AUTOMATICALLY DERIVED UNITS FOR SEGMENT VOCODERS

V. Ramasubramanian and T. V. Sreenivas

Department of Electrical Communication Engineering
Indian Institute of Science, Bangalore 560 012, India
email: vram@ece.iisc.ernet.in

ABSTRACT

Segment vocoders play a special role in very low bitrate speech coding to achieve intelligible speech at bitrates $\sim$ 300 bits/sec. In this paper, we explore the definition and use of automatically derived units for segment quantization in segment vocoders. We consider three automatic segmentation techniques, namely, the spectral transition measures (STM), maximum-likelihood (ML) segmentation (unconstrained) and duration-constrained ML segmentation, towards defining diphone-like and phone-like units. We show that the ML segmentations realize phone-like units which are significantly better than those obtained by STM in terms of match accuracy with TIMIT phone segmentation as well as actual vocoder performance measured in terms of segmental SNR. Moreover, the phone-like units of ML segmentations also outperform the diphone-like units obtained using STM in early vocoders. We also show that the segment vocoder can operate at very high intelligibility when used in a single-speaker mode.

1. INTRODUCTION

Segment vocoders occupy a special place in very low bit-rate speech coding for their ability to achieve intelligible speech at bit-rates of 800 bits/s (and less) down to 300 bits/s [1], [2], [3], [4], [5], [6].

Fig. 1 shows a segment vocoder with four basic components:

1. Segmentation of input speech (a sequence of LP parameter vectors) into a sequence of variable length segments.
2. Segment quantization of each of these segments using a segment codebook and transmission of the best-match code-segment index and input segment duration.
3. Synthesis of speech by LP synthesis using the code-segment time-normalized to match input segment duration.
4. The residual obtained by LP analysis is parameterized and quantized; the residual decoder reconstructs the residual to be used for synthesis in step (3).

The various segment vocoders proposed till date differ primarily in terms of three aspects: i) Definition of segmental units used for segment quantization, ii) How segmentation (step-1) and segment quantization (step-2) are realized and, iii) Type of segment codebook.

The definition of an unit is implicitly tied to the manner in which segmentation and segment quantization are performed. Much of segment vocoder research has focused on the 'phonetic-vocoder' [4] which has evolved into the paradigm of recognition-synthesis coding at very low bit rates [5], [6]. In these systems, segmentation and segment quantization are performed in a single step of 'phone decoding', as in continuous speech recognition, using an inventory of phone models, which are typically HMMs. This implicitly defines the unit of segmentation and segment quantization as 'phonetic units' as the phone inventory is obtained from manually labeled phonetic database (training data).

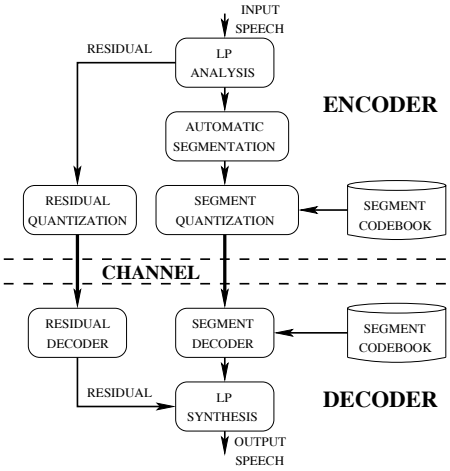

Fig. 1. *Segment vocoder*

In contrast, the earliest vocoder [1], [2] uses 'automatically derived units'. Here, the automatic segmentation and segment quantization are performed as two separate steps. Automatic segmentation is done using a simplistic spectral transition measure (STM) which generates diphone-like units; these are used in both segment codebook design and segment quantization. Other attempts to depart from phonetic units and use automatically derived units are [3] and [5]. In [3] they design an optimized segment codebook by an iterative joint-segmentation and clustering procedure and use it for segment quantization as in connected phone decoding in phonetic vocoders. In [5], they use temporal decomposition and vector quantization to derive 'multi-gram units', which are eventually trained to be phonetic in nature. Both these systems are complex in both segment design and segment quantization operations.

