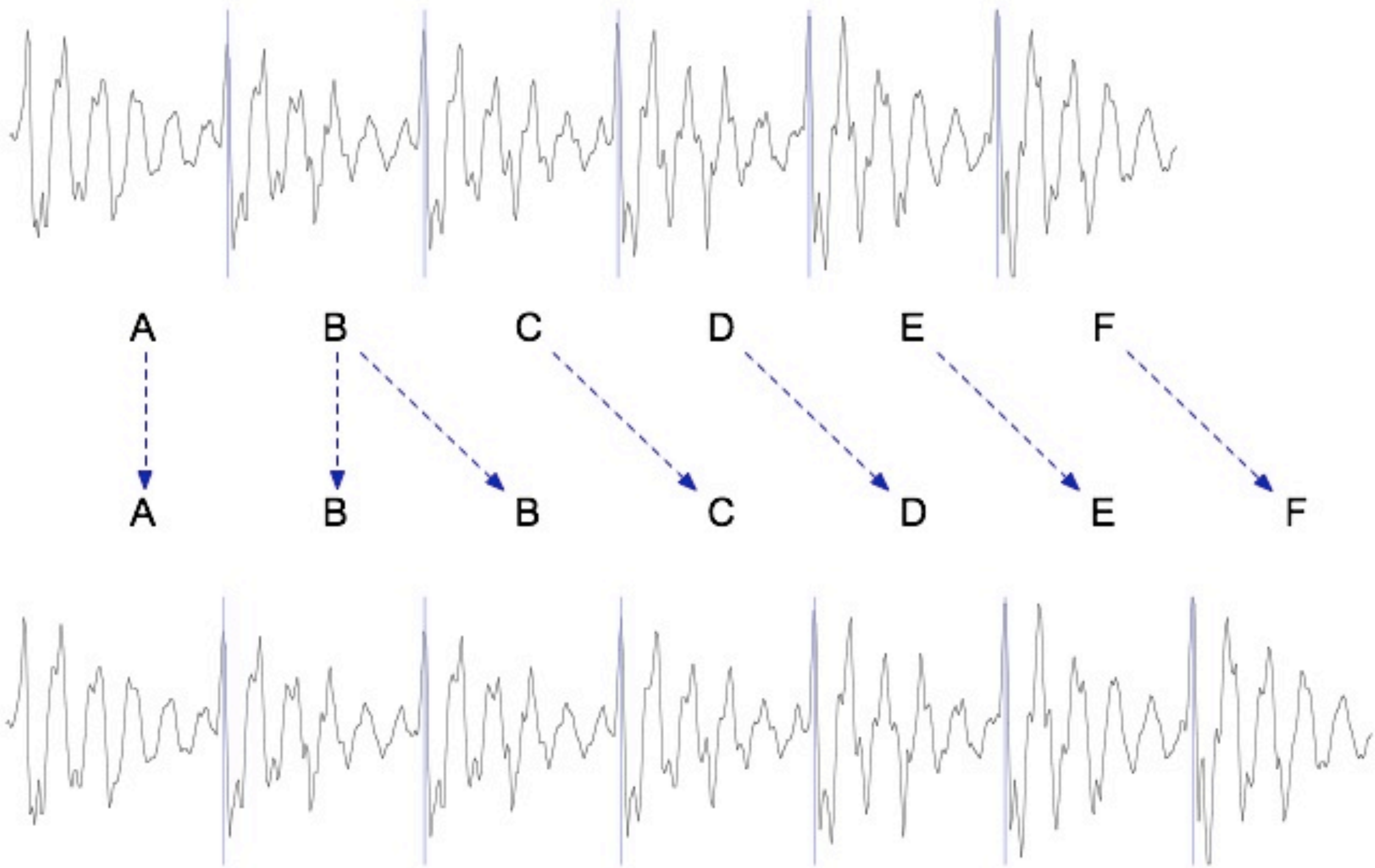
- Duplicate/remove short term signals



Slide from Richard Sproat

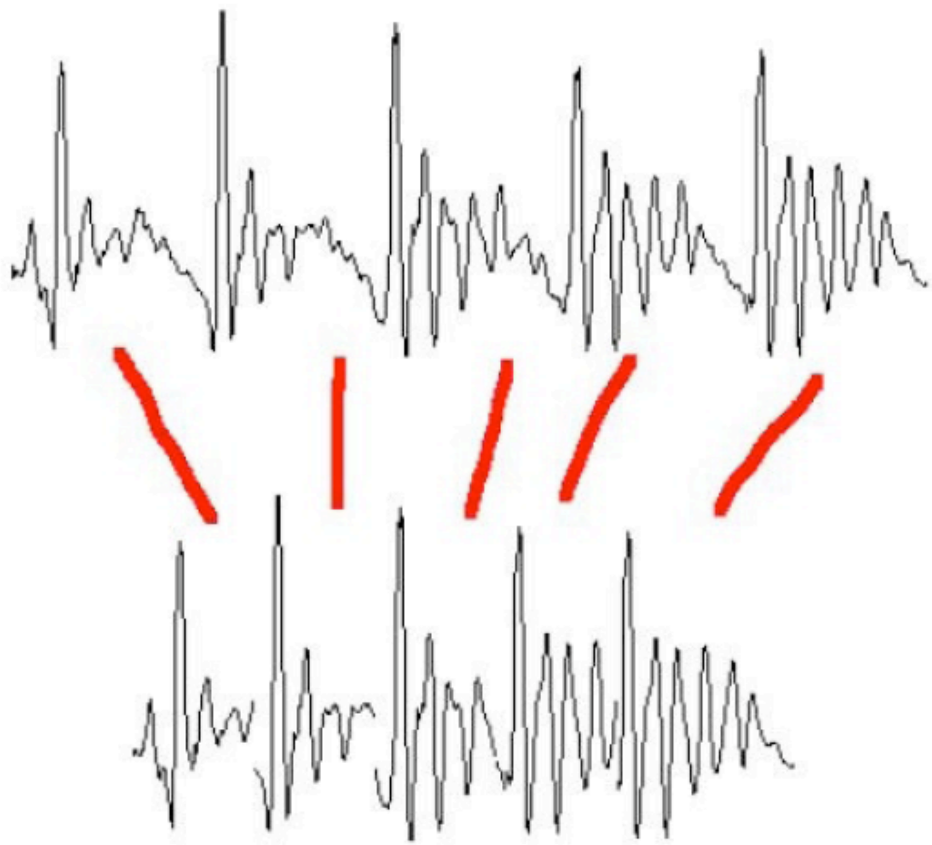
Duration modification

