

Speaker Clustering for Multilingual Synthesis

Alan W Black and Tanja Schultz



Language Technologies Institute
Carnegie Mellon University

MULTILING, Stellenbosch, April 11, 2006

Multilingual Speech Synthesis

- Common technologies
 - (Diphone: too hard to record and label, coverage)
 - Unit selection is the standard method: *select appropriate sub-word units from large database of natural speech*
 - CONS: too much to record and label
 - Requires 200M per voice (single model across languages)
- New technology: Parametric Synthesis “clustergen (CG)”
 - HMM-generation based synthesis
 - Cluster units to form models, Generate from models
 - PRO: can work with little speech (10 minutes)
 - CONS: speech sounds buzzy, lacks natural prosody
 - Requires 2M per voice (single model across languages)

Unit Selection vs Parametric Synthesis

- Unit Selection: 
 - large carefully labeled database (often 5000+ utts)
 - hard to speak 5000 good utterances > professionals
 - quality good when good examples available but rapid degradation when out-of-domain
 - little or no prosodic modifications
 - natural delivery and multiple speakers desired to model speaker variability!!!
- Parametric Synthesis: 
 - smaller less carefully labeled database
 - quality consistent
 - resynthesis requires vocoder, (buzzy)
 - can (must) control prosody
 - model size much smaller than Unit DB

HMM-Generation Synthesis

- NiTech's HTS (Tokuda et al.)
 - Built as modification of HTK
 - FestVox build process exists
 - Hard coded feature values
- HMM-Generation Synthesis
 - Sub-phonetic features are modeled not as set of instances of units but parametric models
 - View these clusters as averages of instances
- High quality understandable speech (Bennett, 2005)
- Language Independence (Tokuda, 2002)
- Multilingual databases (e.g. GlobalPhone) within HMM-generation synthesis (Latorre&Furui, 2005)

ClusterGen

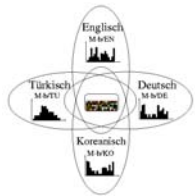
- New synthesis technique added to the FestVox suite
 - Clustering technique for HMM-state sized segments
- Training data is HMM-based labeled speech
 - Labeling system included in FestVox
 - Janus RTk labels are used created by forced alignment
- **CLUSTERGEN**
 - Reversible (analysis/synthesis) parameterization of speech
 - MCEP analysis and MLSA filter for resynthesis (as in HTS)
 - 24-dim MCEP feature vectors
 - Clustered using wagon CART tree builder
 - Features for tree building are the articulatory features derived from GP IPA-based global phone inventory
 - Cluster optimization:
 - minimize sum of SD of each MCEP feature
 - weight by the number of samples in the cluster

Universal Sound Inventory & Data

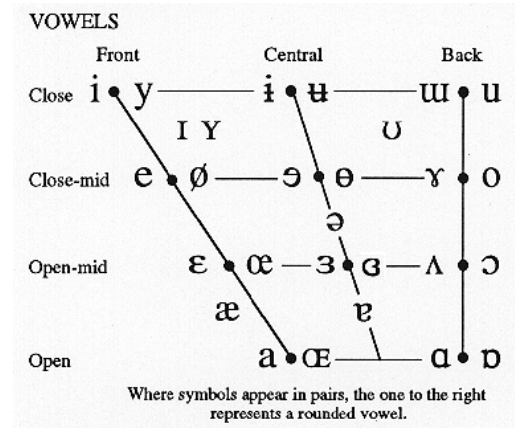
Speech Production is independent from Language \Rightarrow IPA

1) IPA-based **Universal Sound Inventory**

2) Each sound class is trained by **data sharing**



- Reduction from 485 to 162 sound classes
- *m,n,s,l* appear in all 12 languages
- *p,b,t,d,k,g,f* and *i,u,e,a,o* in almost all



GlobalPhone

- Use 10 languages
Ch-Mandarin, Croatian, German, Japanese, Portuguese, Russian, Spanish, Swedish, Turkish + English (WSJ0)
- Use ASR global sound inventory
- Use IPA acoustic features

Clustering by CART

- Update to Wagon (Edinburgh Speech Tools)
- Clustering
 - F0 and MCEP, tested jointly and separately
 - Features for clustering (51): IPA articulatory features + other phonetic, syllable, phrasal context
- Training Output - Three models:
 - Spectral (MCEP) CART tree
 - F0 CART tree
 - Duration CART tree
- F0 model:
 - Smooth extracted F0 through all speech (i.e. unvoiced regions get F0 values)
 - Chose voicing at runtime phonetically

