(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 501 075 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
26.01.2005 Bulletin 2005/04

(51) Int Cl.[7]: **G10L 13/06**

(21) Application number: **04077723.7**

(22) Date of filing: **12.11.1999**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE

(30) Priority: **13.11.1998 US 108201 P**

(62) Document number(s) of the earlier application(s) in accordance with Art. 76 EPC:
99972346.3 / 1 138 038

(71) Applicant: **Lernout & Hauspie Speech Products N.V.**
8900 Ieper (BE)

(72) Inventors:
- **Coorman, Geert**
  8500 Kortrijk (BE)
- **De Brock, Mario**
  9600 Ronse (BE)
- **Deprez, Filip**
  8510 Bellegem-Kortrijk (BE)
- **Fackrell, Justin**
  9000 Gent (BE)
- **Leys, Steven**
  8500 Kortrijk (BE)
- **Rutten, Peter**
  9050 Gent (BE)
- **DeMoortel, Jan**
  8510 Rollegem (BE)
- **Schenk, Andre**
  8500 Kortrijk (BE)
- **Van Coile, Bert**
  8200 Brugge (BE)

(74) Representative: **Greene, Simon Kenneth**
Elkington and Fife LLP,
Prospect House,
8 Pembroke Road
Sevenoaks, Kent TN13 1XR (GB)

Remarks:
This application was filed on 05 - 10 - 2004 as a divisional application to the application mentioned under INID code 62.

(54) Speech synthesis using concatenation of speech waveforms

(57) A high quality speech synthesizer in various embodiments concatenates speech waveforms referenced by a large speech database. Speech quality is further improved by speech unit selection and concatenation smoothing.

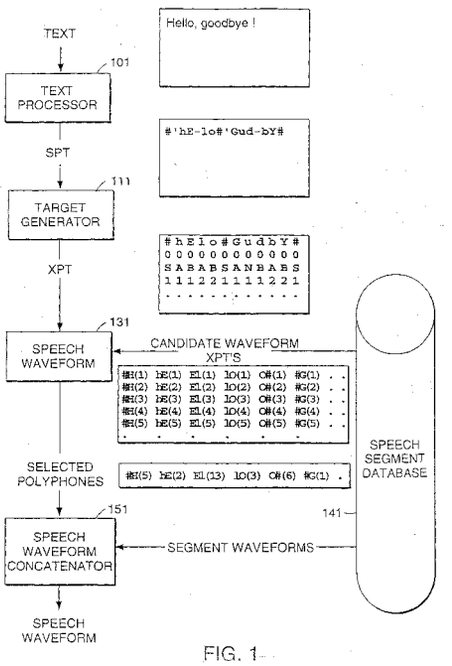


FIG. 1

EP 1 501 075 A2

Description

Technical Field

[0001] The present invention relates to a speech synthesizer based on concatenation of digitally sampled speech units from a large database of such samples and associated phonetic, symbolic, and numeric descriptors.

Background Art

[0002] A concatenation-based speech synthesizer uses pieces of natural speech as building blocks to reconstitute an arbitrary utterance. A database of speech units may hold speech samples taken from an inventory of pre-recorded natural speech data. Using recordings of real speech preserves some of the inherent characteristics of a real person's voice. Given a correct pronunciation, speech units can then be concatenated to form arbitrary words and sentences. An advantage of speech unit concatenation is that it is easy to produce realistic coarticulation effects, if suitable speech units are chosen. It is also appealing in terms of its simplicity, in that all knowledge concerning the synthetic message is inherent to the speech units to be concatenated. Thus, little attention needs to be paid to the modeling of articulatory movements. However speech unit concatenation has previously been limited in usefulness to the relatively restricted task of neutral spoken text with little, if any, variations in inflection.

[0003] A tailored corpus is a well-known approach to the design of a speech unit database in which a speech unit inventory is carefully designed before making the database recordings. The raw speech database then consists of carriers for the needed speech units. This approach is well-suited for a relatively small footprint speech synthesis system. The main goal is phonetic coverage of a target language, including a reasonable amount of coarticulation effects. No prosodic variation is provided by the database, and the system instead uses prosody manipulation techniques to fit the database speech units into a desired utterance.

[0004] For the construction of a tailored corpus, various different speech units have been used (see, for example, Klatt, D.H., "Review of text-to-speech conversion for English," J. Acoust. Soc. Am. 82(3), September 1987). Initially, researchers preferred to use phonemes because only a small number of units was required — approximately forty for American English — keeping storage requirements to a minimum. However, this approach requires a great deal of attention to coarticulation effects at the boundaries between phonemes. Consequently, synthesis using phonemes requires the formulation.of complex coarticulation rules.

[0005] Coarticulation problems can be minimized by choosing an alternative unit. One popular unit is the diphone, which consists of the transition from the center of one phoneme to the center of the following one. This model helps to capture transitional information between phonemes. A complete set of diphones would number approximately 1600, since there are approximately $(40)^2$ possible combinations of phoneme pairs. Diphone speech synthesis thus requires only a moderate amount of storage. One disadvantage of diphones is that they lead to a large number of concatenation points (one per phoneme), so that heavy reliance is placed upon an efficient smoothing algorithm, preferably in combination with a diphone boundary optimization. Traditional diphone synthesizers, such as the ITS-3000 of Lernout & Hauspie Speech And Language Products N.V., use only one candidate speech unit per diphone. Due to the limited prosodic variability, pitch and duration manipulation techniques are needed to synthesize speech messages. In addition, diphones synthesis does not always result in good output speech quality.

[0006] Syllables have the advantage that most coarticulation occurs within syllable boundaries. Thus, concatenation of syllables generally results in good quality speech. One disadvantage is the high number of syllables in a given language, requiring significant storage space. In order to minimize storage requirements while accounting for syllables, demi-syllables were introduced. These half-syllables, are obtained by splitting syllables at their vocalic nucleus. However the syllable or demi-syllable method does not guarantee easy concatenation at unit boundaries because concatenation in a voiced speech unit is always more difficult that concatenation in unvoiced speech units such as fricatives.

[0007] The demi-syllable paradigm claims that coarticulation is minimized at syllable boundaries and only simple concatenation rules are necessary. However this is not always true. The problem of coarticulation can be greatly reduced by using word-sized units, recorded in isolation with a neutral intonation. The words are then concatenated to form sentences. With this technique, it is important that the pitch and stress patterns of each word can be altered in order to give a natural sounding sentence. Word concatenation has been successfully employed in a linear predictive coding system.

[0008] Some researchers have used a mixed inventory of speech units in order to increase speech quality, *e.g.*, using syllables, demi-syllables, diphones and suffixes (see, Hess, W.J., "Speech Synthesis - A Solved Problem, Signal processing VI: Theories and Applications," J. Vandewalle, R. Boite, M. Moonen, A. Oosterlinck (eds.), Elsevier Science Publishers B.V., 1992).

[0009] To speed up the development of speech unit databases for concatenation synthesis, automatic synthesis unit generation systems have been developed (see, Nakajima, S., "Automatic synthesis unit generation for English speech

synthesis based on multi-layered context oriented clustering," Speech Communication 14 pp. 313-324, Elsevier Science Publishers B.V., 1994). Here the speech unit inventory is automatically derived from an analysis of an annotated database of speech - i.e. the system 'learns' a unit set by analyzing the database. One aspect of the implementation of such systems involves the definition of phonetic and prosodic matching functions.

[0010] A new approach to concatenation-based speech synthesis was triggered by the increase in memory and processing power of computing devices. Instead of limiting the speech unit databases to a carefully chosen set of units, it became possible to use large databases of continuous speech, use non-uniform speech units, and perform the unit selection at run-time. This type of synthesis is now generally known as corpus-based concatenative speech synthesis.

[0011] The first speech synthesizer of this kind was presented in Sagisaka, Y., "Speech synthesis by rule using an optimal selection of non-uniform synthesis units," ICASSP-88 New York vol.1 pp. 679-682, IEEE, April 1988. It uses a speech database and a dictionary of candidate unit templates, i.e. an inventory of all phoneme sub-strings that exist in the database. This concatenation-based synthesizer operates as follows.