- Duplicate/remove short term signals



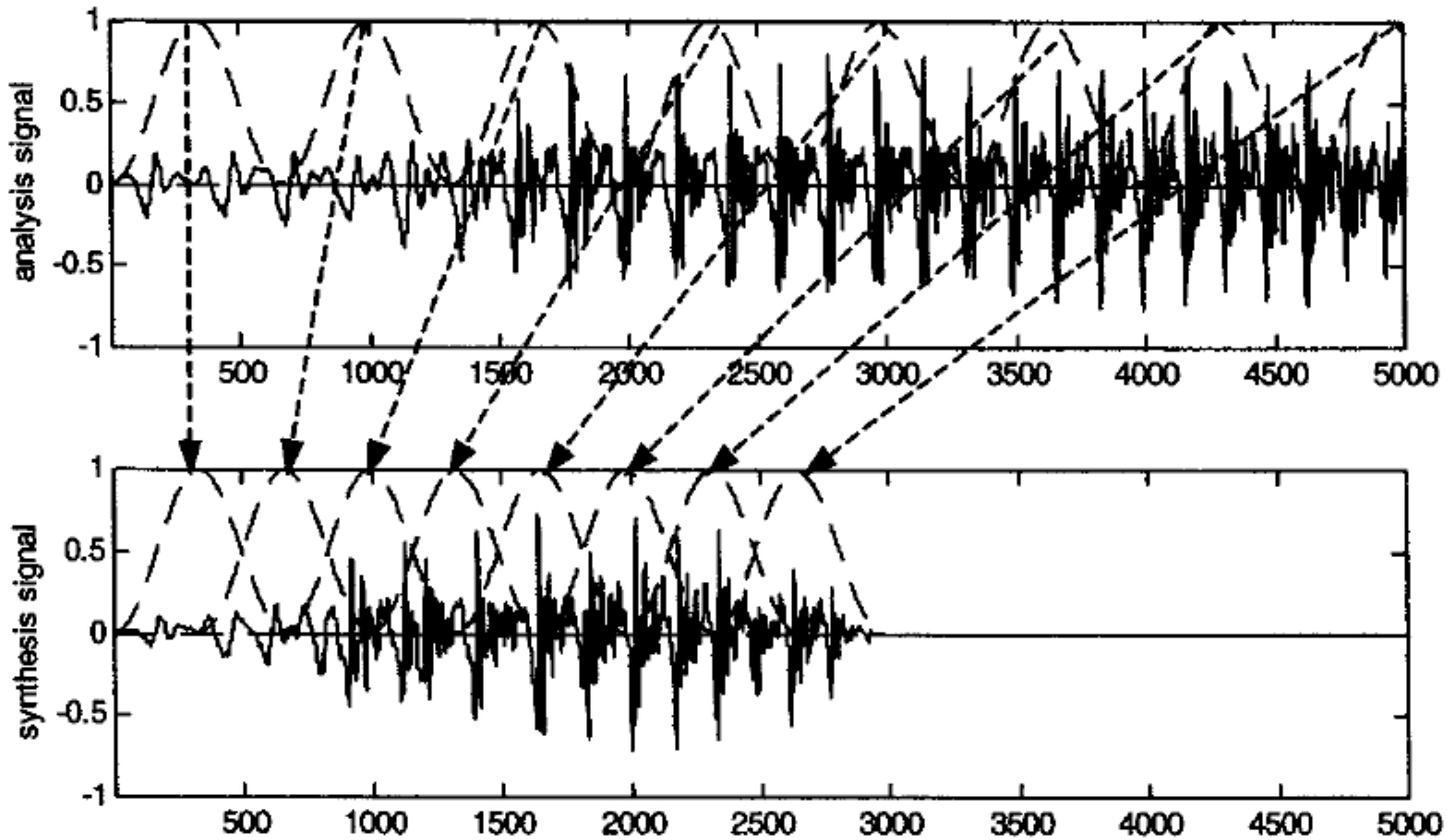
Pitch Modification

- Move short-term signals closer together/further apart



Slide from Richard Sproat

Overlap-and-add (OLA)



