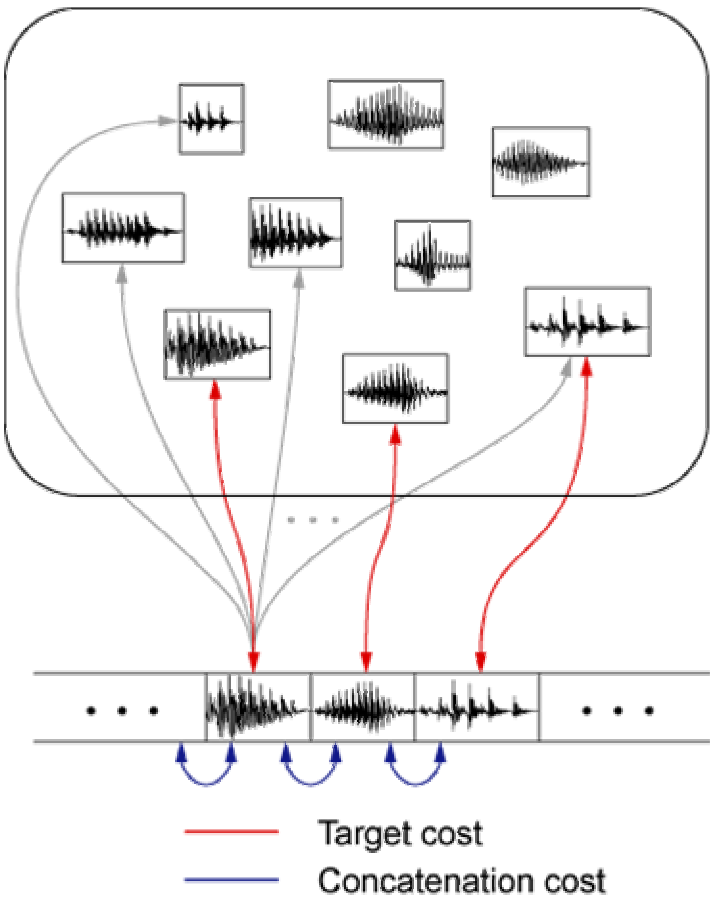
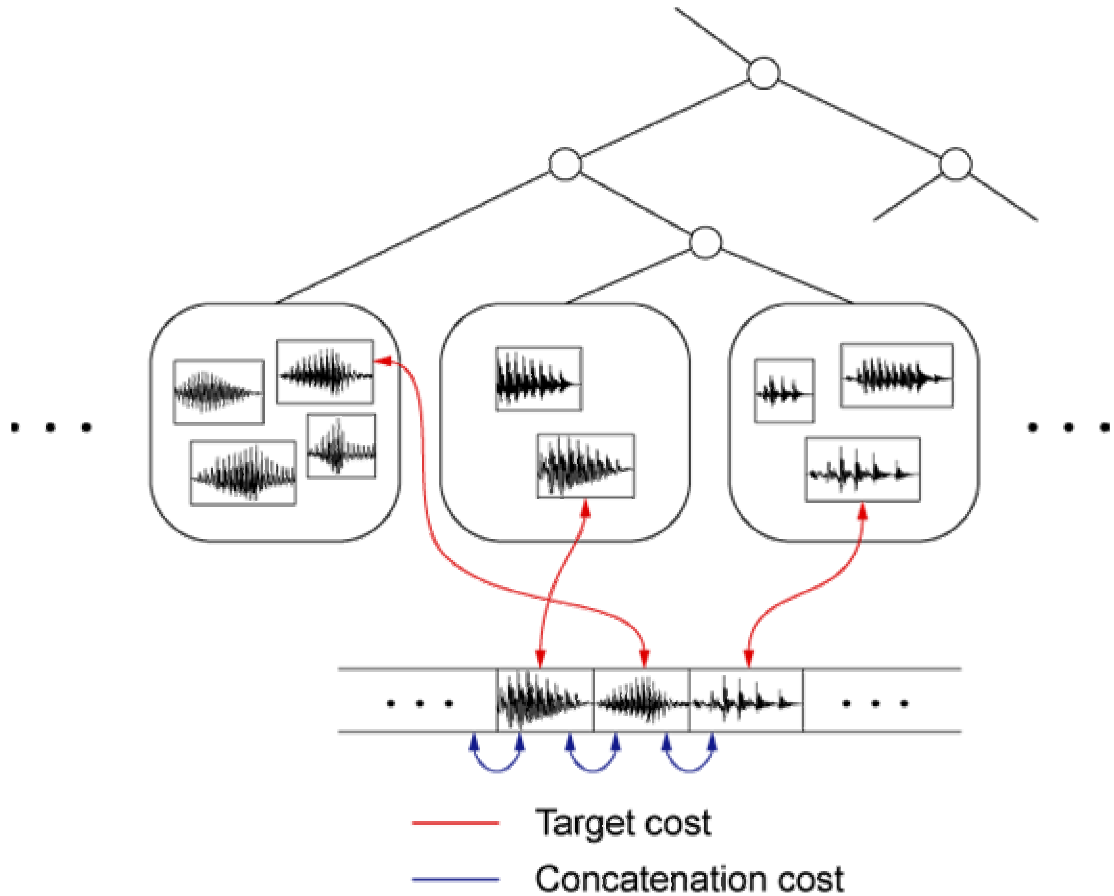
Statistical Parametric Synthesis
And voice conversion techniques

