US00668187B1

(12) **United States Patent**
Conkie

(10) **Patent No.: US 6,684,187 B1**
(45) **Date of Patent: Jan. 27, 2004**

(54) **METHOD AND SYSTEM FOR PRESELECTION OF SUITABLE UNITS FOR CONCATENATIVE SPEECH**

(75) Inventor: **Alistair D. Conkie**, Morristown, NJ (US)

(73) Assignee: **AT&T Corp.**, New York, NY (US)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 461 days.

(21) Appl. No.: **09/607,615**

(22) Filed: **Jun. 30, 2000**

(51) **Int. Cl.**[7] **G10L 13/08**
(52) **U.S. Cl.** **704/260**; 704/258; 704/266
(58) **Field of Search** 704/258, 260, 704/251–257, 239–242, 268, 266

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,659,664	A		8/1997	Kaja	
5,794,197	A	*	8/1998	Alleva et al.	704/255
5,978,764	A		11/1999	Lowry et al.	
6,041,300	A		3/2000	Ittycheriah et al.	
6,163,769	A	*	12/2000	Acero et al.	704/260
6,173,263	B1	*	1/2001	Conkie	704/260
6,317,712	B1	*	11/2001	Kao et al.	704/256
6,366,883	B1		4/2002	Campbell et al.	
2001/0044724	A1	*	11/2001	Hon et al.	704/260

FOREIGN PATENT DOCUMENTS

EP	0942409	A2		9/1999
EP	0942409	A3		1/2000
GB	2313530	A		11/1997
JP	06095696	A	*	4/1994 ..... 704/258
WO	WO 00/30069			5/2000

