Department of Computer Science
University of Toronto
http://learning.cs.toronto.edu

6 King's College Rd, Toronto
M5S 3G4, Canada
fax: +1 416 978 1455

Funded in part by NSERC

January 20, 2004

UTML TR 2004–001

A Segmental HMM for Speech Waveforms

Kannan Achan, Sam Roweis, Brendan Frey
Machine Learning Group, University of Toronto

Abstract

We present a purely time domain approach to speech processing which identifies waveform samples at the boundaries between glottal pulse periods (in voiced speech) or at the boundaries between unvoiced segments. An efficient algorithm for inferring these boundaries is derived from a simple probabilistic generative model of speech and state of the art results are presented on pitch tracking, voiced/unvoiced detection and timescale modification.

A Segmental HMM for Speech Waveforms

Kannan Achan, Sam Roweis, Brendan Frey
Machine Learning Group, University of Toronto

1 Speech Segments in the Time Domain

Processing of speech signals directly in the time domain is commonly regarded to be difficult and unstable, due to fact that perceptually very similar utterances exhibit very large variability in their raw waveforms. As a result, by far the most common preprocessing step for most speech systems is to convert the raw waveform into a time-frequency representation, using a variety of spectral analysis and filterbank techniques. In this paper we pursue a purely time domain approach to speech processing in which we identify the samples at the boundaries between glottal pulse periods (in voiced speech) or at the boundaries between unvoiced segments of similar spectral shape ("colour").

Having identified these segment boundaries, we can perform a variety of important low level speech analysis operations directly and conveniently. For example, we make a voiced/unvoiced decision on each segment by examining the periodicity of the waveform in that segment only. In voiced segments we can estimate the pitch as the reciprocal of the segment length. Timescale modification without pitch or format distortion can be achieved by stochastically eliminating or replicating segments in the time domain directly. More sophisticated operations, such as pitch modification, gender and voice conversion, and companding (volume equalization) are also naturally performed by operating on waveform segments one by one without the need for a cepstral or other such representation.

The computational challenge with this approach is in efficiently and robustly identifying the segment boundaries, across silence, unvoiced and voiced segments. In this paper we introduce a segmental Hidden Markov Model, defined on variable length sections of the time domain waveform, and show that performing inference in this model allows us to identify segment boundaries and achieve excellent results on the speech processing tasks described above.

2 A probabilistic generative model of time-domain speech segments

The goal of our algorithm is to break the time domain speech signal $s_1, \ldots, s_N$ into a set of segments, each of which corresponds to a glottal pulse period or a segment of unvoiced colored noise. Let b_k denote the time index of the beginning of the kth segment and $\mathbf{s}_k = (s_{b_k}, \ldots, s_{b_{k+1}-1})$ denote the waveform in the kth segment, where $k = 1, \ldots, K$ indexes segments. Our algorithm searches for the segment boundaries, $b_1, b_2, \ldots, b_{K+1}$, so that *each segment can be accurately modeled as a time-warped, amplitude-scaled and amplitude-shifted version of the previous segment*. We denote the transformation used to map segment $\mathbf{s}_{k-1}$ into segment $\mathbf{s}_k$ by $\mathbf{T}_k$. (A similar idea is used in [1] to cluster patterns in a way that is invariant to a set of transformations.) Given the segment boundaries $b_1, \ldots, b_{K+1}$ and the transformations $\mathbf{T}_1, \ldots, \mathbf{T_k}$ we

† Thanks to John Hopfield.

assume the probability of each segment depends only on the previous segment and the transformation for that segment: in other words we assume the segments are generated by first order Markov chain:

$$P(\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_K | b_1, \ldots, b_{K+1}, \mathbf{T}_1, \ldots, \mathbf{T}_K) = \prod_{k=1}^{K} P(\mathbf{s}_k | \mathbf{s}_{k-1}, b_{k-1}, b_k, b_{k+1}, \mathbf{T}_k). \quad (1)$$

Each segment is modeled as a noisy copy of the transformed version of the previous segment. These assumptions simplify the inference and estimation algorithm described below. Of course, the segment boundaries are unknown and must be inferred from the speech wave: this inference is the main computation performed by our algorithm.