FestVox CLUSTERGEN Synthesizer

- Prompt transcriptions, Waveform files (well recorded)
- Labeling
 - Used CI models and forced alignment (JRtk – monolingual ASR)
- Parameter extraction:
 - (HTS's) MCEP/MLSA filter for resynthesis
 - F0 extraction
- Clustering
 - Wagon vector clustering for each HMM-state name
- 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 and F0 values from corresponding cluster trees
 - Use MLSA filter to regenerate speech

Measuring Quality

Mean Mel Cepstral Distortion (MCD) over test set
(smaller is better)

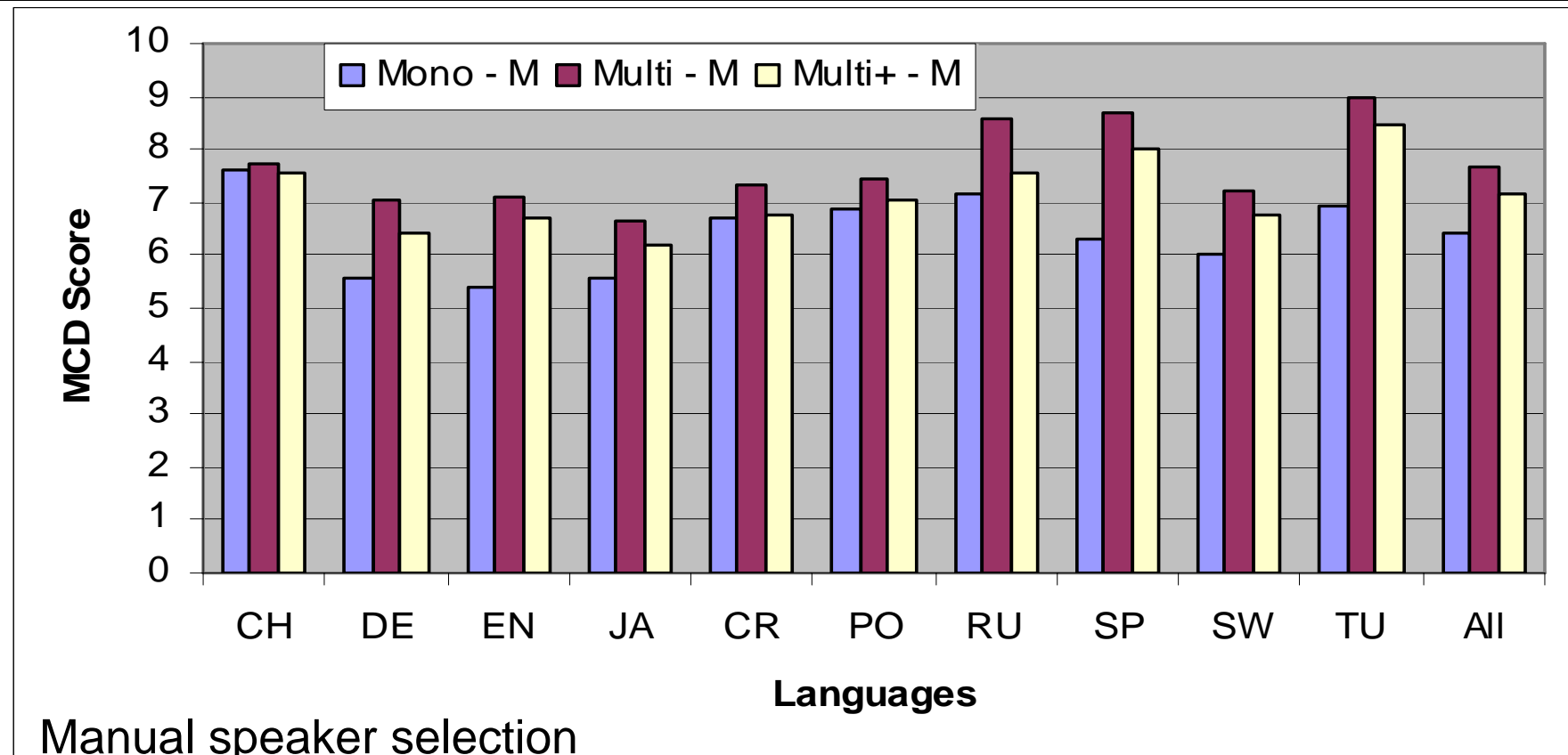
$$10 / \ln 10 \sqrt{2 \sum_{d=1}^{24} \left(mc_d^{(t)} - mc_d^{(e)} \right)^2}$$