Segment vocoders also differ on the nature of the codebook used: i) non-parametric templates and, ii) parametric models such as HMM. Either of these will determine the quality of synthesized speech and the corresponding bit-rate. Considering the non-parametric approach (which is capable of higher quality synthesis), this paper is concerned with determining which type of automatically derived units can provide best performance. Thus, we consider two types of automatically derived units, viz., diphone-like and phone-like and three types of segmentation techniques, viz., spectral transition measure (STM), maximum - likelihood (ML) segmentation (unconstrained) and duration constrained ML seg-

mentation and evaluate their performance objectively in a vocoder using segmental SNR.

2. AUTOMATIC SEGMENTATION

2.1. Spectral transition measure (STM)

The 'spectral transition measure' (STM) is based on the principle of measuring the spectral derivative at every frame instant. STM was adopted in early segment vocoders for diphone-like segmentation [1], [2]. We consider two types of STM as used in [1], namely, the d_1 and d_3 measures. These are defined as follows: Let $\mathbf{x}_n$ be the LP parameter vector at the n^{th} frame. The STM at frame n, $d_i(n)$, is given by $d_i(n) = \|\mathbf{x}_n - \mathbf{x}_{n-i}\|^2,\ i = 1, 3$.

$d_1(n)$ as a function of n exhibits peaks at fast spectral transitions (such as from one phone to another) and valleys at steady-state regions (such as within a vocalic segment). $d_3(n)$ gives a smoother measure of the spectral derivative. Thus, peak-picking of $d_1(n)$ or $d_3(n)$ locates transitions or phone boundaries and results in a phone-like segmentation. Picking the minima (valleys) of these functions locates a frame within steady-state regions that has maximum local stationarity and corresponds to a diphone boundary. Successive peaks therefore mark phone-like (PL) segments and successive valleys mark diphone-like (DPL) segments.

We use the extrema picking algorithm (EPA) used in [7] for peak- and valley- picking on $d_1(n)$ and $d_3(n)$ functions. This algorithm employs a threshold (δ) to detect the extrema (peaks and valleys) alternatingly in a left-to-right scanning. The algorithm can be stated as follows: `Keep searching` *for a peak (valley) by repeated updating of current maximum p (current minimum v) every time a local maximum (minimum) is detected* `until` *a function value smaller than* $(1-\delta)p$ *(larger than* $(1+\delta)v$*) is encountered.* `After this`*, start searching for a valley (peak).*

While small values of δ (close to 0) result in over-segmentation, large values of δ (close to 1) result in under-segmentation and δ needs to be optimized for a desired segment rate (See Sec. 2.3.1).

2.2. Maximum-likelihood (ML) segmentation

2.2.1. ML segmentation – unconstrained (ML(UC))

Let a speech utterance be given by $\mathbf{X}_1^T = (\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T)$, which is a LP parameter vector sequence of T speech frames, where, $\mathbf{x}_n$ is a p-dimensional parameter vector at frame 'n'. The segmentation problem is to find 'm' consecutive segments in the observation sequence $\mathbf{X}_1^T$. Let the segment boundaries be denoted by the sequence of integers $\mathcal{B} = (b_o, b_1, \ldots, b_m)$. The i^{th} segment starts at frame $b_{i-1} + 1$ and ends at frame b_i; $b_0 = 0$, and $b_m = T$.

The maximum likelihood (ML) segmentation is based on using the piecewise stationarity of speech as the acoustic criterion for determining segments. The criteria is to obtain segments which exhibit maximum acoustic homogeneity within their boundaries. The acoustic inhomogeneity of a segment is measured in terms of an 'intra-segmental distortion', given by the sum of distances from the frames that span the segment, to the centroid of the frames comprising the segment. The optimal segmentation $(b_o^*, b_1^*, \ldots, b_m^*)$ is obtained so as to minimize the sum of intra-segment distortion over all possible segment boundaries, i.e., minimize

$$D(m, T) = \sum_{i=1}^{m} \sum_{n=b_{i-1}+1}^{b_i} d(\mathbf{x}_n, \mu_i) \qquad (1)$$

where, $D(m, T)$ is the total distortion of a m - segment segmentation of $\mathbf{X}_1^T = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$; μ_i is the centroid of the i^{th} segment consisting of the spectral sequence $\mathbf{X}_{b_{i-1}+1}^{b_i} = \{\mathbf{x}_{b_{i-1}+1}, \ldots, \mathbf{x}_{b_i}\}$ for a specific distance measure $d(\cdot, \cdot)$. For the Euclidean distance 'd', μ_i is the average of the frames in the segment $\mathbf{X}_{b_{i-1}+1}^{b_i}$.