For concreteness, we assume that each successive segment $\mathbf{s}_k$ is equal to a transformed version of the previous segment, plus isotropic, zero-mean normal noise with variance σ_k^2. Denoting the transformed version of segment $k-1$ by $\mathbf{T}_k \mathbf{s}_{k-1}$, the conditional probability density of $\mathbf{s}_k$ is:

$$P(\mathbf{s}_k | \mathbf{s}_{k-1}, b_{k-1}, b_k, b_{k+1}, \mathbf{T}_k) = \frac{1}{(2\pi\sigma_k^2)^{(b_{k+1}-b_k+1)/2}} \cdot \exp\Big(-\frac{1}{2\sigma_k^2} (\mathbf{s}_k - \mathbf{T}_k \mathbf{s}_{k-1})^{\mathrm{T}} (\mathbf{s}_k - \mathbf{T}_k \mathbf{s}_{k-1}) \Big). \quad (2)$$

The noise levels $\sigma_2^2, .., \sigma_K^2$ are estimated automatically by the inference procedure along with the segment boundaries

(As the boundary condition of the Markov chain, we assume that the segment before the first is a vector of all zeros ($\mathbf{s}_0 = \mathbf{0}$) and hence the probability density of the initial segment is given by $(2\pi\sigma_1^2)^{-b_2/2} \exp(-\mathbf{s}_1^{\mathrm{T}} \mathbf{s}_1 / 2\sigma_1^2)$. We also set σ_1^2 equal to the variance of all time-domain samples, since *a priori* we do not know what the content of the initial segment should be.)

We assume that the boundaries and transformations are independent, and that the prior distribution over transformations is uniform on some bounded set. In our experiments, we parameterize the transformation by $\mathbf{T}_k(\alpha_k, \beta_k, \gamma_k)$, where α_k, β_k and γ_k are time-warp, amplitude-scaling and amplitude-shift. We use a prior that is uniform over a 3-dimensional hypercube that includes all reasonable values for these parameters.

Generally the joint prior probability mass function on segment boundaries $P(b_1, \ldots, b_{K+1})$ can be quite complex. Since the computational complexity of the inference algorithm will depend on the number of allowed configurations of segment boundaries, we use a prior that is non-zero only on an appropriate subset of configurations. In particular, we exploit a very simple heuristic (first suggested by John Hopfield in 1998) by *restricting segments to begin and end only on zero crossings of the signal* (or possibly only on upward or downward going zero crossings). This restriction also allows arbitrary segments to be relocated beside each other and still preserve waveform continuity, which will be important in our later applications. To further restrict the range of inferred segment lengths, we require that $\Delta_{\min} \leq b_k - b_{k-1}$, where $\Delta_{\min}$ is the minimum segment length, satisfying $\Delta_{\min} > 0$. This minimum length is selected by hand and is determined by the expected range of pitch periods and the sampling frequency, in a straightforward fashion. We assume the probability $P(b_1, \ldots, b_{K-1})$ is otherwise uniform, subject to the above constraints.

In fact, we also allow the case $b_k = b_{k-1}$. This enables the inference algorithm to coalesce neighboring segments, effectively "removing" a segment boundary if it needs to. We assume that $b_1 = 1$ (the first segment begins on the first signal sample) and that $b_{K+1} = N + 1$ (the last segment ends on the last signal sample). We initialize the inference procedure with enough segment boundaries so that only removals (as described above) but not insertions are