(1) For an arbitrary input phoneme string, all phoneme sub-strings in a breath group are listed,
(2) All candidate phoneme sub-strings found in the synthesis unit entry dictionary are collected,
(3) Candidate phoneme sub-strings that show a high contextual similarity with the corresponding portion in the input string are retained,
(4) The most preferable synthesis unit sequence is selected mainly by evaluating the continuities (based only on the phoneme string) between unit templates,
(5) The selected synthesis units are extracted from linear predictive coding (LPC) speech samples in the database,
(6) After being lengthened or shortened according to the segmental duration calculated by the prosody control module, they are concatenated together.

[0012] Step (3) is based on an appropriateness measure - taking into account four factors: conservation of consonant-vowel transitions, conservation of vocalic sound succession, long unit preference, overlap between selected units. The system was developed for Japanese, the speech database consisted of 5240 commonly used words.

[0013] A synthesizer that builds further on this principle is described in Hauptmann, A.G., "SpeakEZ: A first experiment in concatenation synthesis from a large corpus," Proc. Eurospeech '93, Berlin, pp.1701-1704,1993. The premise of this system is that if enough speech is recorded and catalogued in a database, then the synthesis consists merely of selecting the appropriate elements of the recorded speech and pasting them together. It uses a database of 115,000 phonemes in a phonetically balanced corpus of over 3200 sentences. The annotation of the database is more refined than was the case in the Sagisaka system: apart from phoneme identity there is an annotation of phoneme class, source utterance, stress markers, phoneme boundary, identity of left and right context phonemes, position of the phoneme within the syllable, position of the phoneme within the word, position of the phoneme within the utterance, pitch peak locations.

[0014] Speech unit selection in the SpeakEZ is performed by searching the database for phonemes that appear in the same context as the target phoneme string. A penalty for the context match is computed as the difference between the immediately adjacent phonemes surrounding the target phoneme with the corresponding phonemes adjacent to the database phoneme candidate. The context match is also influenced by the distance of the phoneme to its left and right syllable boundary, left and right word boundary, and to the left and right utterance boundary.

[0015] Speech unit waveforms in the SpeakEZ are concatenated in the time domain, using pitch synchronous overlap-add (PSOLA) smoothing between adjacent phonemes. Rather than modify existing prosody according to ideal target values, the system uses the exact duration, intonation and articulation of the database phoneme without modifications. The lack of proper prosodic target information is considered to be the most glaring shortcoming of this system.

[0016] Another approach to corpus-based concatenation speech synthesis is described in Black, A.W., Campbell, N., "Optimizing selection of units from speech databases for concatenative synthesis," Proc. Eurospeech '95, Madrid, pp. 581-584, 1995, and in Hunt, A.J., Black, A.W., "Unit selection in a concatenative speech synthesis system using a large speech database," ICASSP-96, pp. 373-376, 1996. The annotation of the speech database is taken a step further to incorporate acoustic features: pitch (F_0), power and spectral parameters are included. The speech database is segmented in phone-sized units. The unit selection algorithm operates as follows:

(1) A unit distortion measure $D_u(u_i, t_i)$ is defined as the distance between a selected unit u_1 and a target speech unit t_i, i.e. the difference between the selected unit feature vector $\{uf_1, uf_2,..., uf_n\}$ and the target speech unit vector $\{tf_1, tf_2,..., tf_n\}$ multiplied by a weights vector W_u $\{w_1, w_2,..., w_n\}$.
(2) A continuity distortion measure $D_c(u_i, u_{i-1})$ is defined as the distance between a selected unit and its immediately adjoining previous selected unit, defined as the difference between a selected units unit's feature vector and its previous one multiplied by a weight vector W_c.
(3) The best unit sequence is defined as the path of units from the database which minimizes:

$$\sum_{i=1}^{n}(D_c(u_i,u_{i-1})*W_c+D_u(u_i,t_i)*W_u)$$

where n is the number of speech units in the target utterance.

[0017] In continuity distortion, three features are used: phonetic context, prosodic context, and acoustic join cost. Phonetic and prosodic context distances are calculated between selected units and the context (database) units of other selected. units. The acoustic join cost is calculated between two successive selected units. The acoustic join cost is based on a quantization of the mel-cepstrum, calculated at the best joining point around the labeled boundary.

[0018] A Viterbi search is used to find the path with the minimum cost as expressed in (3). An exhaustive search is avoided by pruning the candidate lists at several stages in the selection process. Units are concatenated without doing any signal processing (i.e., raw concatenation).

[0019] A clustering technique is presented in Black, A.W., Taylor, P.," Automatically clustering similar units for unit selection in speech synthesis," Proc. Eurospeech '97, Rhodes, pp. 601-604, 1997, that creates a CART (classification and regression tree) for the units in the database. The CART is used to limit the search domain of candidate units, and the unit distortion cost is the distance between the candidate unit and its cluster center.

[0020] As an alternative to the mel-cepstrum, Ding, W., Campbell, N., "Optimising unit selection with voice source and formants in the CHATR speech synthesis system," Proc. Eurospeech '97, Rhodes, pp. 537-540, 1997, presents the use of voice source parameters and formant information as acoustic features for unit selection.

[0021] Each of the references mentioned above is hereby incorporated herein by reference.

[0022] Hunt and Black, in "Unit selection in a concatenative speech synthesis system using a large speech database", IEEE 1996, describes a speech synthesizer with a large speech database. The database uses phonemes.

[0023] Banga and Garcia Mateo, "Shape invariant pitch-synchronous text-to-speech conversion" in ICASSP 90, the International conference on acoustics, speech and signal processing 1990, describes a text to speech system that uses, in an example, diphones.

[0024] According to the invention, there is provided a speech synthesizer comprising:

a. a large speech database referencing speech waveforms and associated symbolic prosodic features, wherein the database is accessed by the symbolic prosodic features and polyphone designators;
b. a speech waveform selector, in communication with the speech database, that selects waveforms referenced by the database using symbolic prosodic features and polyphone designators that correspond to a phonetic transcription input; and
c. a speech waveform concatenator in communication with the speech database that concatenates the waveforms selected by the speech waveform selector to produce a speech signal output.

[0025] In a further related embodiment, the polyphone designators are diphone designators. In a related set of embodiments, the synthesizer also includes (i) a digital storage medium in which the speech waveforms are stored in speech-encoded form; and (ii) a decoder that decodes the encoded speech waveforms when accessed by the waveform selector.

[0026] Also optionally, the synthesizer operates to select among waveform candidates without recourse to specific target duration values or specific target pitch contour values over time.

[0027] In another embodiment, there is provided a speech synthesizer using a context-dependent cost function, and the embodiment includes:

a large speech database;
a target generator for generating a sequence of target feature vectors responsive to a phonetic transcription input;
a waveform selector that selects a sequence of waveforms referenced by the database, each waveform in the sequence corresponding to a first non-null set of target feature vectors, wherein the waveform selector attributes, to at least one waveform candidate, a node cost, wherein the node cost is a function of individual costs associated with each of a plurality of features, and wherein at least one individual cost is determined using a cost function that varies in accordance with linguistic rules; and
a speech waveform concatenator in communication with the speech database that concatenates the waveforms selected by the speech waveform selector to produce a speech signal output.

[0028] In another embodiment, there is provided a speech synthesizer with a context-dependent cost function, and the embodiment includes:

a large speech database;
a target generator for generating a sequence of target feature vectors responsive to a phonetic transcription input;
a waveform selector that selects a sequence of waveforms referenced by the database,

wherein the waveform selector attributes, to at least ordered sequence of two or more waveform candidates, a transition cost, wherein the transition cost is a function of individual costs associated with each of a plurality of features, and wherein at least one individual cost is determined using a cost function that varies nontrivially according to linguistic rules; and

a speech waveform concatenator in communication with the speech database that concatenates the waveforms selected by the speech waveform selector to produce a speech signal output. In a further related embodiment, the cost function has a plurality of steep sides.