Huang, Acero and Hon

Windowing

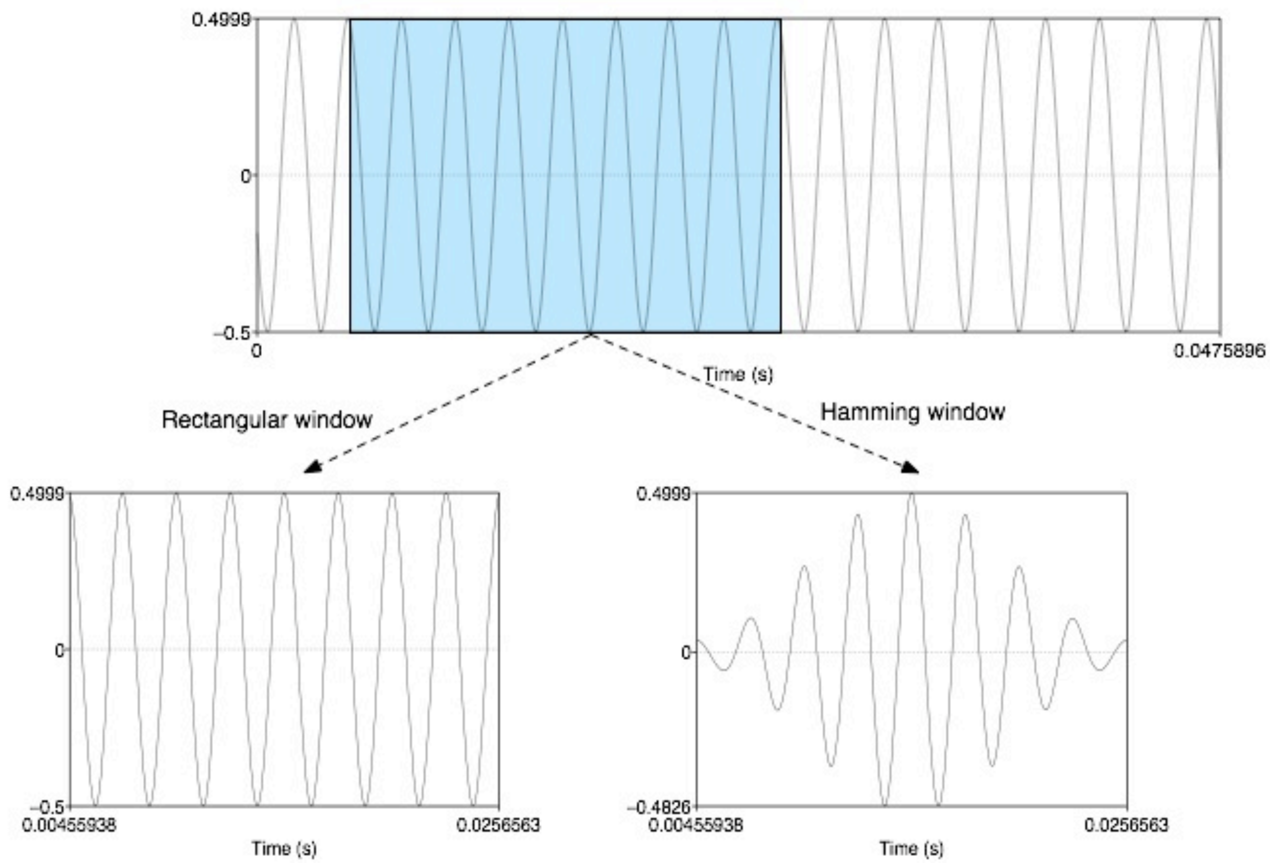
- Multiply value of signal at sample number n by the value of a windowing function
- y[n] = w[n]s[n]

$$\textit{rectangular} \qquad w[n] = \begin{cases} 1 & 0 \le n \le L-1 \\ 0 & \text{otherwise} \end{cases}$$

$$\textit{hamming} \qquad w[n] = \begin{cases} 0.54 - 0.46\cos(\frac{2\pi n}{L}) & 0 \le n \le L-1 \\ 0 & \text{otherwise} \end{cases}$$

Windowing

- $y[n] = w[n]s[n]$



Overlap and Add (OLA)

- Hanning windows of length 2N used to multiply the analysis signal
- Resulting windowed signals are added
- Analysis windows, spaced 2N
- Synthesis windows, spaced N
- Time compression is uniform with factor of 2
- Pitch periodicity somewhat lost around 4th window