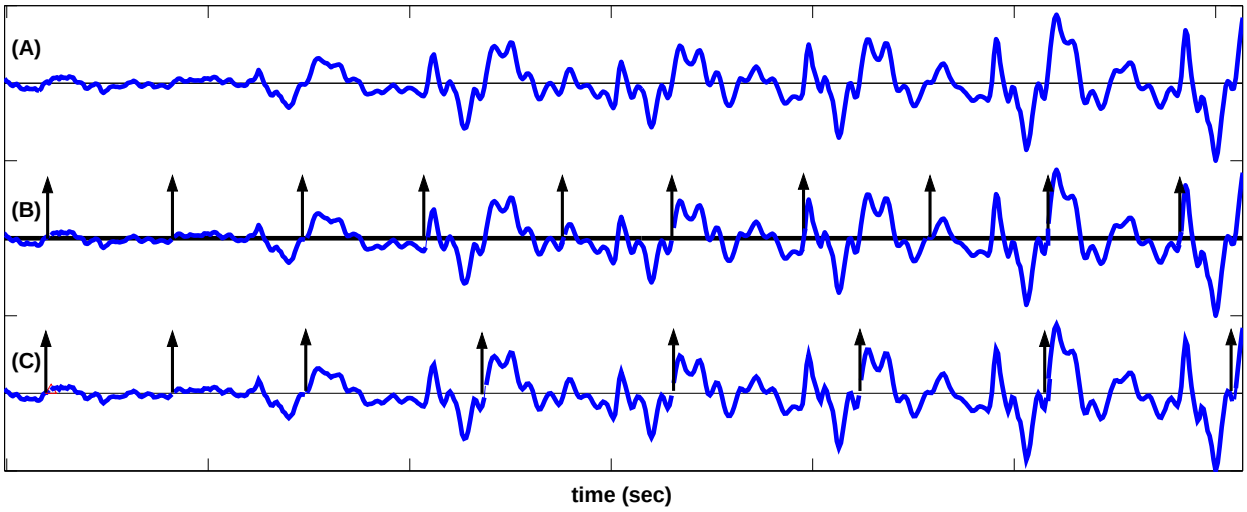

Figure 1: *(A)*Input signal; notice the transition from unvoiced to voiced region. *(B)*Inferred segments after the first iteration - lack of a reliable template at the beginning of the voiced segment results in bad estimates. The upward arrows are used to mark the inferred segment boundaries *(C)* Inferred segments after 3 iterations - segment boundaries have been inferred correctly.

necessary. The joint distribution over segments, segment boundaries and transformations can now be written as:

$$P(\mathbf{s}_1, \ldots, \mathbf{s}_K, b_1, \ldots, b_{K+1}, \mathbf{T}_1, \ldots, \mathbf{T}_K) \propto P(b_1, \ldots, b_{K+1}) \prod_{k=1}^{K} P(\mathbf{T}_k) P(\mathbf{s}_k | \mathbf{s}_{k-1}, b_{k-1}, b_k, b_{k+1}, \mathbf{T}_k), \tag{3}$$

where $P(b_1, \ldots, b_{K+1})$ enforces the constraints on the boundaries; constraints on the allowable limits of the time domain scale, amplitude-domain scale and amplitude-domain shift are enforced by $\left(\prod_{k=1}^{K} P(\mathbf{T}_k)\right)$ although these constraints rarely affect the optimization.

3 Using dynamic programming to efficiently infer segment boundaries and transformations

Given a time-domain signal, the computational task now at hand is to determine the segment boundaries and transformations. Of course, the number of valid configurations of the boundary variables is exponential in the length of the waveform, so exact inference is intractable. We present a greedy, iterative technique for finding the maximum a-posteriori (MAP) estimates of these variables. At iteration i, the technique computes the current MAP estimates, $b_1^{(i)}, \ldots, b_{K+1}^{(i)}$ and $\mathbf{T}_1^{(i)}, \ldots, \mathbf{T}_{K+1}^{(i)}$ in a fashion that monotonically improves the model likelihood of the observed waveform.

To simplify the algorithm, we note that according to (3), given the boundary variables, the MAP estimates of the transformations can be computed locally:

$$\begin{aligned} &\arg\max_{\mathbf{T}_k} P(\mathbf{s}_1, \ldots, \mathbf{s}_K, b_1, \ldots, b_{K+1}, \mathbf{T}_1, \ldots, \mathbf{T}_K) \\ &= \arg\max_{\mathbf{T}_k} P(\mathbf{T}_k) P(\mathbf{s}_k | \mathbf{s}_{k-1}, b_{k-1}, b_k, b_{k+1}, \mathbf{T}_k). \end{aligned} \tag{4}$$

In particular, the time-warping is unique and is given by $\alpha_k = (b_{k+1} - b_k)/(b_k - b_{k-1})$. The warped version of $\mathbf{s}_{k-1}$ is denoted by $\hat{\mathbf{s}}_{k-1}$ and can be obtained using standard signal processing

techniques for time-domain interpolation or decimation. Note that whereas $\mathbf{s}_{k-1}$ contains $b_k - b_{k-1}$ samples, $\hat{\mathbf{s}}_{k-1}$ contains $b_{k+1} - b_k$ samples. The amplitude-domain scale β_k and shift γ_k are obtained by performing a least-squares regression of $\hat{\mathbf{s}}_{k-1}$ onto $\mathbf{s}_k$, *i.e.* by solving

$$\arg\min_{\beta_k,\gamma_k} \sum_{j=1}^{b_{k+1}-b_k} (\beta_k \hat{\mathbf{s}}_{k-1}(j) + \gamma_k - \mathbf{s}_k(j))^2, \tag{5}$$

where (j) indexes the elements of $\mathbf{s}_k$ and $\hat{\mathbf{s}}_{k-1}$. After optimizing β_k and γ_k, the estimate of the variance σ_k^2 is set to the argument in the above minimization, divided by $b_{k+1} - b_k$. For a given configuration of b_{k-1}, b_k, b_{k+1}, we denote the optimal transformation obtained in the above fashion by $\mathbf{T}_k^*$. This optimization is performed at each step of the search over the boundary variables, described below.