[0029] In a further embodiment, there is provided a speech synthesizer, and the embodiment provides:

a large speech database;
a waveform selector that selects a sequence of waveforms referenced by the database,

wherein the waveform selector attributes, to at least one waveform candidate, a cost, wherein the cost is a function of individual costs associated with each of a plurality of features, and wherein at least one individual cost of a symbolic feature is determined using a non-binary numeric function; and

a speech waveform concatenator in communication with the speech database that concatenates the waveforms selected by the speech waveform selector to produce a speech signal output.

In a related embodiment, the symbolic feature is one of the following: (i) prominence, (ii) stress, (iii) syllable position in the phrase, (iv) sentence type, and (v) boundary type. Alternatively or in addition, the non-binary numeric function is determined by recourse to a table. Alternatively, the non-binary numeric function may be determined by recourse to a set of rules.

[0030] In yet another embodiment, there is provided a speech synthesizer, and the embodiment, includes:

a large speech database;
a target generator for generating a sequence of target feature vectors responsive to a phonetic transcription input;
a waveform selector that selects a sequence of waveforms referenced by the database, each waveform in the sequence corresponding to a first non-null set of target feature vectors,

wherein the waveform selector attributes, to at least one waveform candidate, a cost, wherein the cost is a function of weighted individual costs associated with each of a plurality of features, and wherein the weight associated with at least one of the individual costs varies nontrivially according to a second non-null set of target feature vectors in the sequence; and

a speech waveform concatenator in communication with the speech database that concatenates the waveforms selected by the speech waveform selector to produce a speech signal output. In further embodiments, the first and second sets are identical. Alternatively, the second set is proximate to the first set in the sequence.

[0031] Another embodiment provides a speech synthesizer, and the embodiment includes:

a speech database referencing speech waveforms;
a speech waveform selector, in communication with the speech database, that selects waveforms referenced by the database using designators that correspond to a phonetic transcription input; and
a speech waveform concatenator, in communication with the speech database, that concatenates waveforms selected by the speech waveform selector to produce a speech signal output,

wherein, for at least one ordered sequence of a first waveform and a second waveform, the concatenator selects (i) a location of a trailing edge of the first waveform and (ii) a location of a leading edge of the second waveform, each location being selected so as to produce an optimization of a phase match between the first and second waveforms in regions near the locations.

[0032] In related embodiments, the phase match is achieved by changing the location only of the leading edge and by changing the location only of the trailing edge. Optionally, or in addition, the optimization is determined on the basis of similarity in shape of the first and second waveforms in the regions near the locations. In further embodiments, similarity is determined using a cross-correlation technique, which optionally is normalized cross correlation. Optionally or in addition, the optimization is determined using at least one non-rectangular window. Also optionally or in addition, the optimization is determined in a plurality of successive stages in which time resolution associated with the first and second waveforms is made successively finer. Optionally, or in addition, the change in resolution is achieved by down-

sampling.

Brief Description of the Drawings

[0033] The present invention will be more readily understood by reference to the following detailed description taken with the accompanying drawings, in which:

Fig. 1 illustrates speech synthesizer according to a representative embodiment.
Fig. 2 illustrates the structure of the speech unit database in a representative embodiment.

Detailed Description of Specific Embodiments

Overview

[0034] A representative embodiment of the present invention, known as the RealSpeak™ Text-to-Speech (TTS) engine, produces high quality speech from a phonetic specification, that can be the output of a text processor, known as a target, by concatenating parts of real recorded speech held in a large database. The main process objects that make up the engine, as shown in Fig. 1, include a text processor **101,** a target generator **111,** a speech unit database **141,** a waveform selector 131, and a speech waveform concatenator **151.**

[0035] The speech unit database **141** contains recordings, for example in a digital format such as PCM, of a large corpus of actual speech that are indexed in individual speech units by their phonetic descriptors, together with associated speech unit descriptors of various speech unit features. In one embodiment, speech units in the speech unit database **141** are in the form of a diphone, which starts and ends in two neighboring phonemes. Other embodiments may use differently sized and structured speech units. Speech unit descriptors include, for example, symbolic descriptors *e.g.*, lexical stress, word position, etc.—and prosodic descriptors *e.g.* duration, amplitude, pitch, etc.

[0036] The text processor **101** receives a text input, *e.g.*, the text phrase "Hello, goodbye!" The text phrase is then converted by the text processor 101 into an input phonetic data sequence. In Fig. 1, this is a simple phonetic transcription—#'hE-10#'Gud-bY#. In various alternative embodiments, the input phonetic data sequence may be in one of various different forms. The input phonetic data sequence is converted by the target generator **111** into a multi-layer internal data sequence to be synthesized. This internal data sequence representation, known as extended phonetic transcription (XPT), includes phonetic descriptors, symbolic descriptors, and prosodic descriptors such as those in the speech unit database **141.**

[0037] The waveform selector **131** retrieves from the speech unit database **141** descriptors of candidate speech units that can be concatenated into the target utterance specified by the XPT transcription. The waveform selector 131 creates an ordered list of candidate speech units by comparing the XPTs of the candidate speech units with the XPT of the target XPT, assigning a node cost to each candidate. Candidate-to-target matching is based on symbolic descriptors,such as phonetic context and prosodic context, and numeric descriptors and determines how well each candidate fits the target specification. Poorly matching candidates may be excluded at this point.

[0038] The waveform selector **131** determines which candidate speech units can be concatenated without causing disturbing quality degradations such as clicks, pitch discontinuities, etc. Successive candidate speech units are evaluated by the waveform selector **131** according to a quality degradation cost function. Candidate-to-candidate matching uses frame-based information such as energy, pitch and spectral information to determine how well the candidates can be joined together. Using dynamic programming, the best sequence of candidate speech units is selected for output to the speech waveform concatenator **151.**

[0039] The speech waveform concatenator **151** requests the output speech units (diphones and/or polyphones) from the speech unit database **141** for the speech waveform concatenator **151.** The speech waveform concatenator **151** concatenates the speech units selected forming the output speech that represents the target input text.

[0040] Operation of various aspects of the system will now be described in greater detail.

Speech Unit Database

[0041] As shown in Fig. 2, the speech unit database **141** contains three types of files:

(1) a speech signal file **61**
(2) a time-aligned extended phonetic transcription (XPT) file **62,** and
(3) a diphone lookup table 63.

Database Indexing

[0042] Each diphone is identified by two phoneme symbols - these two symbols are the key to the diphone lookup table **63.** A diphone index table **631** contains an entry for each possible diphone in the language, describing where the references of these diphones can be found in the diphone reference table **632.** The diphone reference table **632** contains references to all the diphones in the speech unit database **141.** These references are alphabetically ordered by diphone identifier. In order to reference all diphones by identity it is sufficient to specify where a list starts in the diphone lookup table **63,** and how many diphones it contains. Each diphone reference contains the number of the message (utterance) where it is found in the speech unit database **141,** which phoneme the diphone starts at, where the diphone starts in the speech signal, and the duration of the diphone.

XPT

[0043] A significant factor for the quality of the system is the transcription that is used to represent the speech signals in the speech unit database **141.** Representative embodiments set out to use a transcription that will allow the system to use the intrinsic prosody in the speech unit database **141** without requiring precise pitch and duration targets. This means that the system can select speech units that are matched phonetically and prosodically to an input transcription. The concatenation of the selected speech units by the speech waveform concatenator **151** effectively leads to an utterance with the desired prosody.

[0044] The XPT contains two types of data: symbolic features (i.e., features that can be derived from text) and acoustic features (i.e., features that can only be derived from the recorded speech waveform). To effectively extract speech units from the speech unit database **141,** the XPT typically contains a time aligned phonetic description of the utterance. The start of each phoneme in the signal is included in the transcription; The XPT also contains a number of prosody related cues, e.g., accentuation and position information. Apart from symbolic information, the transcription also contains acoustic information related to prosody, e.g. the phoneme duration. A typical embodiment concatenates speech units from the speech unit database **141** without modification of their prosodic or spectral realization. Therefore, the boundaries of the speech units should have matching spectral and prosodic realizations. The necessary information required to verify this match is typically incorporated into the XPT by a boundary pitch value and spectral data. The boundary pitch value and the spectrum are calculated at the polyphone edges.

