# The Blizzard Challenge 2006 CMU Entry introducing hybrid trajectory-selection synthesis

*John Kominek, Alan W Black*

Language Technologies Institute
Carnegie Mellon University, USA
{jkominek, awb}@cs.cmu.edu

## Abstract

Acknowledging the lessons of Blizzard Challenge 2005 – that smooth prosodic cadence supersedes spectral resolution – but wanting a system devoid of vocoding artifacts – we introduce a hybrid trajectory-selection synthesizer. Using a parametric synthesizer to generate a pitch-synchronous sequence of F0/duration/power and spectral vectors, this trajectory serves as the target cost function for a unit selection synthesizer. The combination can unify the best attributes of two distinct categories of synthesizers, provided that the feature representation supports both. To this end, we also introduce a new perceptually-weighted harmonic representation of speech that is pitch-synchronous and retains phase information.

## 1. Introduction

The *Blizzard Challenge* was conceived to fill a gap in the speech synthesis research community: namely, the need for a uniform mechanism for measuring the utility of competing techniques that may be applied to the problem of synthesis [1]. The first edition, held in 2005, not only achieved that goal, but provided a powerful lesson to TTS researchers. We summarize this lesson as "continuity beats everything else." Basically, speech when delivered with a smooth tempo and without any glaring discontinuities more than compensates for lack of spectral brightness. We take as convincing evidence the superior evaluation of HTS-based synthesis [2], handily outperforming a contingent of unit selection synthesizers.

The outcomes of last year's Challenge has effected a reversal of outlook within the Carnegie Mellon synthesis group. Previously we took as our starting point a unit selection synthesizer – as proven technology capable of high quality – and sought to reduce the incidence of join errors and pitch discontinuities. From the opposite outlook we have instead adopted a parametric synthesizer as our starting point (having excellent smoothness but also the unfortunate buzziness characteristic of vocoded speech), and have sought to improve its spectral resolution, hence naturalness.

This has led us to develop a new architecture that may be considered a hybrid of spectrum-based parametric synthesis and of unit selection. To denote this new approach we use the term "trajectory-selection" synthesis; short for "trajectory-defined target-cost-based unit-selection" synthesis. To clarify: this means that selection of units (however defined) from a database is made on the basis of a detailed time sequence of speech parameters, or *trajectory*. This differs from usage in the term "unit-selection," in which the *units* of speech are the entities being selected.

While a parametric synthesizer such as HTS [2] and ClusterGen [3] can generate reasonable sounding speech, we propose the idea of using it as a front-end synthesizer to generate an *intermediate specification* of speech, to be fed into a subsequent synthesizer capable of higher quality waveform generation. The intermediate specification is a time sequence of parameter vectors. Specifically, they are pitch, power, and a perceptually weighted form of cepstrum. Such a temporal trajectory tightly constrains the utterance's prosodic contour, and provides a template to the backend synthesizer.

What is the nature of the backend synthesizer? Two approaches can be taken. In one, the trajectory provides an explicit target cost to a unit selection synthesizer. The selected units, however, will not perfectly match the pitch and duration of the template, thereby requiring post processing to maintain the specified prosody. This extra step is inefficient. Alternatively, the backend synthesizer can instead match the specified prosody using the intermediate parametric representation, then generate speech during the final stage. Though quality suffers compared to playing back recorded snippets of speech, full flexibility is maintained throughout the processing pipeline.

In this paper section 2 describes our synthesizer's novel architecture, explaining how it fulfills our design goals. Highlighted are the three processing stages of prosody modeling, unit search, and waveform generation. A comparison is made to traditional parametric and unit selection architectures, demonstrating that our architecture is a combination of the two forms. To meet our goals we found it necessary to develop a novel pitch-synchronous, harmonic encoding of speech. Our feature representation is described in section 3. The key difference between the frontend and backend synthesizers contained in our system is that the latter incorporates Fourier phase information. Section 4 briefly discusses results.

## 2. System Architecture

### 2.1. Design Criteria

Our effort is guided by four key design criteria.

1. The prosodic pattern of speech must be judged by human listeners as smooth and continuous. We correlate this perception to the duration of phones matching listener expectations (primarily), without unusual changes in speed or spectral discontinuities (emblematic of join mismatches in a unit selection synthesizer).

2. The generated pitch contour must have appropriate descension at phrase-final locations, and possess sufficient sub-word micro-variation to be nearly natural in pattern.