At each iteration of the algorithm, instead of considering all possible values for each boundary variable, we embed a dynamic programming grid in the space of valid configurations of the boundary variables and use the Viterbi algorithm to find the best configuration in the space spanned by this embedded dynamic program. (This is similar to the idea proposed in [2] for doing inference in nonlinear dynamical systems.) At iteration i, we use the estimate of the boundary variables from the previous iteration, $b_1^{(i-1)}, \ldots, b_{K+1}^{(i-1)}$, to generate one *set* of candidate values for each boundary variable. Let $B_k^{(i)}$ be the set of candidate values for boundary variable b_k. We take $B_k^{(i)}$ to be the time indices of the J zero-crossings that are *closest* to $b_k^{(i-1)}$, along with the value b_k itself and the value $b_{k-1}^{(i-1)}$. We include the latter two values so that one path through the embedded dynamic programming grid always corresponds to the existing path (thus the search can never worsen the likelihood) and another corresponds to coalescing segment k with segment $k+1$. Using these candidate values, the Viterbi algorithm is used to find the most probable path:

$$\max_{b_2 \in B_2^{(i)}, \ldots, b_K \in B_K^{(i)}} P(b_1, \ldots, b_{K+1}) \prod_{k=1}^{K} P(\mathbf{s}_k | \mathbf{s}_{k-1}, b_{k-1}, b_k, b_{k+1}, \mathbf{T}_k[b_{k-1}, b_k, b_{k+1}]). \tag{6}$$

In order to make the optimization Markovian, we must actually consider *adjacent pairs* of boundary points (b_{k-1}, b_k) as the states in the dynamic programming. Crucially, we enforce the constraint that for any state and its successor, the boundary point they share must take on the same value. We also enforce the constraint that boundary points cannot appear in time-reversed order in a state. Since any two likelihood functions overlap by at most two boundary variables, the memory required for this dynamic programming is equal to the square of the number of configurations that each boundary variable can take on. Since the embedded DP considers only J such values, the memory requirement is of order is J^2. In Fig.1, we have shown the inferred segments obtained using our algorithm after the first and third iteration.

4 EXPERIMENTS

We have applied our segmental inference procedure to clean, wideband recordings of single-talker speech, from both males and females taken from the the Keele pitch reference dataset [3] and from the Wall Street Journal (WSJ) corpus.

During the embedded dynamic programming search, we used a neighbourhood size in the range 2-4 (giving J=5-9 candidates per state) and the threshold $\Delta_{\min}$ on the minimum pitch period was set at to be $2ms$ (corresponding to a maximum pitch of 500Hz). The optimization was initialized by placing the beginning of the first segment at the beginning of the utterance,