The optimal segment boundaries are solved efficiently using a dynamic programming (DP) procedure [8], [9] using the recursion

$$D(i, b_i) = \min_{b_{i-1}} [D(i-1, b_{i-1}) + \Delta(b_{i-1}+1, b_i)] \qquad (2)$$

for all possible b_{i-1}; $D(i, b_i)$ is the minimum accumulated distortion upto the i^{th} segment (which ends in frame b_i), i.e., $D(i, b_i)$ is the minimum distortion of a segmentation of $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_{b_i}\}$ into i segments; $\Delta(b_{i-1}+1, b_i)$ is the intra-segment distortion of the i^{th} segment $\mathbf{X}_{b_{i-1}+1}^{b_i}$. The segmentation problem is solved by invoking (2) for $D(m, T)$; this is computed efficiently by a trellis realization. The optimal segment boundaries $(b_o^*, b_1^*, \ldots, b_m^*)$ are retrieved by backtracking on the trellis along the optimal alignment path corresponding to $\min\{D(m, T)\}$.

2.2.2. ML segmentation – duration constrained (ML(DC))

By definition, ML segmentation produces a segmentation where each segment is maximally homogenous; when the segment rate equals the phone-rate of natural speech, the resulting segments will be quasi-stationary and would correspond to the steady-state regions of phonetic units. However, even for a correct segment rate, ML segmentation can result in segment lengths which are unnaturally short (1 frame long) or long (upto even 70 frames). Such segments will be distorted significantly during segment quantization and result in poor vocoder performance.

In the distribution of phone durations in TIMIT database, nearly 95% of the labeled phonetic segments are in the range of 1-20 frames. In order to limit the segment lengths of ML segmentation to such a meaningful range (of actual phones durations), we modify the ML segmentation to have 'duration constraints'. Here, the optimal segments are forced to be within a duration range of $[\alpha, \beta]$, where α and β are the minimum and maximum lengths possible (in frames). Segments of lengths $< \alpha$ and $> \beta$ are not generated at all. This is achieved by restricting the candidate boundaries in the search for optimal segment boundaries in (2) as follows:

$$D(i, b_i) = \min_{b_i - \beta \le b_{i-1} \le b_i + \alpha} [D(i-1, b_{i-1}) + \Delta(b_{i-1}+1, b_i)]$$

This also has the advantage of reducing the computational complexity of ML segmentation from $O(T^2)$ to $O(lT)$ where $l = \beta - \alpha + 1$ with typical values of $[\alpha, \beta] = [2, 20]$.

2.3. Performance evaluation of STM and ML segmentations

Here, we evaluate the segmentation performance of the two segmentation techniques: i) STM with d_1 and d_3 functions (STM(d_1) and STM(d_3)) and, ii) ML segmentation – unconstrained (ML(UC)) and duration constrained (ML(DC)) for phone-like segmentation using the manually labeled phonetic database TIMIT. We use 4 measures to quantify the degree of match between the automatic segmentation and TIMIT segmentation. These are: % Match (%M), % Segment match (%S), % Insertions (%I) and % Deletions (%D).

%M gives the percentage of segments boundaries obtained by the automatic segmentation, within a specified interval of 2 frames (or $\Delta = 20$ ms, corresponding to a 10ms analysis frame) of the manual segments. %S gives the percentage of two successive segment boundaries in the automatic segmentation each to be within the interval Δ ms of successive manual segments. This essentially