Database Storage

[0045] Different types of data in the speech unit database **141** may be stored on different physical media, *e.g.*, hard disk, CD-ROM, DVD, random-access memory (RAM), etc. Data access speed may be increased by efficiently choosing how to distribute the data between these various media. The slowest accessing component of a computer system is typically the hard disk. If part of the speech unit information needed to select candidates for concatenation were stored on such a relatively slow mass storage device, valuable processing time would be wasted by accessing this slow device. A much faster implementation could be obtained if selection-related data were stored in RAM. Thus in a representative embodiment, the speech unit database **141** is partitioned into frequently needed selection-related data 21—stored in RAM, and less frequently needed concatenation-related data 22—stored, for example, on CD-ROM or DVD. As a result, RAM requirements of the system remain modest, even if the amount of speech data in the database becomes extremely large (~Gbytes). The relatively small number of CD-ROM retrievals may accommodate multi-channel applications using one CD-ROM for multiple threads, and the speech database may reside alongside other application data on the CD (*e.g.*, navigation systems for an auto-PC).

[0046] Optionally, speech waveforms may be coded and/or compressed using techniques well-known in the art.

Waveform Selection

[0047] Initially, each candidate list in the waveform selector **131** contains many available matching diphones in the speech unit database **141**. Matching here means merely that the diphone identities match. Thus in an example of a diphone'#1' in which the initial '1' has primary stress in the target, the candidate list in the waveform selector **131** contains every '#1' found in the speech unit database **141**, including the ones with unstressed or secondary stressed '1'. The waveform selector **131** uses Dynamic Programming (DP) to find the best sequence of diphones so that:

(1) the database diphones in the best sequence are similar to the target diphones in terms of stress, position, context, etc., and
(2) the database diphones in the best sequence can be joined together with low concatenation artifacts.

In order to achieve these goals, two types of costs are used - a *NodeCost* which scores the suitability of each candidate diphone to be used to synthesize a particular target, and a *TransitionCost* which scores the 'joinability' of the diphones. These costs are combined by the DP algorithm, which finds the optimal path.

Cost Functions

[0048] The cost functions used in the unit selection may be of two types depending on whether the features involved are symbolic (*i.e.*, non numeric *e.g.*, stress, prominence, phoneme context) or numeric (*e.g.*, spectrum, pitch, duration).

Cost Functions for Symbolic Features

[0049] For scoring candidates based on the similarity of their symbolic features (*i.e.*, non numeric features) to specified target units, there are 'grey' areas between what is a good match and what is a bad match. The simplest cost weight function would be a binary 0/1. If the candidate has the same value as the target, then the cost is 0; if the candidate is something different, then the cost is 1. For example, when scoring a candidate for its stress (sentence accent (strongest), primary, secondary, unstressed (weakest)) for a target with the strongest stress, this simple system would score primary, secondary or unstressed candidates with a cost of 1. This is counter-intuitive, since if the target is the strongest stress, a candidate of primary stress is preferable to a candidate with no stress.

[0050] To accommodate this, the user can set up tables which describe the cost between any 2 values of a particular symbolic feature. Some examples are shown in Table 1 and Table 2 in the Tables Appendix which are called 'fuzzy tables' because they resemble concepts from fuzzy logic. Similar tables can be set up for any or all of the symbolic features used in the NodeCost calculation.

[0051] Fuzzy tables in the waveform selector 131 may also use special symbols, as defined by the developer linguist, which mean 'BAD' and 'VERY BAD'. In practice, the linguist puts a special symbol /1 for BAD, or /2 for VERY BAD in the fuzzy table, as shown in Table 1 in the Tables Appendix, for a target prominence of 3 and a candidate prominence of 0. It was previously mentioned that the normal minimum contribution from any feature is 0 and the maximum is 1. By using /1 or /2 the cost of feature mismatch can be made much higher than 1, such that the candidate is guaranteed to get a high cost. Thus, if for a particular feature the appropriate entry in the table is /1, then the candidate will rarely be used, and if the appropriate entry in the table is /2, then the candidate will almost never be used. In the example of Table 1, if the target prominence is 3, using a /1 makes it unlikely that a candidate with prominence 0 will ever be selected.

Context Dependent Cost Functions

[0052] The input specification is used to symbolically choose the best combination of speech units from the database which match the input specification. However, using fixed cost functions for symbolic features, to decide which speech units are best, ignores well-known linguistic phenomena such as the fact that some symbolic features are more important in certain contexts than others.

[0053] For example, it is well-known that in some languages phonemes at the end of an utterance, i.e.,the last syllable, tend to be longer than those elsewhere in an utterance. Therefore, when the dynamic programming algorithm searches for candidate speech units to synthesize the last syllable of an utterance, the candidate speech units should also be from utterance-final syllables, and so it is desirable that in utterance-final position, more importance is placed on the feature of "syllable position". This sort of phenomena varies from language to language, and therefore it is useful to have a way of introducing context-dependent speech unit selection in a rule-based framework, so that the rules can be specified by linguistic experts rather than having to manipulate the actual parameters of the waveform selector 131 cost functions directly.

Thus the weights specified for the cost functions may also be manipulated according to a number of rules related to features, e.g. phoneme identities. Additionally, the cost functions themselves may also be manipulated according to rules related to features, e.g. phoneme identities. If the conditions in the rule are met, then several possible actions can occur, such as

(1) For symbolic or numeric features, the weight associated with the feature may be changed — increased if the feature is more important in this context, decreased if the feature is less important. For example, because 'r' often colors vowels before and after it, an expert rule fires when an 'r' in vowel-context is encountered which increases the importance that the candidate items match the target specification for phonetic context.

(2) For symbolic features, the fuzzy table which a feature normally uses may be changed to a different one.

(3) For numeric features, the shape of the cost functions can be changed. Some examples are shown in Table 3 in the Tables Appendix, in which * is used to denote 'any phone', and [] is used to surround the current focus

diphone. Thus r[at]# denotes a diphone 'at' in context r_#.

Scalability

[0054] System scalability is also a significant concern in implementing representative embodiments. The speech unit selection strategy offers several scaling possibilities. The waveform selector **131** retrieves speech unit candidates from the speech unit database **141** by means of lookup tables that speed up data retrieval. The input key used to access the lookup tables represents one scalability factor. This input key to the lookup table can vary from minimal—*e.g.*, a pair of phonemes describing the speech unit core—to more complex—*e.g.*, a pair of phonemes + speech unit features (accentuation, context,...). A more complex the input key results in fewer candidate speech units being found through the lookup table. Thus, smaller (although not necessarily better) candidate lists are produced at the cost of more complex lookup tables.

[0055] The size of the speech unit database **141** is also a significant scaling factor, affecting both required memory and processing speed. The more data that is available, the longer it will take to find an optimal speech unit. The minimal database needed consists of isolated speech units that cover the phonetics of the input (comparable to the speech data bases that are used in linear predictive coding-based phonetics-to-speech systems). Adding well chosen speech signals to the database, improves the quality of the output speech at the cost of increasing system requirements.

[0056] The pruning techniques described above also represents a scalability factor which can speed up unit selection. A further scalability factor relates to the use of a speech coding and/or speech compression techniques to reduce the size of the speech database.

Signal Processing/Concatenation

[0057] The speech waveform concatenator 151 performs concatenation-related signal processing. The synthesizer generates speech signals by joining high-quality speech segments together. Concatenating unmodified PCM speech waveforms in the time domain has the advantage that the intrinsic segmental information is preserved. This implies also that the natural prosodic information, including the micro-prosody, is transferred to the synthesized speech. Although the intra-segmental acoustic quality is optimal, attention should be paid to the waveform joining process that may cause inter-segmental distortions. The major concern of waveform concatenation is in avoiding waveform irregularities such as discontinuities and fast transients that may occur in the neighborhood of the join. These waveform irregularities are generally referred to as concatenation artifacts.