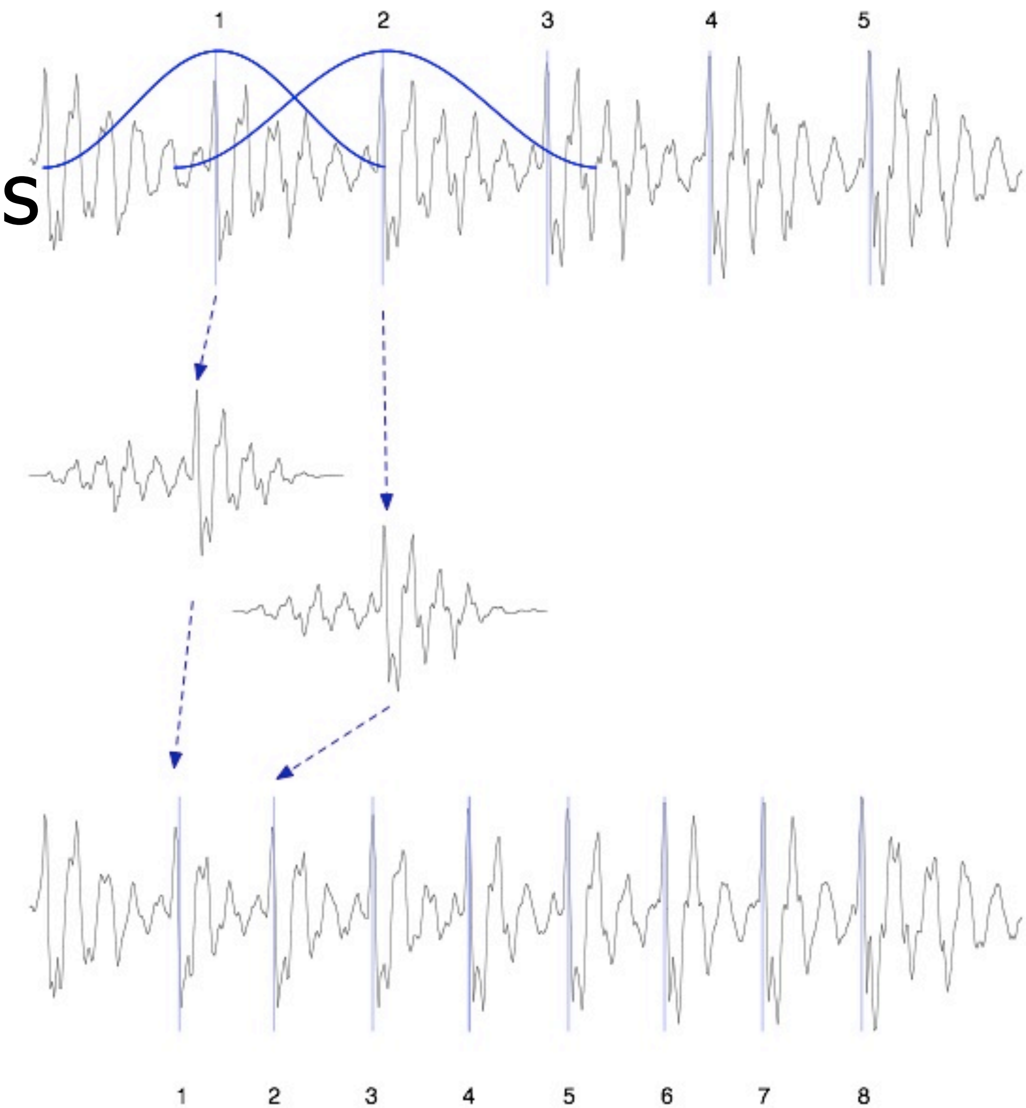
TD-PSOLA ™

- Time-Domain Pitch Synchronous Overlap and Add
- Patented by France Telecom (CNET)
- Very efficient
  - No FFT (or inverse FFT) required
- Can modify Hz up to two times or by half