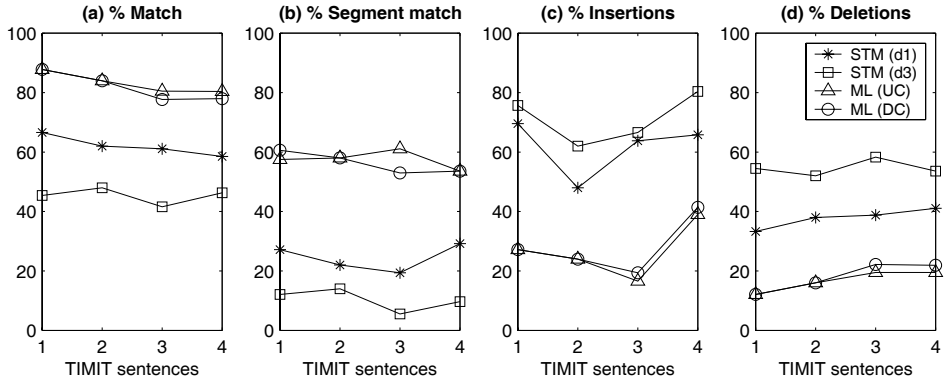

Fig. 2. *Segmentation performance of segmentation techniques STM(d_1), STM(d_3), ML(UC) and ML(DC) using segmentation match measures (a) % Match (%M), (b) % Segment match (%S), (c) % Insertions (%I) and (d) % Deletions (%D) for 4 TIMIT sentences.*

measures the percentage of 'segment match', i.e., a segment in automatic segmentation matching a segment in manual segmentation. %I gives the percentage of segments obtained by the automatic segmentation without a corresponding manual segment within the interval of Δ ms. %D gives the percentage of manually obtained segments without any corresponding automatically segmented boundary within the interval of Δ ms. For a good segmentation match with the manual segmentation, it is desired to have as high a %M and %S and as low a %I and %D as possible.

2.3.1. STM segmentation

In STM, the threshold δ used in the extrema-picking-algorithm plays a crucial role in determining the quality of segmentation and hence needs to be optimized to yield good segmentation match as defined above. For this purpose, we used the segment rate (number of segment/sec R) as the primary measure to be matched. For instance, TIMIT has a phone-rate of $R = 12.5$ phones/second, as measured over 300 sentences. In STM, we set δ to a value that results in this segment rate. The optimal δ corresponding to a segment rate of 12.5 seg/sec also results (automatically and interestingly) in the highest %M for the lowest (%I + %D) values.

Fig. 2 shows the measures (%M, %S, %I, %D) for 4 TIMIT sentences (`sx213, si916, si1086, sa1`) from 4 different speakers (2 male and 2 female) in the database; these TIMIT sentences are marked as (1, 2, 3, 4) on the x-axis. From this figure, it is clear that STM(d_1) is significantly better than STM(d_3) for realizing phone-like segments, with d_1 yielding as much as 20% higher %M and %S than d_3. The corresponding %I and %D are 10-20% lower for d_1 than d_3.

This follows from the fact that d_1 is well suited for detecting fast spectral transitions such as phone boundaries and hence in better phone-like segmentation than d_3. Conversely, the smoother d_3 is more suited for a diphone-like segmentation than d_1, where it is necessary to detect steady-states by valley-picking. We therefore focus on STM(d_1) for phone-like (PL) segmentation and STM(d_3) for diphone-like (DPL) segmentation in further experiments with STM in the performance of the overall vocoder in Sec. 3.

2.3.2. ML segmentation

The ML segmentation performed using the DP recursion (2) can segment the input speech into a pre-specified number of segments. We specify the number of segments required as $m = Rt$, where R is the desired segment rate (such as 12.5 seg/sec) and t is the duration (in secs) of the speech interval being segmented. Such a choice of segment-rate results in high segmentation performance as given by (%M, %S, %I, %D) with respect to TIMIT boundaries.

In the case of ML(DC), in addition to the segment rate, we also need to specify the duration constraints $[\alpha, \beta]$. We studied the performance of ML(DC) in terms of the 4 measures (%M, %S, %I, %D), for various duration constraint ranges $[\alpha, \beta]$ with respect to ML(UC). Based on this, we chose a duration constraint range of [2,16] as a conservative range for ML(DC) so as to retain the performance of ML(UC) while ensuring phone-like segments.