[0058] It is thus important to minimize signal discontinuities at each junction. The concatenation of two segments can be performed by using the well-known weighted overlap-and-add (OLA) method. The overlap and-add procedure for segment concatenation is in fact nothing else than a (non-linear) short time fade-in/fade-out of speech segments. To get high-quality concatenation, we locate a region in the trailing part of the first segment and we locate a region in the leading part of the second segment, such that a phase mismatch measure between the two regions is minimized. This process is performed as follows:

- We search for the maximum normalized cross-correlation between two sliding windows, one in the trailing part of the first speech segment and one in the leading part of the second speech segment.
- The trailing part of the first speech segment and the leading part of the second speech segment are centered around the diphone boundaries as stored in the lookup tables of the database.
- In the preferred embodiment the length of the trailing and leading regions are of the order of one to two pitch periods and the sliding window is bell-shaped.

In order to reduce the computational load of the exhaustive search, the search can be performed in multiple stages. The first stage performs a global search as described in the procedure above on a lower time resolution. The lower time resolution is based on cascaded downsampling of the speech segments. Successive stages perform local searches at successively higher time resolutions around the optimal region determined in the previous stage.

Conclusion

[0059] Representative embodiments can be implemented as a computer program product for use with a computer system. Such implementation may include a series of computer instructions fixed either on a tangible medium, such as a computer readable medium (*e.g.*, a diskette, CD-ROM, ROM, or fixed disk) or transmittable to a computer system, via a modem or other interface device, such as a communications adapter connected to a network over a medium.

The medium may be either a tangible medium (*e.g.*, optical or analog communications lines) or a medium implemented with wireless techniques (*e.g.*, microwave, infrared or other transmission techniques). The series of computer instructions embodies all or part of the functionality previously described herein with respect to the system. Those skilled in the art should appreciate that such computer instructions can be written in a number of programming languages for use with many computer architectures or operating systems. Furthermore, such instructions may be stored in any memory device, such as semiconductor, magnetic, optical or other memory devices, and may be transmitted using any communications technology, such as optical, infrared, microwave, or other transmission technologies. It is expected that such a computer program product may be distributed as a removable medium with accompanying printed or electronic documentation (*e.g.*, shrink wrapped software), preloaded with a computer system (*e.g.*, on system ROM or fixed disk), or distributed from a server or electronic bulletin board over the network (*e.g.*, the Internet or World Wide Web). Of course, some embodiments of the invention may be implemented as a combination of both software (*e.g.*, a computer program product) and hardware. Still other embodiments of the invention are implemented as entirely hardware, or entirely software (*e.g.*, a computer program product).

Glossary

[0060] The definitions below are pertinent to both the present description and the claims following this description.

"Diphone" is a fundamental speech unit composed of two adjacent half-phones. Thus the left and right boundaries of a diphone are in-between phone boundaries. The center of the diphone contains the phone-transition region. The motivation for using diphones rather than phones is that the edges of diphones are relatively steady-state, and so it is easier to join two diphones together with no audible degradation, than it is to join two phones together.

"High level" linguistic features of a polyphone or other phonetic unit include, with respect to such unit, accentuation, phonetic context, and position in the applicable sentence, phrase, word, and syllable.

"Large speech database" refers to a speech database that references speech waveforms. The database may directly contain digitally sampled waveforms, or it may include pointers to such waveforms, or it may include pointers to parameter sets that govern the actions of a waveform synthesizer. The database is considered "large" when, in the course of waveform reference for the purpose of speech synthesis, the database commonly references many waveform candidates, occurring under varying linguistic conditions. In this manner, most of the time in speech synthesis, the database will likely offer many waveform candidates from which to select. The availability of many such waveform candidates can permit prosodic and other linguistic variation in the speech output, as described throughout herein, and particularly in the Overview.

"Low level" linguistic features of a polyphone or other phonetic unit includes, with respect to such unit, pitch contour and duration.

"Non-binary numeric" function assumes any of at least three values, depending upon arguments of the function.

"Polyphone" is more than one diphone joined together. A triphone is a polyphone made of 2 diphones.

"SPT (simple phonetic transcription)" describes the phonemes. This transcription is optionally annotated with symbols for lexical stress, sentence accent, etc... Example (for the word 'worthwhile') : #'werT-'wYl#

"Triphone" has two diphones joined together. It thus contains three components - a half phone at its left border, a complete phone, and a half phone at its right border.

"Weighted overlap and addition of first and second adjacent waveforms" refers to techniques in which adjacent edges of the waveforms are subjected to fade-in and fade-out.

TABLES APPENDIX

XPT: 26 phonemes - 2029.400024 ms - CLASS: S

PHONEME :	#	Y	k	U	d	n	b	i	S	U
DIFF :	0	0	0	0	0	0	0	0	0	0
SYLL_BND :	S	S	A	B	A	B	A	B	A	N
BND_TYPE-> :	N	W	N	S	N	W	N	W	N	N
sent_acc :	U	U	S	S	U	U	U	U	S	S
PROMINENCE :	0	0	3	3	0	0	0	0	3	3
TONE :	X	X	X	X	X	X	X	X	X	X
SYLL_IN_WRD :	F	F	I	I	F	F	F	F	F	F
SYLL_IN_PHRS :	L	1	2	2	M	M	P	P	L	L
syll_count-> :	0	0	1	1	2	2	3	3	4	4
syll_count<- :	0	4	3	3	2	2	1	1	0	0
SYLL_IN_SENT :	I	I	M	M	M	M	M	M	M	M
NR_SYLL_PHRS :	1	5	5	5	5	5	5	5	5	5
WRD_IN_SENT :	I	I	M	M	M	M	M	M	f	f
PHRS_IN_SENT :	n	n	n	n	n	n	n	n	n	n
Phon_Start :	0.0	50.0	120.7	250.7	302.5	325.6	433.1	500.7	582.7	734.7
Mid_F0 :	-48.0	23.7	-48.0	27.4	27.0	25.8	24.0	22.7	-48.0	23.3
Avg_F0 :	-48.0	23.2	-48.0	27.4	26.3	25.7	23.8	22.4	-48.0	23.2
Slope_F0 :	0.0	-28.6	0.0	0.0	-165.8	-2.2	84.2	-34.6	0.0	-29.1
CepVecInd :	37	0	2	1	16	21	8	20	1	0

r	h	i	w	$	z	s	t	I	l	$	s
0	0	0	0	0	0	0	0	0	0	0	0
B	A	B	A	N	B	A	N	N	B	S	A
P	N	W	N	N	W	N	N	N	W	S	N
X	X	X	X	X	X	X	X	X	X	X	X
S	U	U	U	U	U	S	S	S	S	U	S
3	0	0	0	0	0	3	3	3	3	0	3
F	F	F	F	F	F	F	F	F	F	I	F
L	1	1	2	2	2	M	M	M	M	P	L
4	0	0	1	1	1	2	2	2	2	3	4
0	4	4	3	3	3	2	2	2	2	1	0
M	M	M	M	M	M	M	M	M	M	M	F
5	5	5	5	5	5	5	5	5	5	5	5
f	i	i	M	M	M	M	M	M	M	F	F
n	f	f	f	f	f	f	f	f	f	f	f
826.6	894.7	952.7	1023.2	1053.6	1112.7	1188.7	1216.7	1288.7	1368.7	1429.9	1481.8
22.1	20.0	21.4	18.9	20.0	19.5	-48.0	-48.0	21.4	20.0	19.5	-48.0
22.0	20.2	21.3	19.1	19.9	-48.0	-48.0	-48.0	21.2	20.0	19.6	-48.0
-6.9	2.2	-23.1	-5.9	5.5	0.0	0.0	0.0	-27.0	0.0	-9.2	0.0
21	1	22	2	33	11	38	30	25	28	58	35