3. The spectral resolution must be high enough that the voice sound "bright" – nearly as clear sounding as recorded speech. Conversely, the voice should be free from the "buzziness" characteristic of vocoded speech, such as afflicts LPC-based generation.

4. The build process must contend with automatically generated features: for example, phone boundary labels, pitchmarks and F0 determinations. This rules out labor-intensive database corrections that are one of the hallmarks of commercial-grade synthesizers.

At face value, these criteria are mutually exclusive. The desire for human-sounding speech (3) normally mandates a unit selection approach, but full control over duration and pitch normally requires a parametric synthesizer (1,2). Hence the desire for a hybrid approach that combines the best attributes of each, while being robust to errors (4).

### 2.2. Architectural Comparisons

To establish a frame of comparison, we review two synthesis methods available in Festival: a traditional cluster-organized unit-selection approach ("clunits" [4]) and a parametric synthesizer newly incorporated into the software distribution ("ClusterGen" [3]).

In *clunits* the speech is preprocessed into 12D cepstral vectors, which are organized into thousands of clusters of similar units (roughly equivalent to allophonic phoneme variants), where a cluster has around 20-40 units each. This processing is performed by a CART tree learner. During this offline training, the learner produces a decision tree that is used during synthesis to predict a cluster identifier for each phoneme in the utterance. In isolation, each unit of a cluster is considered equally good. One particular example is selected as the best using a Viterbi search to find the sequence of units that minimizes total join costs. The units (wavefile segments) are concatenated together with inter-frame smoothing.

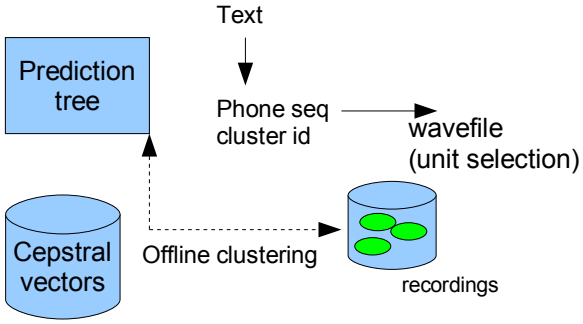
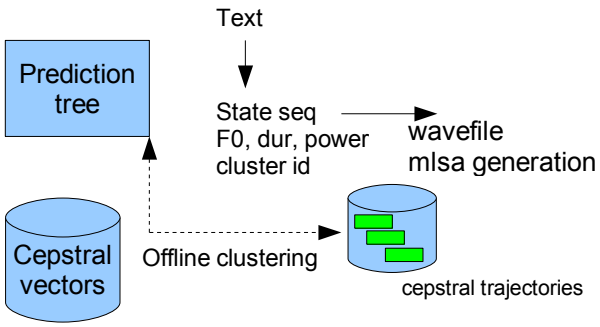



**Figure 1.** Unit selection architecture. The green ellipses denote clusters of speech units (allophonic variations).

Similarly, construction of a ClusterGen voice involves offline segmentation of speech into clusters. There are three differences: 1) the cluster element is not a set of waveform segments, but is a trajectory of cepstral vectors, 2) the trajectory is the average of all contributing units, and 3) the units are HMM-state length segments. During synthesis, CART trees generate a sequence of phoneme states with predicted duration, power, F0, and cluster id. An utterance's parameter sequence is fed to a synthesizer capable of modifying pitch and inverting cepstral vectors. For this ClusterGen currently uses the establish algorithm of MLSA (melcep log spectral approximation [5]). MLSA converts a spectral vector to an LPC filter representation, and sends as an excitation signal either a pulse for voiced speech, or white noise for unvoiced speech.

**Figure 2.** (below) Parametric synthesis architecture. The green rectangles denote sequences of feature vectors that substitute for the clusters of Figure 1.




### 2.3. Hybrid Trajectory-Selection Synthesis

Approximately considered, our trajectory-selection system affixes a unit selection synthesizer onto a ClusterGen front-end. An utterance is processed in four stages, beginning with the prosodic prediction of duration, pitch, and power, along with a cluster id for each phone-state. This is converted into a trajectory track of spectral vectors. At this stage it is possible to generate a waveform. Instead, the trajectory is passed to a search module, which re-segments the state-based trajectory into a sequence of diphones. These diphones serve as target costs to the unit selection process. By comparison, in a clunits synthesizer, all units within a cluster have zero target cost, while those outside the selected cluster have infinite cost.

At this stage a concatenative waveform can be generated. This picks up whatever pitch contour that the segment has, which often does not fit well with the surrounding context. To maintain control over the pitch contour, the parametric representation of the selected diphones is passed to a prosody modification module. It modifies the diphones to match the specified pitch contour. Then the waveform is generated.