Unit Selection



Parametric Synthesis



Unit Selection vs SPS

- *Unit Selection*
  - *Can be very good*
  - *Requires large databases*
  - *One error can destroy a whole sentence*
- *Statistical Parametric Synthesis*
  - *Never very bad*
  - *Shown to be overall better (on average)*
  - *Resynthesis is problematic*

Nitech's HTS

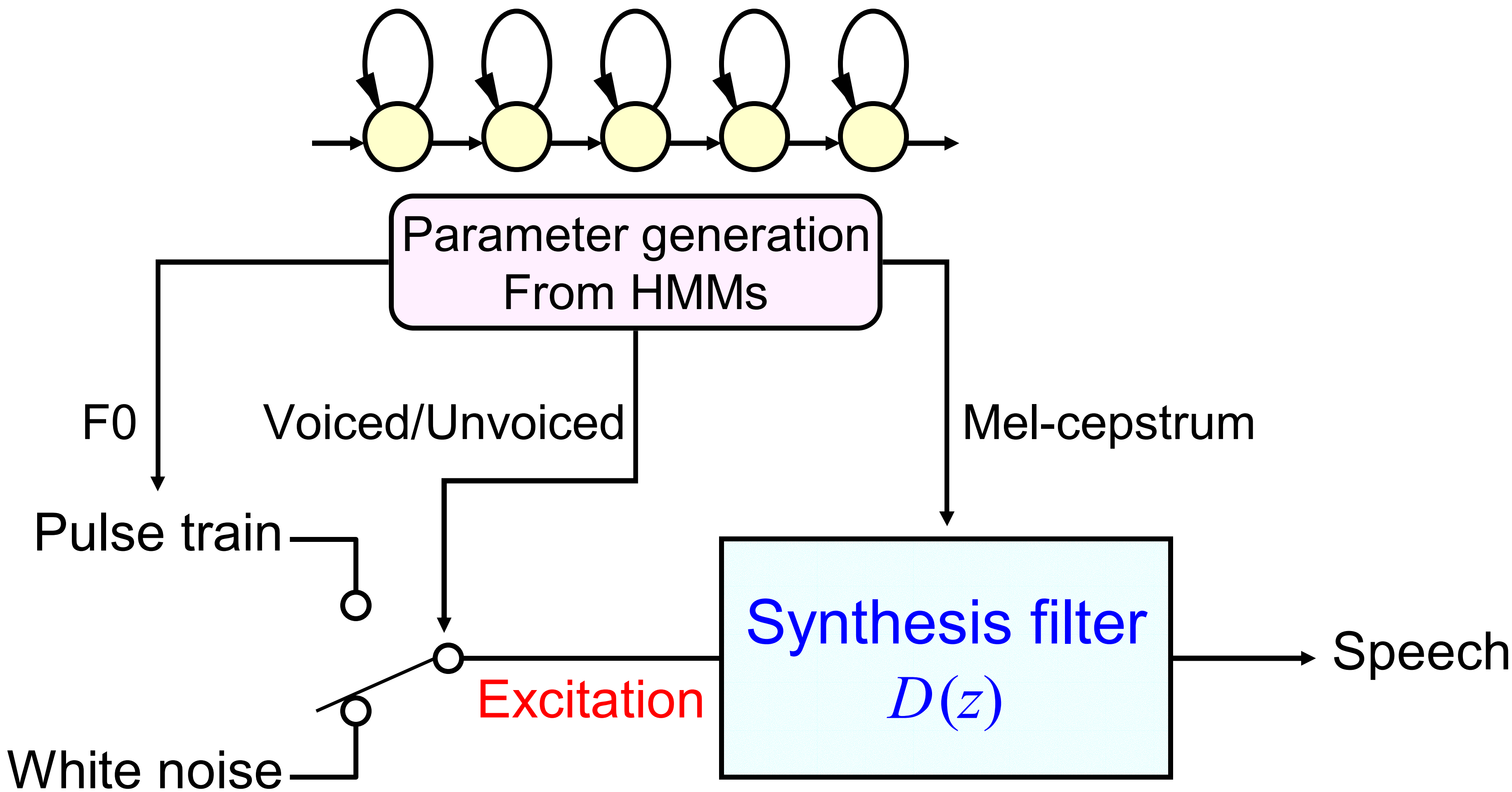
- *HTS from Nagoya Institute Technology*
  - *HMM generation synthesis*
  - *Train models from speech corpora*
  - *Cluster resulting HMM-states into trees*
  - *Generate parameters from trees*
  - *MLPG: find "best" generation path with dynamics*
  - *MLSA: Mel Cep resynthesis*
  - *(Fully Supported in Festival)*

Analysis/Resynthesis

- *Require reversible parameterization*
  - *MEL CEP (MLSA)*
  - *LSP*
  - *STRAIGHT (with residual models)*
  - *HNM (with noise/excitation models)*
- *Resynthesis:*
  - *Can sound buzzy and muffled*

Resynthesis by Vocoder

Mel-cepstal vocoder with pulse/noise excitation



Old vs New

Unit Selection:

- large carefully labelled database
- quality good when good examples available
- quality will sometimes be bad
- no control of prosody

Parametric Synthesis:

- smaller less carefully labelled database
- quality consistent
- resynthesis requires vocoder, (buzzy)
- can (must) control prosody
- model size much smaller than Unit DB

Synthesizer

Requires:
- Prompt transcriptions (txt.done.data)
- Waveform files (well recorded)

FestVox Labelling
- EHMM (Kishore)
- Context Independent models and forced alignment
- (Have used Janus labels too).

Parameter extraction:
- (HTS's) melcep/mlsa filter for resynthesis
- F0 extraction

Clustering
- Wagon vector clustering
- for each HMM-state name

Clustering by CART

Update to Wagon (Edinburgh Speech Tools).