l	i	p	#
0	0	0	0
N	N	B	S
N	N	P	N
X	X	X	X
S	S	S	U
3	3	3	0
F	F	F	F
L	L	L	L
4	4	4	0
0	0	0	0

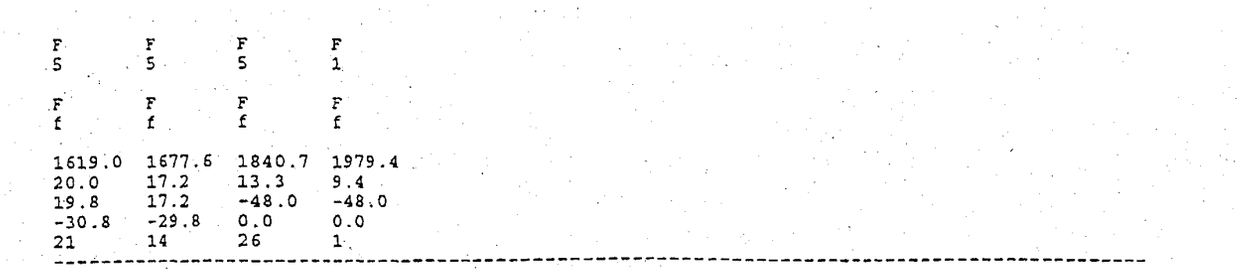


Table 1a -

XPT Transcription Example			
SYMBOLIC FEATURES (XPT)			
name & acronym	applies to	possible values	When?
phonetic differentiator DIFF	phoneme	0 (not annotated)	no annotation symbol present after phoneme
		1 (annotated with first symbol)	first annotation symbol present after phoneme
		2 (annotated with second symbol)	second annotation symbol
		etc	etc
phoneme position in syllable SYLL_BND	phoneme	A(fter syllable boundary)	phoneme after syllable boundary
		B(efore syllable boundary)	phoneme before, but not after, syllable boundary
		S(urrounded by syllable boundaries)	phoneme surrounded by syllable boundaries, or phoneme is silence
		N(ot near syllable boundary)	phoneme not before or after syllable boundary
type of boundary following phoneme BND_TYPE->	phoneme	N(o)	no boundary following phoneme
		S(yllable)	Syllable boundary following phoneme
		W(ord)	Word boundary following phoneme
		P(hrase)	Phrase boundary following phoneme
lexical stress lex_str	syllable	(P)rimary	phoneme in syllable with primary stress
		(S)econdary	phoneme in syllable with secondary stress
		(U)nstressed	phoneme in syllable without lexical stress, or phoneme is silence

Table 1a - (continued)

XPT Transcription Example			
SYMBOLIC FEATURES (XPT)			
name & acronym	applies to	possible values	When?
sentence accent sent_acc	syllable	(S)tressed	phoneme in syllable with sentence accent
		(U)nstressed	phoneme in syllable without sentence accent, of phoneme is silence
prominence PROMINENCE	syllable	0	lex_str = U and sent_acc = U
		1	lex_str = S and sent_acc = U
		2	lex_str = P and sent_acc = U
		3	sent_acc = S
tone value TONE	syllable (mora)	X (missing value)	phoneme in syllable (mora) without tone marker, or phoneme = #, or optional feature is not
		L(ow tone)	supported phoneme in mora with tone = L
		R(ising tone)	phoneme in mora with tone = R
		H(igh tone)	phoneme in mora with tone = H
		F(alling tone)	phoneme in mora with tone = F
syllable position in word SYLL_IN_WRD	syllable	I(nitial)	phoneme in first syllable of multi-syllabic word
		M(edial)	phoneme neither in first nor last syllable of word
		F(inal)	phoneme in last syllable of word (including mono-syllabic words), or phoneme is silence
syllable count in phrase (from first) syll_count->	syllable	0..N-1 (N= nr syll in phrase)	
syllable count in phrase (from last) syll_count<-	syllable	N-1..0 (N= nr syll in phrase)	
syllable position in phrase SYLL_IN_PHRS	syllable	1 (first)	syll_count-> = 0
		2 (second)	syll_count-> = 1
		I (nitial)	syll_count-> < 0.3*N
		M(edial)	all other cases
		F(inal)	syll_count<- < 0.3*N
		P(enultimate)	syll_count<- = N-2
		L(ast)	syll_count<- = N-1

Table 1a - (continued)

XPT Transcription Example			
SYMBOLIC FEATURES (XPT)			
name & acronym	applies to	possible values	When?
syllable position in sentence SYLL_IN_SENT	syllablle	I(nitial) M(edial) F(inal)	first syllable in sentence following initial silence, and initial silence all other cases last syllable in sentence preceding final silence, mono-syllable, and final silence
number of syllables in phrase NR_SYLL_PHRS	phrase	N (number of syll)	
word position in sentence WRD_IN_SENT	word	1(nitial) M(edial) f(inal in phrase, but sentence medial) i(initial in phrase, but sentence medial) F(inal)	first word in sentence not first or last word in sentence or phrase last word in phrase, but not last word in sentence first word in phrase, but not first word in sentence last word in sentence
phrase position in sentence PHRS_IN_SENT	phrase	n(ot final) f(inal)	not last phrase in sentence last phrase in sentence

Table 1b -

XPT Descriptors		
ACOUSTIC FEATURES (XPT)		
name & acronym	applies to	possible values
start of phoneme in signal Phon_Start	phoneme	0..length_of_signal
pitch at diphone boundary in phoneme Mid_F0	diphone boundary	expressed in semitones
average pitch value within the phoneme Avg_F0	phoneme	expressed in semitones
pitch slope within phoneme Slope_F0	phoneme	expressed in semitones per second
cepstral vector index at diphone boundary in phoneme CepVecInd	diphone boundary	unsigned integer value (usually 0..128)

Table 2

Example of a fuzzy table for prominence matching					
		Candidate Prominence			
		0	1	2	3
Target Prominence	0	0	0.1	0.5	1.0
	1	0.2	0	0.1	0.8
	2	0.8	0.3	0	0.2
	3	1.0	1.0	0.3	0

Table 3

Example of a fuzzy table for the left context phone							
		Candidate left context phone					
		a	e	I	p	...	$
Target Left Context Phone	a	0	0.2	0.4	1.0	...	0.8
	e	0.1	0	0.8	1.0	...	0.8
	i	0.9	0.8	0	1.0	...	0.2
	p	1.0	1.0	1.0	0	...	1.0
	...	...	...	...	...	...	...
	$	0.2	0.8	0.8	1.0	...	0

Table 4

Example of a fuzzy table for prominence matching					
		Candidate Prominence			
		0	1	2	3
Target Prominence	0	0	0.1	0.5	1.0
	1	0.2	0	0.1	0.8
	2	0.8	0.3	0	0.2
	3	/1	1.0	0.3	0

Table 5

Examples of context-dependent weight modifications		
Rule	Action	Justification
[r]*	Make the left context more important	r can be colored by the preceding vowel
r[V*]*, V=any vowel	Make the left context more important	The vowel can be colored by the r.
[X], X=unvoiced stop	Make the left context more important	If left context is s then X is not aspirated. This encourages exact matching for s[X*]*, but also includes some side effects.
*[*V]r	Make the right context more important	Vowel coloring
[X]* X=non-sonorant	Make syllable position weights and prominence weights zero.	Sonorants are more sensitive to position and prominence than non-sonorants

Table 6

Transition Cost Calculation Features (Features marked * only 'fire' on accented vowels)				
Feature number	Feature	Lowest cost if....	Highest cost if..	Type of scoring
1	Adjacent in database (*i.e.*, adjacent in donor recorded item)	The two speech units are in adjacent position in same donor word	They are not adjacent	0/1

Table 6 (continued)

Transition Cost Calculation Features (Features marked * only 'fire' on accented vowels)				
Feature number	Feature	Lowest cost if....	Highest cost if..	Type of scoring
2	Pitch difference	There is no pitch difference	There is a big pitch difference	Bigger mismatch = bigger cost (also depends on cost function)
3	Cepstral distance	There is cepstral continuity	There is no cepstral continuity	Bigger mismatch = bigger cost (also depends on cost function)
4	Duration pdf	The duration of the phone (the 2 demiphones joined together) is within expected limits for the target phone ID, accent and position	The duration of the phone is outside that expected for the target phone ID, accent and position	Bigger mismatch = bigger cost
5	Vowel pitch continuity Acc-acc or unacc-unacc (for declination)	Pitch of this accented(unacc) syl is same or slightly lower than the previous accented (unacc) syl in this phrase	Pitch is higher than previous acc (unacc) syl, or pitch is much lower than previous acc (unacc) syl	Flat-bottomed cost function
6	Vowel pitch continuity Unacc -Acc* (for rising pitch from unacc-acc)	Pitch is same or slightly higher than the previous unaccented syllable in this phrase	Pitch is lower than previous unacc syl, or pitch is much higher than previous acc syl.	Flat bottomed asymmetric cost function.