Measured on a Cross evaluation set

MCD: Voice Conversion ranges 4.5-6.0

MCD: ClusterGen scores 5.0-8.0

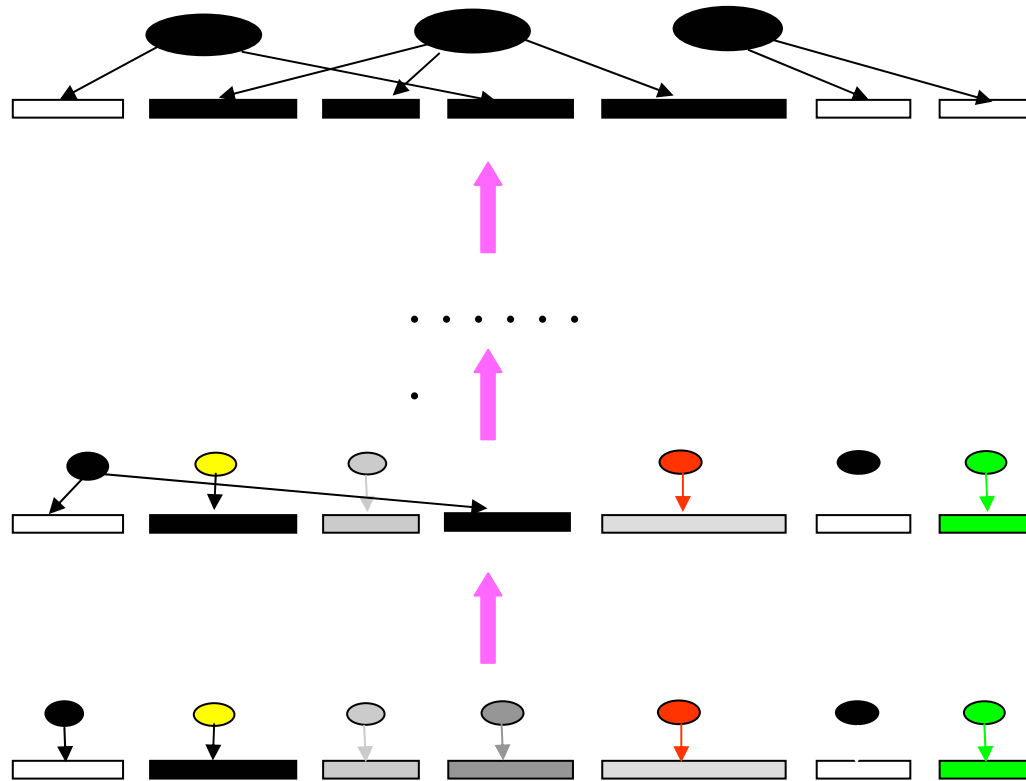
Mono vs Multilingual Models



- ⇒ For all languages monolingual TTS performs best
- ⇒ Multilingual Models perform well ...
 - ... only if knowledge about language is preserved (Multi+)
 - (only small amount of sharing actually happens)

Speaker Clustering

Hierarchical Bottom-up Clustering



BIC stopping criterion

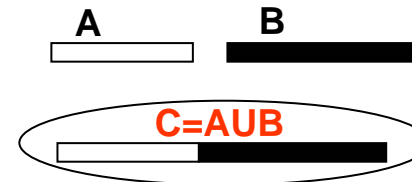
$$\Delta BIC = BIC(M_C) - BIC(M_{A,B})$$