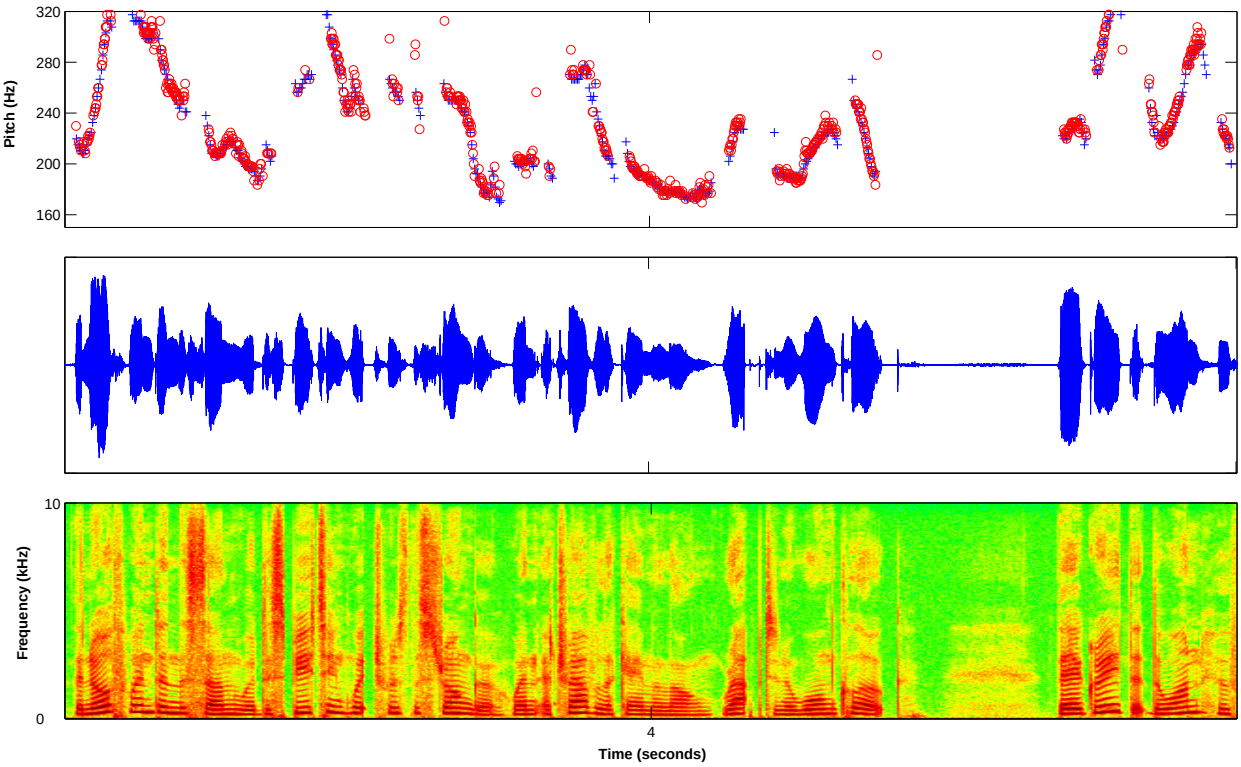

Figure 2: *(top)*Pitch estimates using segmental HMM for a female speaker in the Keele dataset. Notice that the inferred pitch(red circle) consistently agrees with the reference provided (blue plus mark). Further, our approach clearly discriminates between voiced/unvoiced regions (samples without reference estimates are unvoiced). *(center)* input time domain signal *(bottom)* spectrogram of input

and the end of the first segment at the closest zero crossing to $4.5ms$. From this initial segment, a forward sweep of greedy initialization is performed as follows: we take the current segment and search a set of candidate future zero crossings (corresponding to a reasonable allowable range of time-warpings) to find the best endpoint of the next segment. For each candidate, we perform a least-squares fit to the time-warped version of the current segment, and identify the best transformation parameters. The candidate with the lowest fit error is chosen, and we repeat the procedure greedily for the next segment. After initialization, the embedded dynamic programming procedure is run to further improve the estimates of the segment boundaries. We can apply the results of our segment inference algorithm to a wide range of speech processing tasks. By replicating or deleting some or all of the inferred segments, we can easily achieve high quality timescale modification without changing the perceived pitch or formant structure of the utterance. By examining the periodicity of each segment, we can attempt to distinguish voiced from unvoiced portions of the waveform. In voiced regions, we can directly estimate the pitch by taking the reciprocal of the segment length. Below, we present results on timescale modification, voiced/unvoiced discrimination, and pitch tracking. Other applications such as gender and voice conversion, companding and concert hall effects are also possible. We emphasize that all the experiments were performed in *time domain* using the inferred pitch periods. For audio demonstrations and samples, please check http://www.psi.toronto.edu/∼kannan/Segmental

For voicing detection and pitch tracking, we evaluated the estimates obtained using our algorithm using the Keele dataset, since it has ground truth values for these quantities. (In particular, the Keele data has utterances spoken by both male and female speakers and includes a reference estimate for the fundamental frequency at a resolution of 10ms. Each utterance is