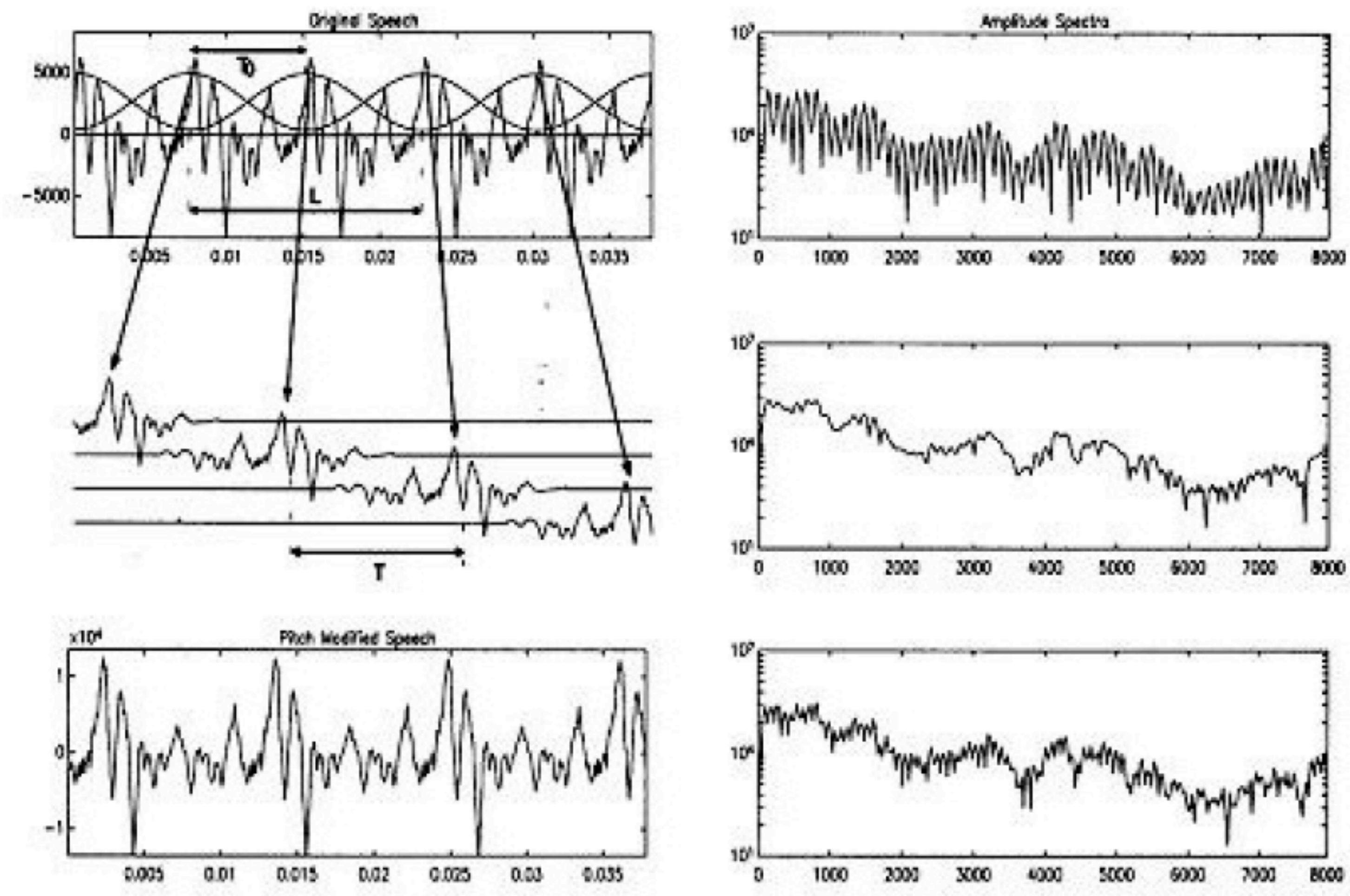
Slide from Richard Sproat

TD-PSOLA ™

- Windowed
- Pitch-synchronous
- Overlap-
- -and-add



TD-PSOLA ™



Thierry Dutoit

Summary: Diphone Synthesis

- Well-understood, mature technology
- Augmentations
  - Stress
  - Onset/coda
  - Demi-syllables
- Problems:
  - Signal processing still necessary for modifying durations
  - Source data is still not natural
  - Units are just not large enough; can't handle word-specific effects, etc

Problems with diphone synthesis

- Signal processing methods like TD-PSOLA leave artifacts, making the speech sound unnatural
- Diphone synthesis only captures local effects
  - But there are many more global effects (syllable structure, stress pattern, word-level effects)

Unit Selection Synthesis

- Generalization of the diphone intuition
  - Larger units
    - From diphones to sentences
  - Many many copies of each unit
    - 10 hours of speech instead of 1500 diphones (a few minutes of speech)
  - Little or no signal processing applied to each unit
    - Unlike diphones

Why Unit Selection Synthesis

- Natural data solves problems with diphones
  - Diphone databases are carefully designed but:
    - Speaker makes errors
    - Speaker doesn't speak intended dialect
    - Require database design to be right
  - If it's automatic
    - Labeled with what the speaker actually said
    - Coarticulation, schwas, flaps are natural
- "There's no data like more data"
  - Lots of copies of each unit mean you can choose just the right one for the context
  - Larger units mean you can capture wider effects

Unit Selection Intuition

- Given a big database
- For each segment (diphone) that we want to synthesize
  - Find the unit in the database that is the *best* to synthesize this target segment
- What does "best" mean?
  - "Target cost": Closest match to the target description, in terms of
    - Phonetic context
    - F0, stress, phrase position
  - "Join cost": Best join with neighboring units
    - Matching formants + other spectral characteristics
    - Matching energy
    - Matching F0

$$C(t_1^n, u_1^n) = \sum_{i=1}^{n} C^{target}(t_i, u_i) + \sum_{i=2}^{n} C^{join}(u_{i-1}, u_i)$$