- Tight coupling of features with FestVox utts
- Support for arbitrary vectors
- Define impurity on clusters of *N* vectors

$$(\sum_{i=1}^{24} \sigma_i) * N$$

Clustering

- F0 and MCEP
- Tested jointly and separately
- Features for clustering (51):
  - phonetic, syllable, phrasal context

Training Output

Three models:

- Spectral (MCEP) CART tree
- F0 CART tree
- Duration CART tree

F0 model:

- Smoothed extracted F0 through all speech
- (i.e. unvoiced regions get F0 values)
- Chose voicing at runtime phonetically

CLUSTERGEN Synthesis

Generate phoneme strings (as before)

For each phone:

- Find HMM-state names: ah_1, ah_2, ah_3
- Predict duration of each
- Create empty mcep vector to fill duration
- Predict mcep values from cluster tree
- Predict F0 value from cluster tree

Use MLSA filter to regenerate speech

Example CG Voices

7 Arctic databases:

1200 utterances, 43K segs, 1hr speech

awb	bdl
clb	jmk
ksp	rms
slt	

Scoring the results

Unit selection:

comparative listening tests

CLUSTERGEN

Mean Mel Cepstral Distortion over test set

$$10/\ln 10 \sqrt{2 \sum_{d=1}^{24} \left(mc_d^{(t)} - mc_d^{(e)} \right)^2}$$

MCD: Voice Conversion ranges 4.5-6.0

MCD: CG scores 5.0-8.0

smaller is better

Making it Actually Work

Engineering takes most of the time

Making it work for 10,000 utterances

Finding the best options:

- N *Using the most predictive samples*
  - *score samples based on predictability*
- N *Stepwise training*
- Y *Ensure mcep and F0 are aligned*
- Y *Use state duration in MCEP prediction*

…

Data size vs Quality

slt_arctic data size

Utts	Clusters	RMS F0	MCD	
50	230	24.29	6.761	
100	435	19.47	6.278	
200	824	17.41	6.047	
500	2227	15.02	5.755	
1100	4597	14.55	5.685	

More Examples

Cepstral: larger voices (3.5K utts)

David — *Diane*

Joint voices

awbslt — *rmsksp*

Non-English

German — *French*

SPS Advantages

- *More stable*
  - *Smaller dbs, and less accurate labeling*
  - *End footprint much smaller*
- *Parametric Domain*
  - *Adaptation: small amounts of data covert larger databases*
  - *Style, emotion, dialect, language*
- *ICASSP2007*
  - *Special session of SPS (6 papers from around the world)*

Voice Transformation

- *Don't collect lots of data*
  - *Collect 50 or so utterances*
  - *Convert an existing databases*
- *Requires (probably) parallel audio*
  - *But one side can be synthesized*
- *Can be used as a post-filter on a synthesizer*

Standard VC

- *Collect parallel examples*
- *Align them at the frame level*
  - *Using DTW*
- *Learn GMM (joint) model*
  - *From aligned parameters*
- *Requires vocoder resynthesis (buzzy)*

Building a VC model

- As post-filter to diphone (kal) voice
- See festvox/src/vc/HOWTO
- From Festvox Transformation Voice
  - Ensure ESTDIR and FESTVOXDIR are set
  - mkdir cmu_us_me
  - $FESTVOXDIR/src/unitsel/setup_clunits cmu us me
  - $FESTVOXDIR/src/vc/build_transform setup
  - $FESTVOXDIR/src/vc/build_transform default_us
  - Record files in etc/txt.transform.data
    - Use prompt_them or ensure waves in wav/*.wav
  - $FESTVOXDIR/src/vc/build_transform train (about 60 minutes)
  - $FESTVOXDIR/src/vc/build_transform festvox
  - festival festvox/cmu_us_me_transform.scm

    festival> (voice_cmu_us_me_transform)

    festival> (SayText “This is an example of the transformed voice”)

Voice Transformation

- Collect small amount of data
  - 50 utterances
- Adapt existing voice to target voice
- Adaptation: What makes a voice:
  - Lexical choice
  - Phonetic variation
  - Prosody
  - Spectral/vocal tract/articulatory movement
  - Excitation mode
- Use articulatory modeling for transformation (Toth)

Voice Transformation

- Festvox GMM transformation suite (Toda)

	awb	bdl	jmk	slt
awb				
bdl				
jmk				
slt				

Voice Transterpolation

- *Incremental conversion between voices*
  - *bdl-slt (male to female)*
  - *slt-bdl (female to male)*

Electromagneticarticulatograph