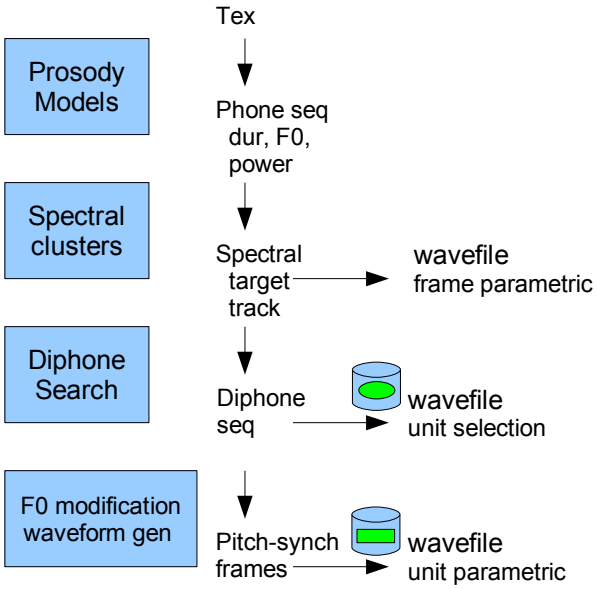



**Figure 3.** The hybrid trajectory-selection architecture.

Unique to this hybrid trajectory-selection synthesizer is that it can generate three distinctly different waveforms from identical input, though only the third and final variant is considered the "real" output.

Another notable aspect is that the parametric representation has to serve the multiple purposes of clustering and unit selection, pitch modification, and waveform generation. In ClusterGen this is provided through the dual

cepstral/LPC representation supported by MLSA. However, the resulting vocoded speech is excessively buzzy. To overcome this deficiency while maintaining flexibility, we've developed a pitch-synchronous harmonic representation of speech. This choice emerged from the observation that a synthetic voice considered "bright and lively" requires accurate phase reconstruction of the harmonic components.

## 3. Feature Representation and Manipulation

In the speech representation known as Harmonic plus Noise Modeling (HNM [6]), speech is decomposed into spectral bands and classified as either voiced or unvoiced. Voiced bands are encoded and decoded using a harmonic (Fourier) representation, while unvoiced bands are generated by applying a filter to a white noise generator. Speech is reconstructed as the sum of these two components. In contrast, we forgo the harmonic/noise distinction and treat all Fourier components up to the Nyquist frequency as harmonics. No inherent distinction is made between voiced and unvoiced speech, except that during the pitch-marking stage of analysis, imaginary marks are inserted into silence and unvoiced sections.

### 3.1. Perceptually Weighted, Pitch Synchronous, Principal Component, Harmonic Analysis

Our front-end speech analysis module is unique to this work, and so we briefly state four motivating considerations.

a) To enable convenient manipulation of F0 contours, our feature representation is pitch synchronous. b) To preserve perceptual fidelity, all harmonic phase components up to the Nyquist frequency (8 kHz) are saved. c) To support perceptually relevant distance comparisons, the absolute power spectrum is warped on the Mel frequency scale and reshaped according to equal loudness contours. d) To provide a compact representation of the power spectrum, dimensionality reduction is performed through principal components analysis. PCA is performed separately on each corpus of speech data. Consequently, unlike cepstral analysis, our representation is speaker-specific.

In outline, the analysis stage consists of 12 steps. A more thorough development is available in [7].