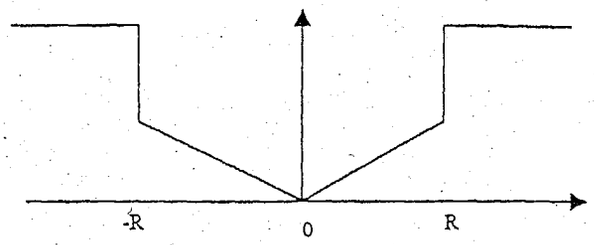
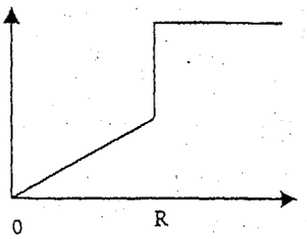
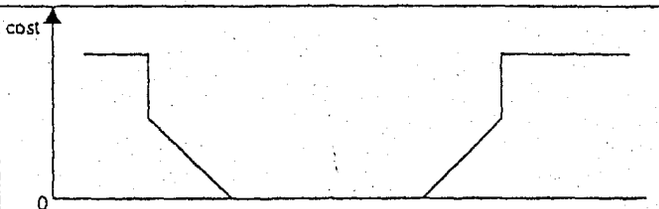
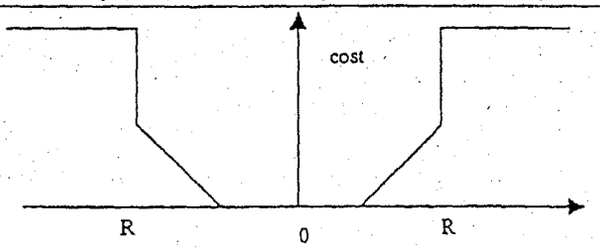
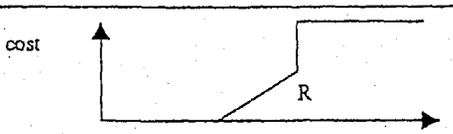
Transition Cost Feature	Shape of cost function
1 Adjacent in database	If items are adjacent cost =0. Otherwise cost=1)
2 Pitch Difference	 Pitch(right demiphone)-pitch(left demiphone) R = range
3 Cepstral Distance	 Cepstral distance between left demiphone and right demiphone
4 Duration PDF	 Duration of phone (=dur of left demiphone+dur of right demiphone)
5 Vowel pitch continuity (I) *	 Pitch(now)-pitch(prev syl with same accentuation)
6 Vowel pitch continuity(II) *	 Pitch(now)-pitch(prev unacc syl)

Table 7 - Weight function shapes used in Transistion Cost calculation

Table 8

Example of a cost function table for categorical variables					
		x2			
		a	e	...	z
x1	a	0.0	0.4	...	0.1
	e	0.1	0.0	...	0.2
	...	...	...	...	...
	z	0.9	1.0	...	0

```
[FEATURES]
CLASS          #$?DFLNPRSV
ACCENT         YN
PHRASEFINAL    YN


[DATA]
# N N  48.300000  114.800000
# N Y   0.000000 1000.000000
# Y N   0.000000 1000.000000
# Y Y   0.000000 1000.000000
$ N N  35.300000   60.700000
$ N Y  56.300000   93.900000
$ Y N   0.000000 1000.000000
$ Y Y   0.000000 1000.000000
? N N  50.900000   84.000000
? N Y  59.200000   89.400000
? Y N  51.400000   83.500000
? Y Y  51.500000   88.400000
D N N  96.400000  148.700000
D N Y 154.000000  249.500000
D Y N 117.400000  174.400000
D Y Y 176.800000  275.500000
F N N  39.000000   90.100000
F Y N  56.200000  122.90000
```

Table 9 - Duration PDF Table

Claims

1. A speech synthesizer comprising:

a. a speech database referencing speech waveforms;
b. a speech waveform selector, in communication with the speech database, that selects waveforms referenced by the database using designators that correspond to a phonetic transcription input; and
c. a speech waveform concatenator, in communication with the speech database, that concatenates waveforms selected by the speech waveform selector to produce a speech signal output,

wherein, for at least one ordered sequence of a first waveform and a second waveform, the concatenator

selects (i) a location of a trailing edge of the first waveform and (ii) a location of a leading edge of the second waveform, each location being selected so as to produce an optimization of a phase match between the first and second waveforms in regions near the locations.

2. A speech synthesizer comprising:

a. a speech database referencing speech waveforms;
b. a speech waveform selector, in communication with the speech database, that selects waveforms referenced by the database using designators that correspond to a phonetic transcription input; and
c. a speech waveform concatenator, in communication with the speech database, that concatenates waveforms selected by the speech waveform selector to produce a speech signal output,

wherein, for at least one ordered sequence of a first waveform and a second waveform, the concatenator selects the location of a trailing edge of the first waveform, the location being selected so as to produce an optimization of a phase match between the first and second waveforms in regions near the location and a leading edge of the second waveform.

3. A speech synthesizer comprising:

a. a speech database referencing speech waveforms;
b. a speech waveform selector, in communication with the speech database, that selects waveforms referenced by the database using designators that correspond to a phonetic transcription input; and
c. a speech waveform concatenator, in communication with the speech database, that concatenates waveforms selected by the speech waveform selector to produce a speech signal output,

wherein, for at least one ordered sequence of a first waveform and a second waveform, the concatenator selects the location of a leading edge of the second waveform, the location being selected so as to produce an optimization of a phase match between the first and second waveforms in regions near the location and a trailing edge of the first waveform.

4. A speech synthesizer according to any of claims 1 through 3, wherein the optimization is determined on the basis of similarity in shape of the first and second waveforms in the regions near the locations.

5. A speech synthesizer according to 4, wherein similarity is determined using a cross-correlation technique.

6. A speech synthesizer according to claim 5, wherein the technique is normalized cross correlation.

7. A speech synthesizer according to any of claims 1 through 3 and 5, wherein the optimization is determined using at least one non-rectangular window.

8. A speech synthesizer according to any of claims 1 through 3, and 5, wherein the optimization is determined in a plurality of successive stages in which time resolution associated with the first and second waveforms is made successively finer.

9. A speech synthesizer according to claim 8, wherein the reduction in time resolution is achieved by waveform downsampling.