Targets and Target Costs

- A measure of how well a particular unit in the database matches the internal representation produced by the prior stages
- Features, costs, and weights
- Examples:
  - /ih-t/ from stressed syllable, phrase internal, high F0, content word
  - /n-t/ from unstressed syllable, phrase final, low F0, content word
  - /dh-ax/ from unstressed syllable, phrase initial, high F0, from function word “the”

Slide from Paul Taylor

Target Costs

- Comprised of k subcosts
  - Stress
  - Phrase position
  - F0
  - Phone duration
  - Lexical identity
- Target cost for a unit:

$$C^t(t_i,u_i) = \sum_{k=1}^{p} w_k^t C_k^t(t_i,u_i)$$

Slide from Paul Taylor

How to set target cost weights (1)

- What you REALLY want as a target cost is the perceivable acoustic difference between two units
- But we can't use this, since the target is NOT ACOUSTIC yet, we haven't synthesized it!
- We have to use features that we get from the TTS upper levels (phones, prosody)
- But we DO have lots of acoustic units in the database.
- We could use the acoustic distance between these to help set the WEIGHTS on the acoustic features.

How to set target cost weights (2)

- Clever Hunt and Black (1996) idea:
- Hold out some utterances from the database
- Now synthesize one of these utterances
  - Compute all the phonetic, prosodic, duration features
  - Now for a given unit in the output
  - For each possible unit that we COULD have used in its place
  - We can compute its acoustic distance from the TRUE ACTUAL HUMAN utterance.
  - This acoustic distance can tell us how to weight the phonetic/prosodic/duration features

How to set target cost weights (3)

- Hunt and Black (1996)
- Database and target units labeled with:
  - phone context, prosodic context, etc.
- Need an acoustic similarity between units too
- Acoustic similarity based on perceptual features
  - MFCC (spectral features) (to be defined next week)
  - F0 (normalized)
  - Duration penalty

$$AC^t(t_i,u_i) = \sum_{i=1}^{p} w_i^a \, abs(P_i(u_n) - P_i(u_m))$$

Richard Sproat slide

How to set target cost weights (4)

- Collect phones in classes of acceptable size
  - E.g., stops, nasals, vowel classes, etc
- Find AC between all of same phone type
- Find C^t between all of same phone type
- Estimate w_{1-j} using linear regression

How to set target cost weights (5)

- Target distance is

$$C^t(t_i,u_i) = \sum_{k=1}^{p} w_k^t C_k^t(t_i,u_i)$$

- For examples in the database, we can measure

$$AC^t(t_i,u_i) = \sum_{i=1}^{p} w_i^a abs(P_i(u_n) - P_i(u_m))$$

- Therefore, estimate weights w from all examples of

$$AC^t(t_i,u_i) \approx \sum_{k=1}^{p} w_k^t C_k^t(t_i,u_i)$$

- Use linear regression

Richard Sproat slide

Join (Concatenation) Cost

- Measure of smoothness of join
- Measured between two database units (target is irrelevant)
- Features, costs, and weights
- Comprised of k subcosts:
  - Spectral features
  - F0
  - Energy
- Join cost:

$$C^j(u_{i-1},u_i) = \sum_{k=1}^{p} w_k^j C_k^j(u_{i-1},u_i)$$

Slide from Paul Taylor

Join costs

- Hunt and Black 1996
- If u_{i-1}==prev(u_i) C^c=0
- Used
  - MFCC (mel cepstral features)
  - Local F0
  - Local absolute power
  - Hand tuned weights

Join costs

- The join cost can be used for more than just part of search
- Can use the join cost for *optimal coupling* (Isard and Taylor 1991, Conkie 1996), i.e., finding the best place to join the two units.
  - Vary edges within a small amount to find best place for join
  - This allows different joins with different units
  - Thus labeling of database (or diphones) need not be so accurate

Total Costs

- Hunt and Black 1996
- We now have weights (per phone type) for features set between target and database units
- Find best path of units through database that minimize:

$$C(t_1^n, u_1^n) = \sum_{i=1}^{n} C^{target}(t_i, u_i) + \sum_{i=2}^{n} C^{join}(u_{i-1}, u_i)$$

- Standard problem solvable with Viterbi search with beam width constraint for pruning

$$\hat{u}_1^n = \underset{u_1, \ldots, u_n}{\operatorname{argmin}} C(t_1^n, u_1^n)$$