1. Speech is segmented into pitch periods. The silence and unvoiced section are divided evenly into periods such as to yield a fake F0 maximally close to the speaker's global mean.
2. An analysis frame is defined as a single pitch period with a rectangular window applied. Except for places of rapid transition, simple frame repetition reproduces steady state speech. As an alternative, a frame is defined as three contiguous pitch periods with a Blackman window centered over the extent.
3. The frame is transformed into frequency space using a DFT. The analysis length is equal to the number of samples in the frame. We do not pad the frame with zeros out to a length that is a power of 2. The speed advantage of an FFT is sacrificed to avoid introducing length mismatch artifacts.
4. Compute the power spectrum and perform "Bayesian weighting" of the result with the zero vector. Bayesian weighting computes a modified power spectrum $\boldsymbol{x}'$ by averaging it with a prior vector $\boldsymbol{y}'$ based on the probability $\boldsymbol{p}$ that the signal is reliable, according to a measurement $\boldsymbol{f(x)}$, such that $\boldsymbol{x}' = p(f(\boldsymbol{x}))\boldsymbol{x} + (1 - p(f(\boldsymbol{x})))\boldsymbol{y}$. We use a measure based on the loudness of the frame $\boldsymbol{x}$ (strong signals are more reliable than weak). This concentrates quiet frames into a tight cluster, and prevents discretization errors from contaminating speech with frames from silence segments. Our treatment is a generalization of *dithering* used in speech recognition systems to prevent the occurrence of digital zeros, which cannot be allowed on account of the next step.
5. Convert the modified power spectrum to the decibel scale by taking $\log_{10}(\boldsymbol{x})$.
6. Normalize the loudness of the spectral components according to the psycho-acoustic measurments of the ISO226 standard [8]. We use the 60 dB reference curve for normalization.
7. Upsample the power spectrum to 1001 points. This is done because the next step is to process the spectrum through a filterbank on the mel frequency scale. Due to the nonlinear warping of the mel rescaling, the lowest order filterbank suffer from too few samples without compensation.
8. Transform the upsampled spectrum to the mel frequency scale and compute a K bin filterbank vector. High fidelity reconstruction requires K=48 or 64; the more typical value of 24 is insufficient.
9. The filterbank output is converted from a log base 10 scale to a log base 2 scale, otherwise known as. the *sone* scale in the psycho-acoustic literature. It is known to provide a better scale for computing perceptual differences.
10. To prepare for the next processing step the mean is removed from each sone-scale filterbank vector. Mean removal leaves the representation invariant under loudness transformation
11. The entire corpus of mean-removed filterbank vectors undergo PCA transformation to decorrelate the data. The principal components are stored in a file containing the new basis functions. The first N components of the transformed vectors are written out. When there are K=64 filterbank values, N=24 PCA components is sufficient for high fidelity reconstruction.
12. The Fourier phase components are unwrapped and spline-resampled at 50Hz intervals, or equivalently to 161 points. Converting to equal length vectors permits PCA transformation of phase.

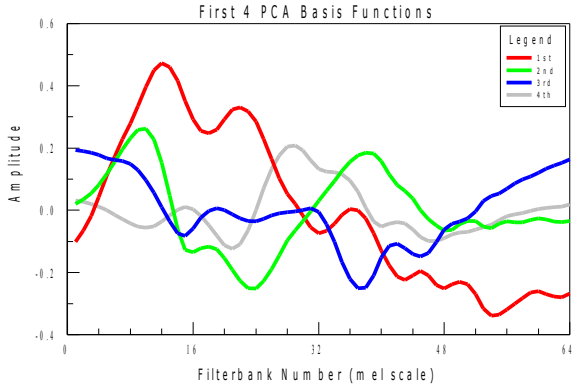



**Figure 4**. The first four basis functions of the Blizzard voice.

Figure 4 plots the first four spectral basis functions for the Blizzard 2006 recordings. Similar to a DCT, lower order functions capture broad spectral patterns.

### 3.2. Criticality of Accurate Phase

To investigate the relative importance of spectral versus phase information, we performed analysis-resynthesis on a set of 10 Blizzard wavefiles. The number of principal components used for each were systematically varied, as laid out in Table 1. In this table we identified regions of comparable quality. Interestingly, one can "get away" with relatively few spectral components (i.e. 12) but not phase components (48). Insufficiently accurate phase for reconstruction leads to throaty/breathy distortions.[1]

| | | *Number of phase components* | | | | | | |
|---|---|---|---|---|---|---|---|---|
| *Number spectral comp.* | | 8 | 16 | 24 | 32 | 48 | 64 | 80 |
| | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | 4 | 1 | 1 | 1 | 1 | 1 | 1 | 2 |
| | 8 | 1 | 1 | 1 | 1 | 2 | 2 | 3 |
| | 12 | 1 | 1 | 1 | 1 | 2 | 3 | 4 |
| | 16 | 1 | 1 | 2 | 2 | 3 | 4 | 5 |
| | 24 | 1 | 1 | 2 | 3 | 4 | 5 | 5 |

**Table 1.** Approximate regions of equal quality for analysis-resynthesis: 1 = poor, 5 = excellent.

In another analysis-resynthesis experiment, the spectral components were fully reconstructed but with the harmonic phase randomized with a uniform distribution of $\pm n\pi$ radians. When completely randomized (n=1), the resulting speech is heavily "breathy." A threshold of n=10% is just barely perceptible and n<20% is acceptable. Setting the phase to zero everywhere results in highly "buzzy" speech, especially during unvoiced fricatives.