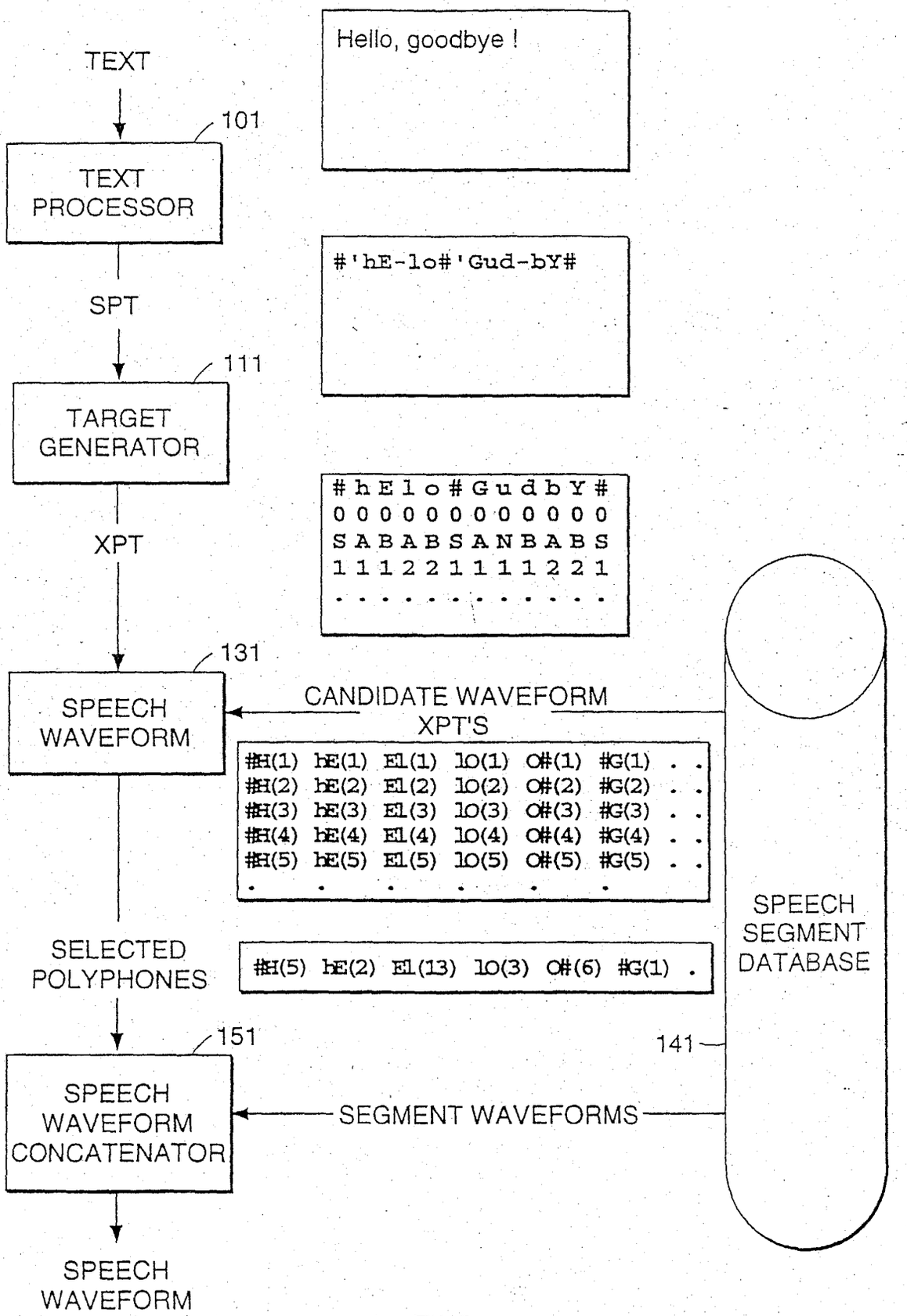

TEXT
Hello, goodbye !
101
TEXT PROCESSOR
#'hE-lo#'Gud-bY#
SPT
111
TARGET GENERATOR
h E l o # G u d b Y
0 0 0 0 0 0 0 0 0 0 0 0
S A B A B S A N B A B S
1 1 1 2 2 1 1 1 1 2 2 1
XPT
131
SPEECH WAVEFORM
CANDIDATE WAVEFORM XPT'S
#H(1) hE(1) El(1) lO(1) O#(1) #G(1)
#H(2) hE(2) El(2) lO(2) O#(2) #G(2)
#H(3) hE(3) El(3) lO(3) O#(3) #G(3)
#H(4) hE(4) El(4) lO(4) O#(4) #G(4)
#H(5) hE(5) El(5) lO(5) O#(5) #G(5)
SPEECH SEGMENT DATABASE
SELECTED POLYPHONES
#H(5) hE(2) El(13) lO(3) O#(6) #G(1)
151
141
SPEECH WAVEFORM CONCATENATOR
SEGMENT WAVEFORMS
SPEECH WAVEFORM

FIG. 1

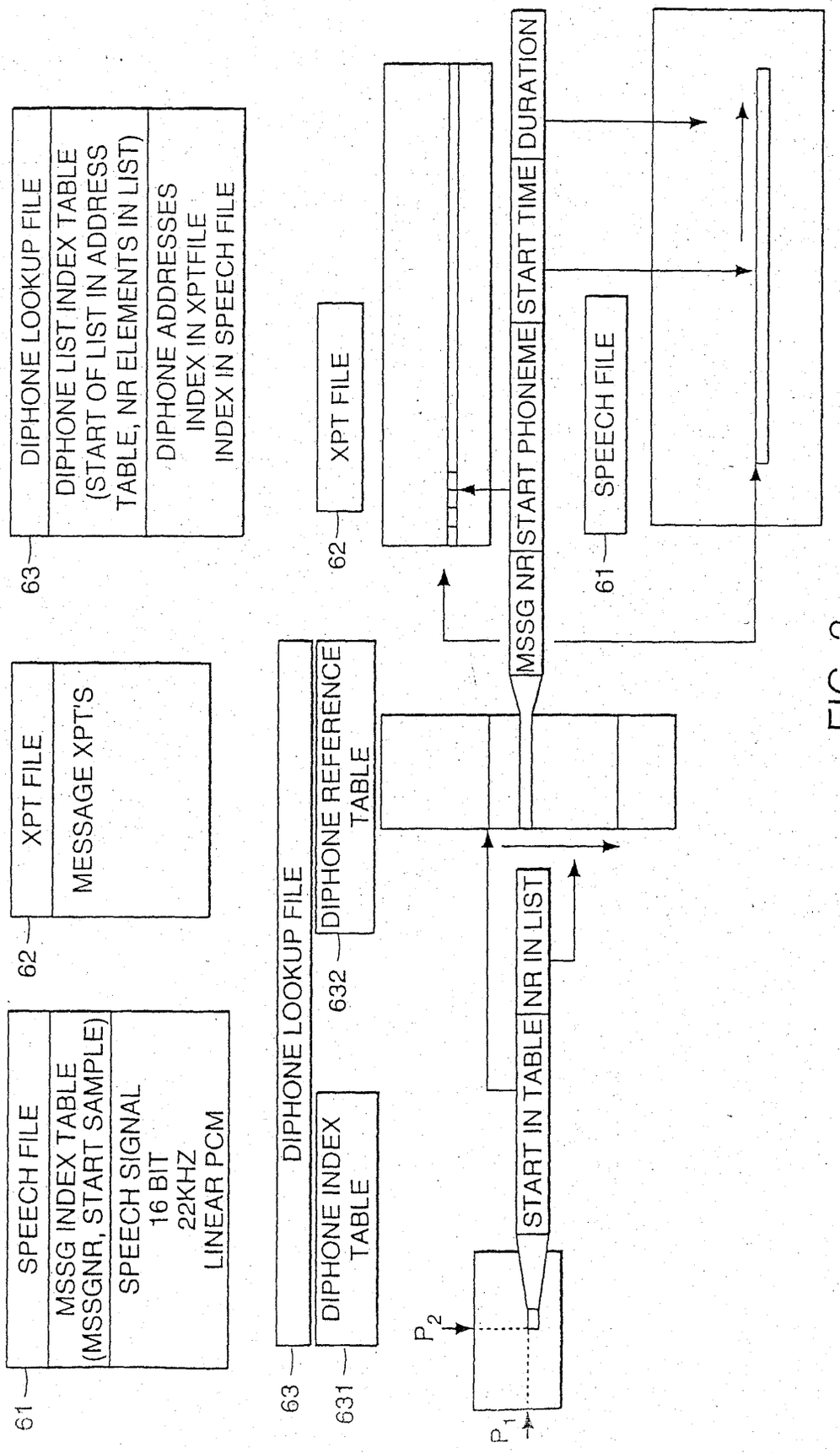


FIG. 2

ORGANISATION OF THE SPEECH UNIT DATABASE