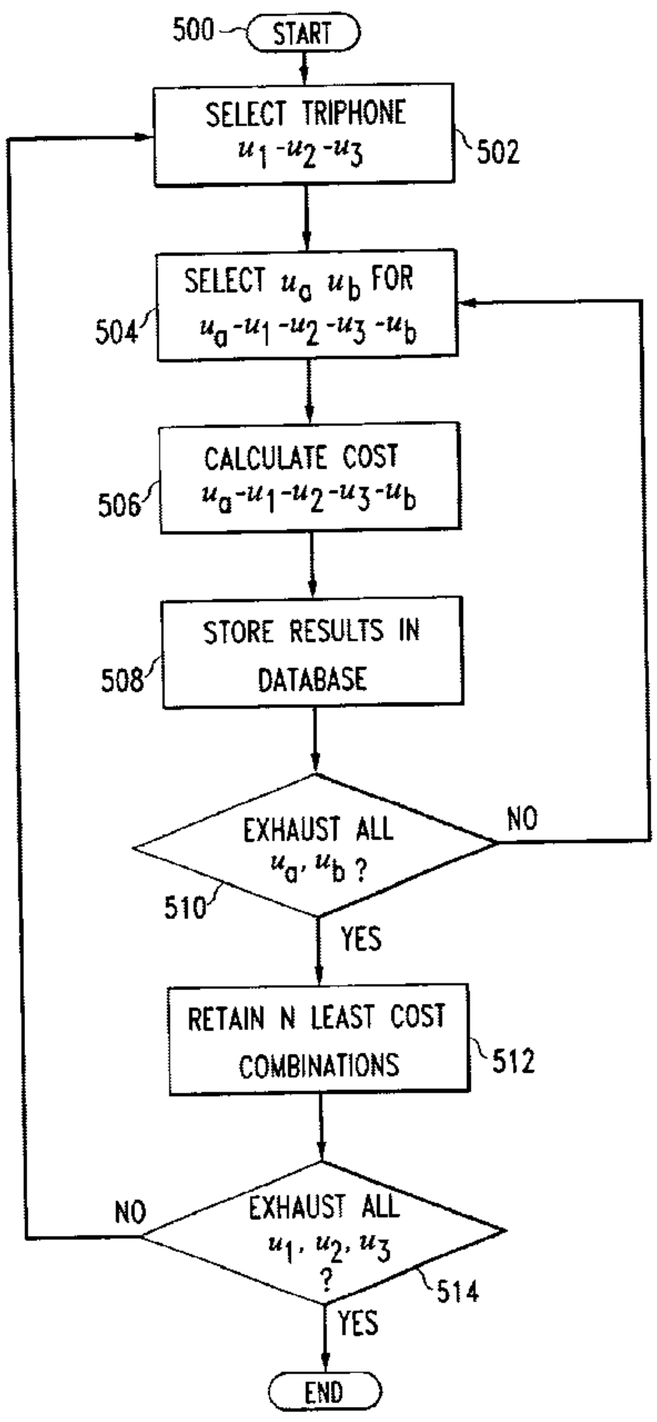
* cited by examiner

Primary Examiner—Abul Azad

(57) **ABSTRACT**

A system and method for improving the response time of text-to-speech synthesis utilizes "triphone contexts" (i.e., triplets comprising a central phoneme and its immediate context) as the basic unit, instead of performing phoneme-by-phoneme synthesis. Prior to initiating the "real time" synthesis, a database is created of ail possible triphones (there are approximately 10000 in the English language) and their associated preselection costs. At run time, therefore, only the most likely candidates are selected from the triphone database, significantly reducing the calculations that are required to be performed in real time.