### 3.3. Pitch Alteration using spline interpolation

Supporting prosodic modification without resorting to LPC excitation is a key motivation for our unusual feature representation. Vowel and fricative durations can be altered without significantly affecting pitch by frame deletion and insertion. Pitch can be modified by interpolating the harmonic representation (and adding/dropping frames to maintain word durations, if desired). We accomplish this by fitting a spline curves in the spectral/phase feature space, then resampling at integer multiples of the new fundamental frequency. The extent of modification can be over 100% (i.e. doubling or halving F0), but operates best within about plus or mind 25%. Processing introduces some distortion into the regenerated waveform, including discontinuities at frame boundaries, which can be partially covered up through overlap-and-add smoothing. The generated F0 contour can be linear, Tobi specified, or CART tree specified with sub-syllable micro fluctuations.

### 3.4. Target cost evaluation

In the processing pipeline of Figure 3, the predicted spectral trajectory provides a target to evaluate each candidate diphone. Since the target and candidate are usually of different length, this discrepancy needs to be accounted for. We adopted the linear time warping of [9]. Specifically, given two vectors $\boldsymbol{U}$ and $\boldsymbol{V}$ the distance $\boldsymbol{D}$ between them is defined to be

$$D(U,V)=P\frac{|U|}{|V|}\sum_{i=1}^{|U|}\sum_{j=1}^{n}\frac{W_j}{n|U|}\left|\left(F_{ij}(U)-F_{kj}(V)\right)\right|,\ k=i\frac{|V|}{|U|}$$

where $P$ is a duration penalty factor (here, set to 1) and $Wj$ values weight the spectral component's contribution to the distance (also all set to 1).

## 4. Evaluation and Conclusions

The Blizzard 2006 evaluation involved creation of two voices: one built from full database and one from the Arctic subset. For the full set, we submitted a clunits voice with only slight modifications from our Blizzard 2005 entry. For the Arctic subset we submitted the work described here. The overall MOS score for the full unit-selection system was 2.5 (a drop from 2.9 of the nearly identical entry of last year). The overall MOS score for the arctic voice was 2.1.

In a postmortem analysis we've identified the distance equation as the likely culprit causing under-performance. Our explanation is that it doesn't sufficiently constrain duration of the candidate units, thereby undoing much of the guidance provided by the target trajectory. Current investigations indicate that more appropriate distance measures substantially improve voice quality.

## 5. Acknowledgments

This work is in part supported by the US National Science Foundation under grant number 0415021 "SPICE: Speech Processing Interactive Creation and Evaluation Toolkit for new Languages." Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation. The authors express thanks to Tina Bennett, David Huggins-Daines, Brian Langner, and Arthur Toth for participating in group discussions.

## 6. References

[1] Black, A. and Tokuda, K. *Blizzard Challenge 2005: Evaluating corpus-based speech synthesis on common databases*, InterSpeech 2005. Lisbon, Portugal.

[2] Zen, Heiga, Toda, Tomoki. *An Overview of Nitech HMM-Based Speech Synthesis System for Blizzard Challenge 2005*. InterSpeech 2005, Lisbon, Portugal.

[3] Black, A. C*LUSTERGEN: A Statistical Parametric Synthesizer using Trajectory Modeling.* InterSpeech 2006, Pittsburgh, USA.

[4] Black, A. and Taylor, P. *Automatically clustering similar units for unit selection in speech synthesis.* Proceedings of Eurospeech 97, vol2 pp 601-604, Rhodes, Greece.

[5] Fukada, T., Tokuda, K., Kobayashi, T. and Imai, S. "*An adaptive algorithm for mel-cepstral analysis of speech*," Proc. ICASSP-92, 1992, pp. 137-40.

[6] Stylianou, Y., Laroche, J., and E. Moulines. *High quality speech modification based on a harmonic + noise model*, Proc. Eurospeech, 1995, pp. 451-454.

[7] Kominek, J. *Pitch synchronous harmonic representation of speech for synthesis*, CMU Technologies Institute, Tech Report CMU-LTI-06-xyz, (in preparation).

[8] ISO226:2003. Acoustics. Normal equal-loudness-level contours, ISBN 0580425487.

[9] Black, A. and Lenzo, K. *Optimal Data Selection for Unit Selection Synthesis*, ISCA 4th Speech Synthesis Workshop, 2001, Scotland, pp 63-67.

[1]Wavefiles of the experiments described here are available with this publication on the workshop web site.