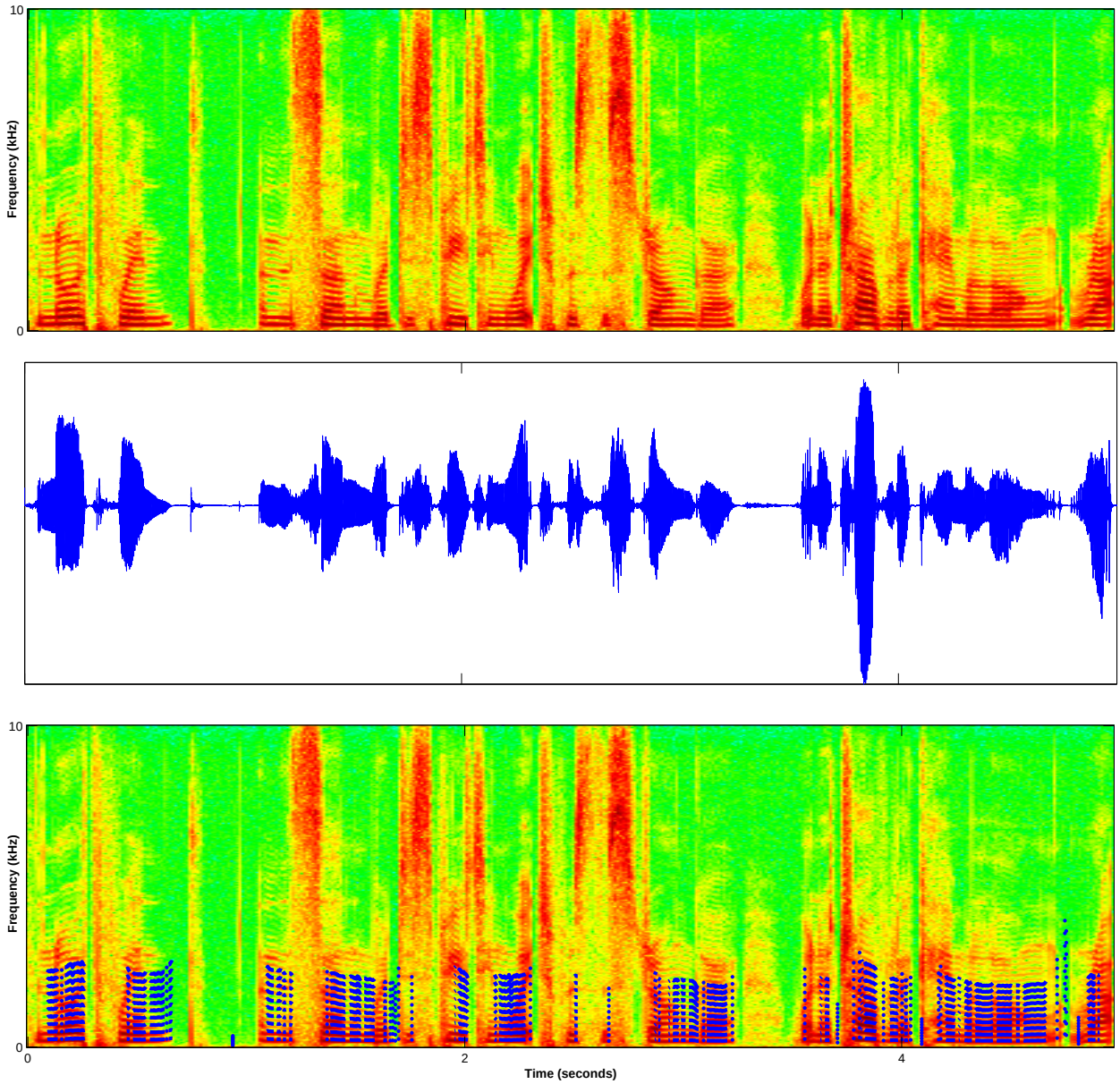

Figure 3: *(Middle and Top)*Time domain signal and the corresponding spectrogram *(Bottom)*The spectrogram of the signal is marked with the pitch estimates obtained using our algorithm (blue marker); for clarity we have marked only the first 10 integer multiples of the fundamental frequency

approximately 30 seconds long and the sampling frequency is 20kHz.)

Once the waveform segments are inferred by the algorithm, we can estimate the periodicity of each segment in a simple way by computing the discrete Fourier transform of the segment waveform and then reconstructing it using a limited number of Fourier coefficients.

Since unvoiced regions tend to be much less periodic, they will have a substantially larger reconstruction error than voiced regions and by selecting an appropriate threshold, we can discriminate between voiced and unvoiced segments. Our method was able to correctly identify 87.2 % of the voiced segments averaged over all the 10 utterances of males and females in the Keele dataset. In Fig.2, the true unvoiced regions are the segments without any reference pitch shown, and the unvoiced regions detected by our algorithm are those without estimated pitches.