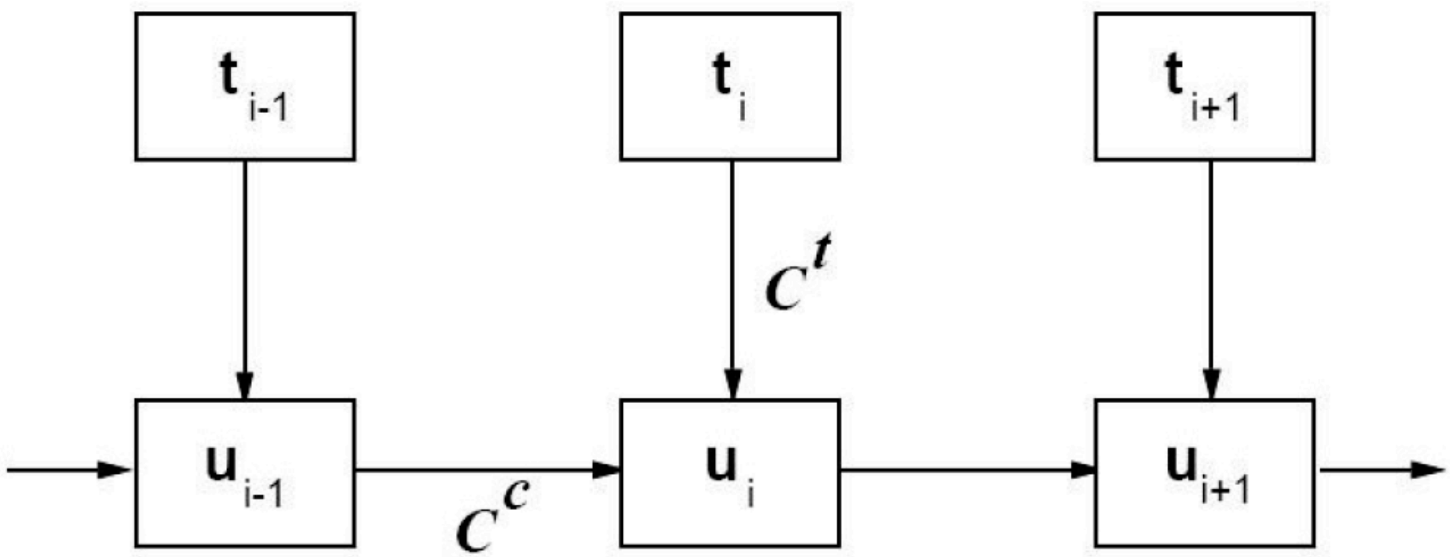
Slide from Paul Taylor

Improvements

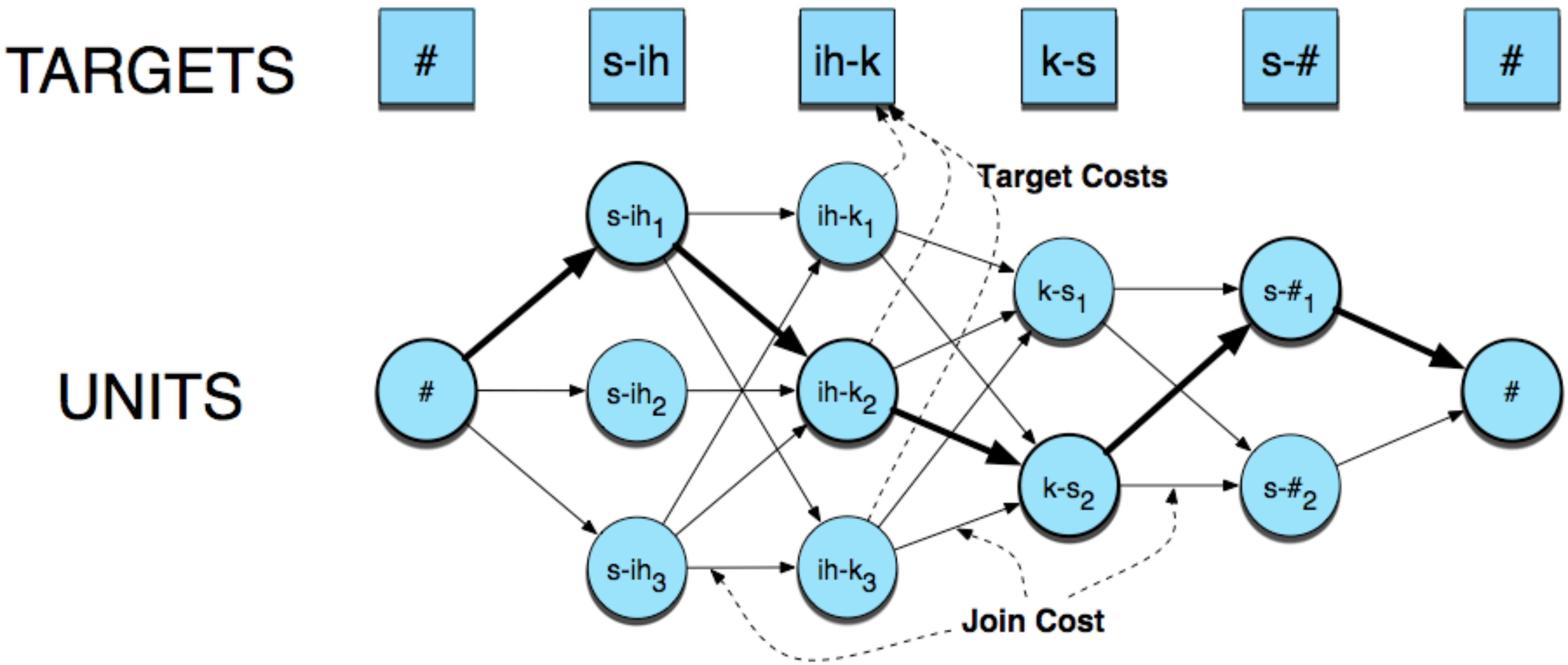
- Taylor and Black 1999: Phonological Structure Matching
- Label whole database as trees:
  - Words/phrases, syllables, phones
- For target utterance:
  - Label it as tree
  - Top-down, find subtrees that cover target
  - Recurse if no subtree found
- Produces list of target subtrees:
  - Explicitly longer units than other techniques
- Selects on:
  - Phonetic/metrical structure
  - Only indirectly on prosody
  - No acoustic cost