Fig. 2 also shows the performance of unconstrained ML segmentation (ML(UC)) and duration constrained ML segmentation (ML(DC) in terms of the 4 measures (%M, %S, %I, %D). Clearly, ML(UC) outperforms STM segmentations with the highest %M and %S and lowest %I and %D across all the TIMIT sentences. ML(UC) is better than STM(d_1) by as much as 20% in %M and 30% in %S. %I and %D for ML(UC) are lower by as much as 40% and 20% respectively when compared to STM(d_1). Duration constrained ML segmentation (ML(DC)) performs as well as ML(UC) with only marginal differences. Thus ML(DC), with its lower complexity, proves to be an efficient way of realizing phone-like segmentation under the ML formulation, clearly outperforming the more simplistic STM based segmentation. Note that the 60% %S (phone-like segment match) obtained by ML segmentation is comparable to state of the art phone recognition accuracies.

3. EXPERIMENTS AND RESULTS

Segment codebook: The segment codebook has to be representative of the acoustic space of the segments generated by the automatic segmentation. The use of large randomly populated codebooks from such segment corpus was considered in [2], motivated by the theoretical observation that a random quantizer is optimal for Gaussian random vectors of large dimensionality – in this case, variable length segments. It was found from experiments that the random codebook indeed performed perceptually close to a quantizer obtained by clustering procedures [2] thus obviating computationally expensive clustering and modeling [3], [5].

In this paper, we use such a large size random codebook of size 8192 corresponding to a bit-rate of 13 bits/segment. This was obtained from automatic segmentation of about 300 sentences (from nearly 30 speakers) in the TIMIT database; this corresponds to the multi-speaker (MS) segment codebook. A single-speaker (SS) codebook of the same size was also designed (for experiments with single-speaker mode) from about 15 minutes of read speech. The speech was sampled at 16kHz and the LP analysis was done on

10ms frames with no overlap; the LP parameters used are log-area-ratios (LARs) of dimension 16.

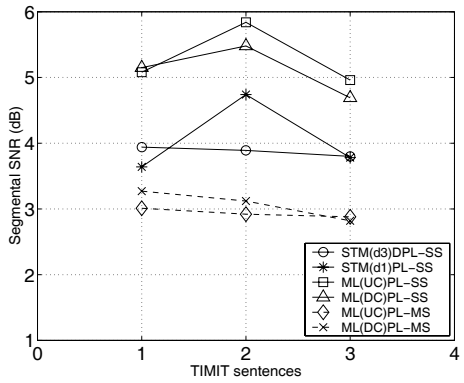

Fig. 3. *Segment vocoder performance in terms of segmental SNR for different segmentation techniques STM(d_3)DPL, STM(d_1)PL, ML(UC)PL and, ML(DC)PL using single-speaker (SS) and multi-speaker (MS) codebooks for 3 TIMIT sentences.*

Segmental SNR: In this paper, our objective is to characterize the performance of segment quantization through different automatic segmentation schemes; therefore, we have isolated the performance of the segment vocoder to have only segment quantization without the residual parameterization and quantization, i.e., the residual is retained as such. The LP synthesis with unparameterized residual retains the sample to sample correspondence between the synthesized speech and input speech. This allows defining an objective measure between the output speech and the input speech in terms of SNR measures such as 'segmental SNR' (Seg-SNR) for evaluating the vocoder performance with respect to its segment quantization performance alone. The segment vocoder has a bit-rate of 300 bits/sec when the residual is quantized.

Results: Fig. 3 shows the Seg-SNR obtained on 3 TIMIT sentences (`sx213, si1086, sa2`) for the following 4 cases of automatic segmentations: i) STM(d_3)DPL-SS: STM with d_3 function for diphone-like segmentation, ii) STM(d_1)PL-SS: STM with d_1 function for phone-like segmentation, iii) ML(UC)PL-SS: Unconstrained ML segmentation for phone-like segmentation and, iv) ML(DC)PL-SS: Duration constrained ML segmentation for phone-like segmentation. These are for vocoder operation using a single-speaker (SS) segment codebook and are hence labeled with SS as a suffix; these 4 cases are marked as solid lines in the figure. Fig. 3 also shows the Seg-SNR for the 3 TIMIT sentences when coded using a multi-speaker (MS) segment codebook for the two cases of ML(UC)PL-MS and ML(DC)PL-MS. These are labeled with MS as suffix and marked as dashed lines in the figure. The single-speaker codebook is obtained from the same speaker as the speaker of the 3 test TIMIT sentences; this speaker is outside the 30 speakers (TIMIT) used for building the multi-speaker codebook.