Pitch tracking is trivially achieved by taking the reciprocal of the segment lengths in the voiced regions. Results for a single utterance in the Keele dataset spoken by a female speaker

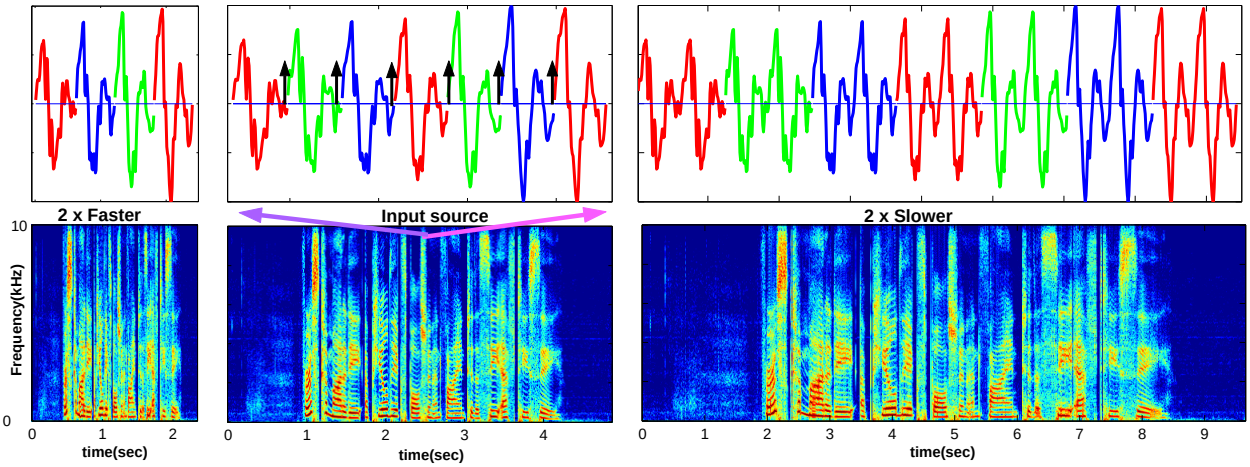

Figure 4: The spectrogram of time scale modified faster and slower versions of a signal are shown. The actual time domain operation is shown on top for a particular time instant in the spectrogram.

is shown in Fig.2. Pitch estimates obtained using our approach are very consistent with the reference estimates; similar performance was obtained on other utterances in the dataset as well. Averaged over 10 utterances the median absolute pitch error was 9Hz.

It is well known that excitation for voiced speech manifests as sharp bursts at integer multiples of fundamental frequency. In Fig.3, we have shown a few integer multiples of the fundamental frequency of a signal on its spectrogram using pitch estimates obtained from the application of our algorithm. For timescale modification experiments, we have used utterances from the WSJ corpus. Once the segments are identified by our algorithm, we can play the signal twice as fast by deleting every other segment and concatenating the remaining ones; similarly by replicating each segment we can achieve the effect of playing the at half the speed (two times slower); this is further illustrated in Fig.4. This approach is substantially different from methods such as [4] that manipulate spectrograms. By doing all of our operations directly in the time domain we never need to worry about inconsistent phase estimates.

5 CONCLUSION

We have presented a simple segmental Hidden Markov Model for generating a speech waveform and derived an efficient algorithm for approximate inference in the model. Applied to an observed signal, this inference algorithm operates entirely in the time domain and is capable of identifying the boundaries of glottal pulse periods in voiced speech and of unvoiced segments. Using these inferred boundaries we are able to easily and accurately detect voicing, track pitch and modify the timescales. We are investigating other possible applications of the same basic model, including voice conversion, volume equalization and reverberant filtering.

References

[1] B. J. Frey and N. Jojic. Transformation-invariant clustering using the EM algorithm. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(1), 2003.

[2] R. Neal, M. Beal, and S. T. Roweis. Inferring state sequences for non-linear systems with embedded hidden markov models,. In *NIPS 16 (to appear)*. 2003.

[3] F Plante, W.A. Ainsworth, and G.F Meyer. A pitch extraction reference database. In *Eurospeech*, 1995.

[4] Roucos. S. and A. M. Wilgus. High quality time-scale modification for speech. In *ICASSP*. IEEE, 1985.