Slide from Richard Sproat

Unit Selection Search



Slide from Richard Sproat


TARGETS
#
s-ih
ih-k
k-s
s-#
#
Target Costs
UNITS
s-ih1
ih-k1
k-s1
s-#1
#
s-ih2
ih-k2
#
s-ih3
ih-k3
k-s2
s-#2
Join Cost

Database creation (1)

- Good speaker
  - Professional speakers are always better:
    - Consistent style and articulation
    - Although these databases are carefully labeled
  - Ideally (according to AT&T experiments):
    - Record 20 professional speakers (small amounts of data)
    - Build simple synthesis examples
    - Get many (200?) people to listen and score them
    - Take best voices
  - Correlates for human preferences:
    - High power in unvoiced speech
    - High power in higher frequencies
    - Larger pitch range

Text from Paul Taylor and Richard Sproat

Database creation (2)

- Good recording conditions
- Good script
  - Application dependent helps
    - Good word coverage
    - News data synthesizes as news data
    - News data is bad for dialog.
  - Good phonetic coverage, especially wrt context
  - Low ambiguity
  - Easy to read
- Annotate at phone level, with stress, word information, phrase breaks

Text from Paul Taylor and Richard Sproat

Creating database

- Unliked diphones, prosodic variation is a good thing
- Accurate annotation is crucial
- Pitch annotation needs to be very very accurate
- Phone alignments can be done automatically, as described for diphones

Practical System Issues

- Size of typical system (Rhetorical rVoice):
  - ~300M
- Speed:
  - For each diphone, average of 1000 units to choose from, so:
  - 1000 target costs
  - 1000x1000 join costs
  - Each join cost, say 30x30 float point calculations
  - 10-15 diphones per second
  - 10 billion floating point calculations per second
- But commercial systems must run ~50x faster than real time
- Heavy pruning essential: 1000 units -> 25 units

Slide from Paul Taylor

Unit Selection Summary

- Advantages
  - Quality is far superior to diphones
  - Natural prosody selection sounds better
- Disadvantages:
  - Quality can be very bad in places
    - HCI problem: mix of very good and very bad is quite annoying
  - Synthesis is computationally expensive
  - Can't synthesize everything you want:
    - Diphone technique can move emphasis
    - Unit selection gives good (but possibly incorrect) result

Slide from Richard Sproat

Recap: Joining Units (+F0 + duration)

- unit selection, just like diphone, need to join the units
  - Pitch-synchronously
- For diphone synthesis, need to modify F0 and duration
  - For unit selection, in principle also need to modify F0 and duration of selection units
  - But in practice, if unit-selection database is big enough (commercial systems)
    - no prosodic modifications (selected targets may already be close to desired prosody)

Joining Units (just like diphones)

- Dumb:
  - just join
  - Better: at zero crossings
- TD-PSOLA
  - Time-domain pitch-synchronous overlap-and-add
  - Join at pitch periods (with windowing)

Evaluation of TTS

- Intelligibility Tests
  - Diagnostic Rhyme Test (DRT)
    - Humans do listening identification choice between two words differing by a single phonetic feature
      - Voicing, nasality, sustenation, sibilation
    - 96 rhyming pairs
    - Veal/feel, meat/beat, vee/bee, zee/thee, etc
      - Subject hears "veal", chooses either "veal or "feel"
      - Subject also hears "feel", chooses either "veal" or "feel"
    - % of right answers is intelligibility score.
- Overall Quality Tests
  - Have listeners rate space on a scale from 1 (bad) to 5 (excellent) (Mean Opinion Score)
- AB Tests (prefer A, prefer B) (preference tests)

Huang, Acero, Hon

Recent stuff

- Problems with Unit Selection Synthesis
  - Can't modify signal
  - (mixing modified and unmodified sounds bad)
  - But database often doesn't have exactly what you want
- Solution: HMM (Hidden Markov Model) Synthesis
  - Won recent TTS bakeoffs.
  - Sounds unnatural to researchers
  - But naïve subjects preferred it
  - Has the potential to improve on both diphone and unit selection.
  - Is the future of TTS

HMM Synthesis, ~2007

- Unit selection (Roger)
- HMM (Roger)

- Unit selection (Nina)
- HMM (Nina)

Summary

- Diphone Synthesis
- Unit Selection Synthesis
  - Target cost
  - Unit cost
- HMM Synthesis