$$BIC(M) = \log L(X | \mu, \Sigma) - \frac{\lambda}{2} V(M) \log N$$

$$M_{A,B}: X_A \sim N(\mu_A, \Sigma_A) \quad X_B \sim N(\mu_B, \Sigma_B)$$

$$M_C: X_C \sim N(\mu, \Sigma)$$

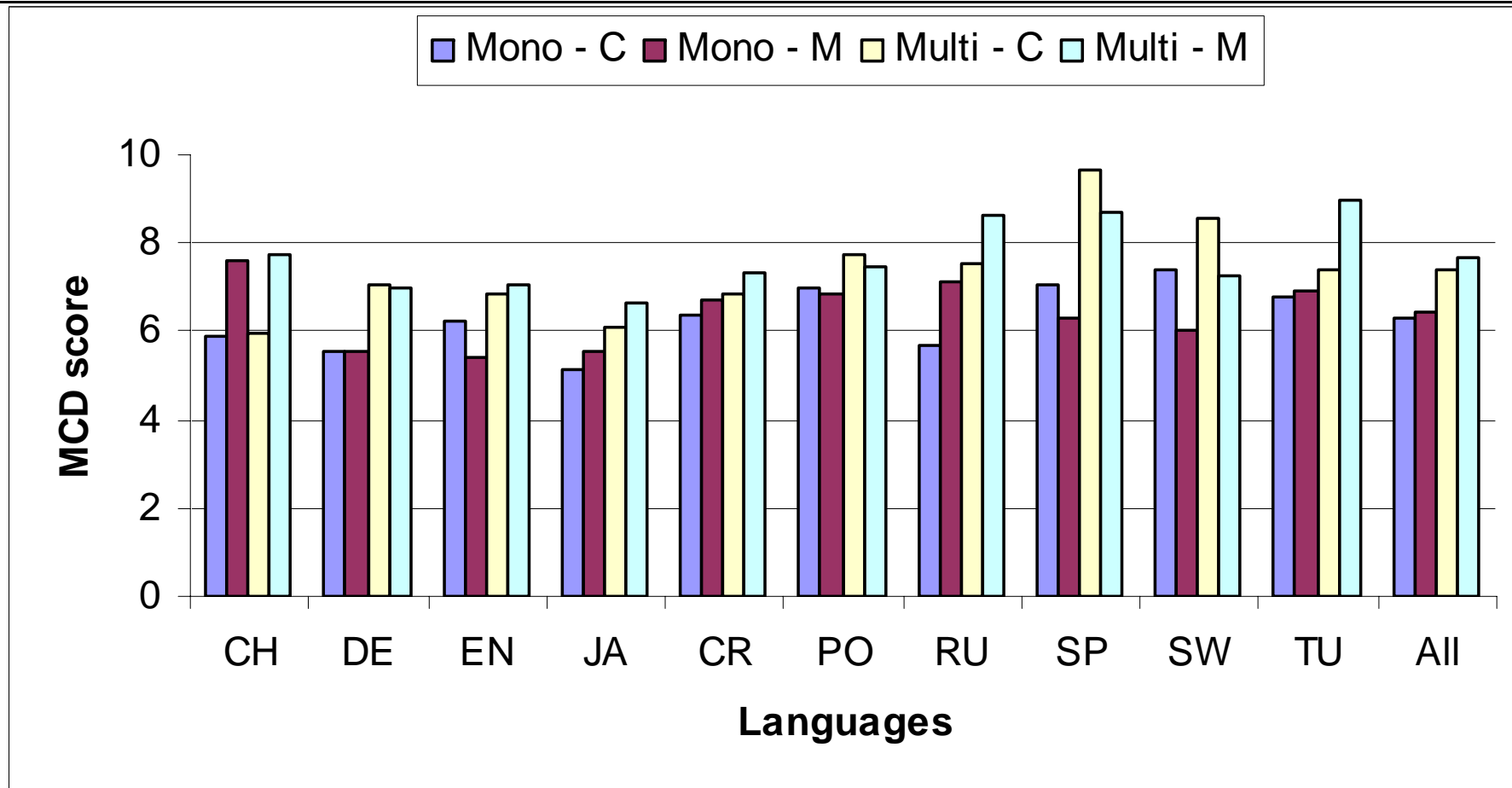
TGMM-GLR distance



$$D(S_A, S_B) = -\log \frac{P(X_{A \cup B} | \theta_{A \cup B})}{P(X_A | \theta_A) P(X_B | \theta_B)}$$

$$= \log \frac{P(X_A | \theta_A) P(X_B | \theta_B)}{P(X_C | \theta_C)}$$

Manual vs Clustered Speaker Selection



- ⇒ Selecting similar speakers helps for both Mono and Multi
- ⇒ Multi benefits more as expected: similarity more important
- ⇒ Large variation across languages (label quality? Data size?)

Conclusion & Future Work

- ClusterGen allows for much smaller voices
 - Multilingual voice for unit selection: 200Mb
 - Multilingual voice for ClusterGen:2Mb
- Preserving language information (Multi+) helps
- Selecting similar speakers helps
- All voices are understandable (no formal tests!)

Future Work:

- ClusterGen is *very* young
 - No treatment of dynamics yet
 - Multilingual, Multispeaker DBS
 - Grapheme Based models
 - Voice/Style conversion: model interpolation
 - Signal reconstruction: proper residual modeling