The following can be observed from the figure:

STM(d_1)PL-SS performs as well as or even better than STM (d_3)DPL-SS, indicating that phone-like units are possibly better for segment quantization. This is brought out more definitely with ML segmentations. Both ML(UC)PL-SS and ML(DC)PL-SS have significantly higher Seg-SNR than STM based segmentations (by upto 1 dB with respect to STM(d_1)PL-SS and upto 2 dB with respect to STM(d_3)DPL-SS). Thus, ML segmentations clearly outperform STM and ML proves to be an effective segmentation technique for realizing phone-like units which are significantly better than diphone-like units for segment quantization. The above Seg-SNR differences also translated very well in terms of perceptual quality of the synthesized speech. The ML segmentation resulted in perceptually superior speech quality, with very high intelligibility and with the speaker identity intact when compared to diphone-like segmentation using STM.

Considering the performance on multi-speaker codebook, both ML(UC)PL-MS and ML(DC)PL-MS have a lower Seg-SNR when compared to their single-speaker performance (ML(UC)PL-SS and ML(DC)PL-SS). This clearly indicates that when a test speech is coded using a speaker codebook that matches the input speaker, a performance gain of as much as 3dB can be accrued. This points to the importance of handling speaker-variability for speaker - independent segment vocoder operation by means of speaker adaptation techniques, such as codebook adaptation.

4. CONCLUSIONS

In this paper, we have explored the effectiveness of two types of 'automatically derived units', namely, diphone-like units and phone - like units for segment quantization in a segment vocoder. These units are obtained from three automatic segmentation techniques: spectral transition measure (STM), maximum - likelihood (ML) segmentation (unconstrained) and duration constrained ML segmentation. We have shown that STM based diphone-like units perform poorly in comparison to phone-like units in vocoder performance. More importantly, we have shown that ML segmentation based phone-like units outperforms STM based phone-like units in terms of both segmentation match measures (with TIMIT segmentation) and actual vocoder performance in terms of segmental SNR. The proposed segment vocoder operates at high intelligibility retaining speaker-identity when used in single-speaker mode.

5. REFERENCES

[1] S. Roucos, R. M. Schwartz, and J. Makhoul. Vector quantization for very-low-rate coding of speech. In *Proc. IEEE Globcom' 82*, pages 1074–1078, 1982.

[2] S. Roucos, R. M. Schwartz, and J. Makhoul. A segment vocoder at 150 b/s. In *Proc. ICASSP*, pages 61–64, 1983.

[3] Y. Shiraki and M. Honda. LPC speech coding based on variable-length segment quantization. *IEEE Trans. on Acoustics, Speech and Signal Proc.*, 36(9):1437–1444, Sept. 1988.

[4] J. Picone and G. Doddington. A phonetic vocoder. In *Proc. ICASSP*, pages 580–583, 1989.

[5] J. Cernocky, G. Baudoin, and G. Chollet. Segmental vocoder – going beyond the phonetic approach. In *Proc. ICASSP*, volume 2, pages 605–608, 1998.

[6] K. Tokuda et al. A very low bit rate speech coder using HMM-based speech recognition / synthesis techniques. In *Proc. ICASSP*, volume 2, pages 609–612, May 1998.

[7] J. P. Martens and L. Depuydt. Braod phonetic classification and segmentation of continuous speech by means of neural networks and dynamic programming. *Speech Communication*, 10(1):81–90, Feb. 1991.

[8] T. Svendsen and F. K. Soong. On the automatic segmentation of speech signals. In *Proc. ICASSP*, pages 77–80, 1987.

[9] A. K. V. Sai Jayram, V. Ramasubramanian, and T. V. Sreenivas. Robust parameters for automatic segmentation of speech. In *Proc. ICASSP*, pages I–513–I–516, May 2002.