7 Claims, 5 Drawing Sheets



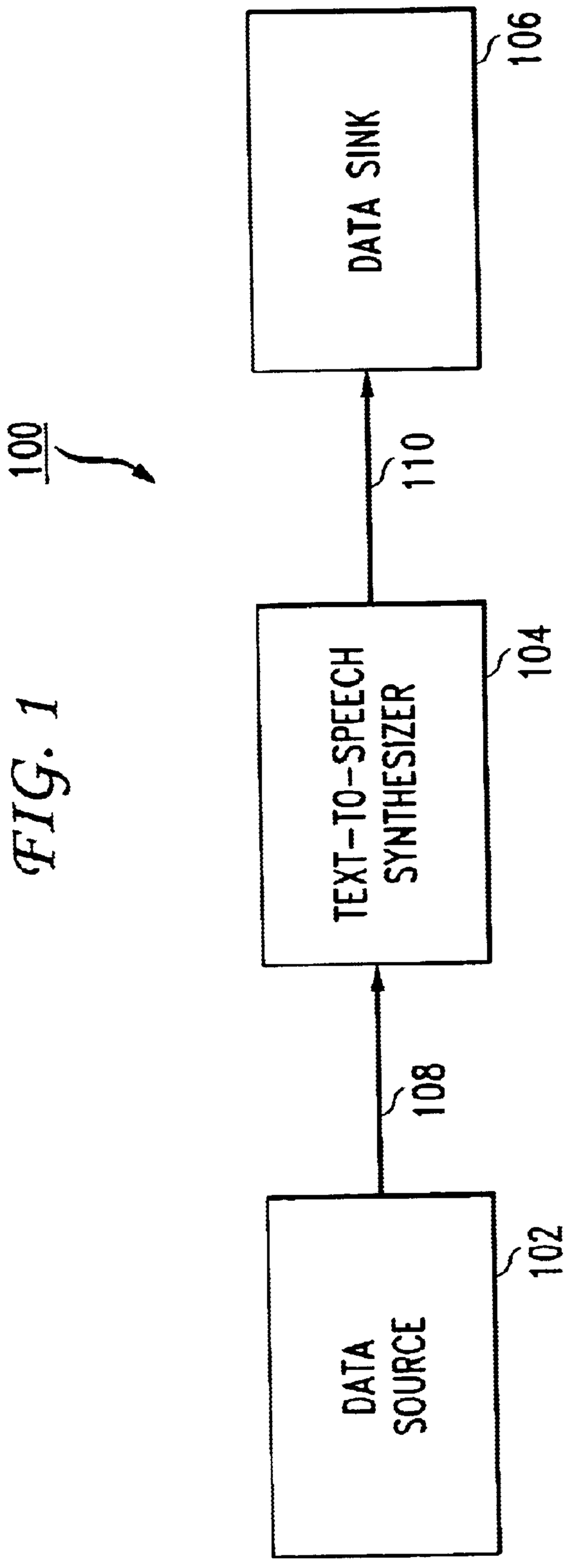

100
DATA SOURCE
102
108
TEXT-TO-SPEECH SYNTHESIZER
104
110
DATA SINK
106

FIG. 1

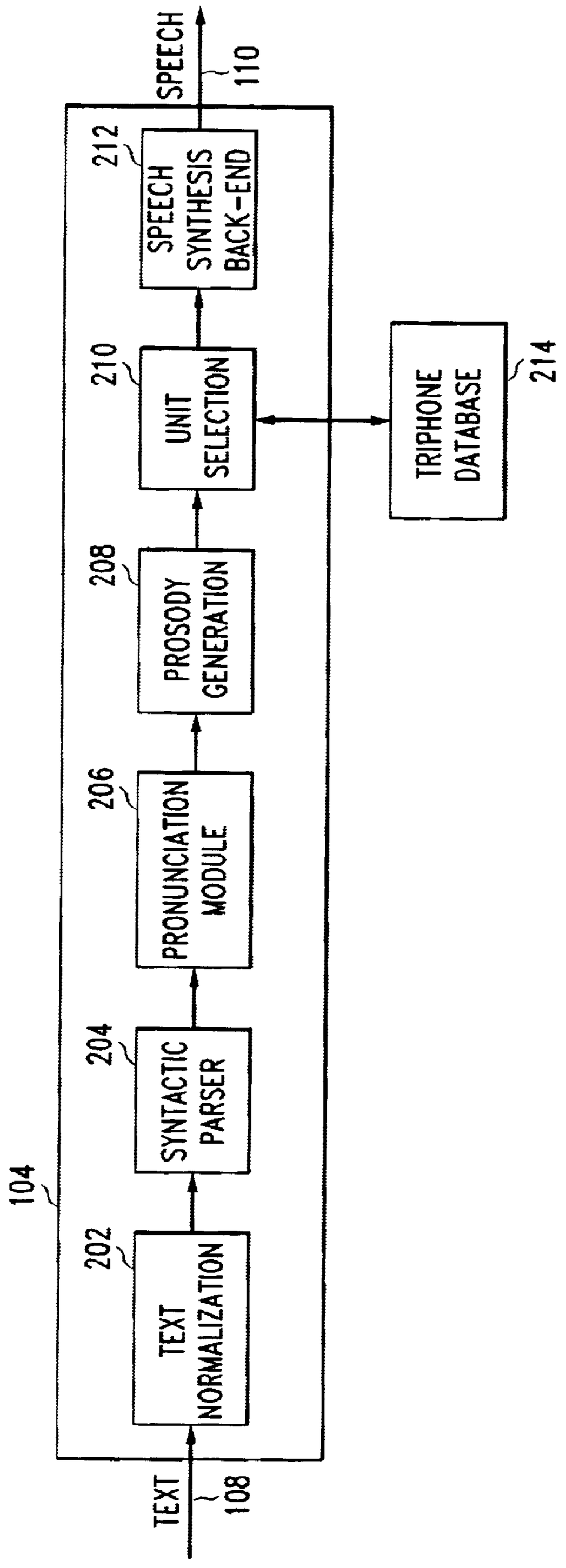

TEXT
108
TEXT NORMALIZATION
202
SYNTACTIC PARSER
204
PRONUNCIATION MODULE
206
PROSODY GENERATION
208
UNIT SELECTION
210
TRIPHONE DATABASE
214
SPEECH SYNTHESIS BACK-END
212
SPEECH
110
104

FIG. 2

FIG. 3

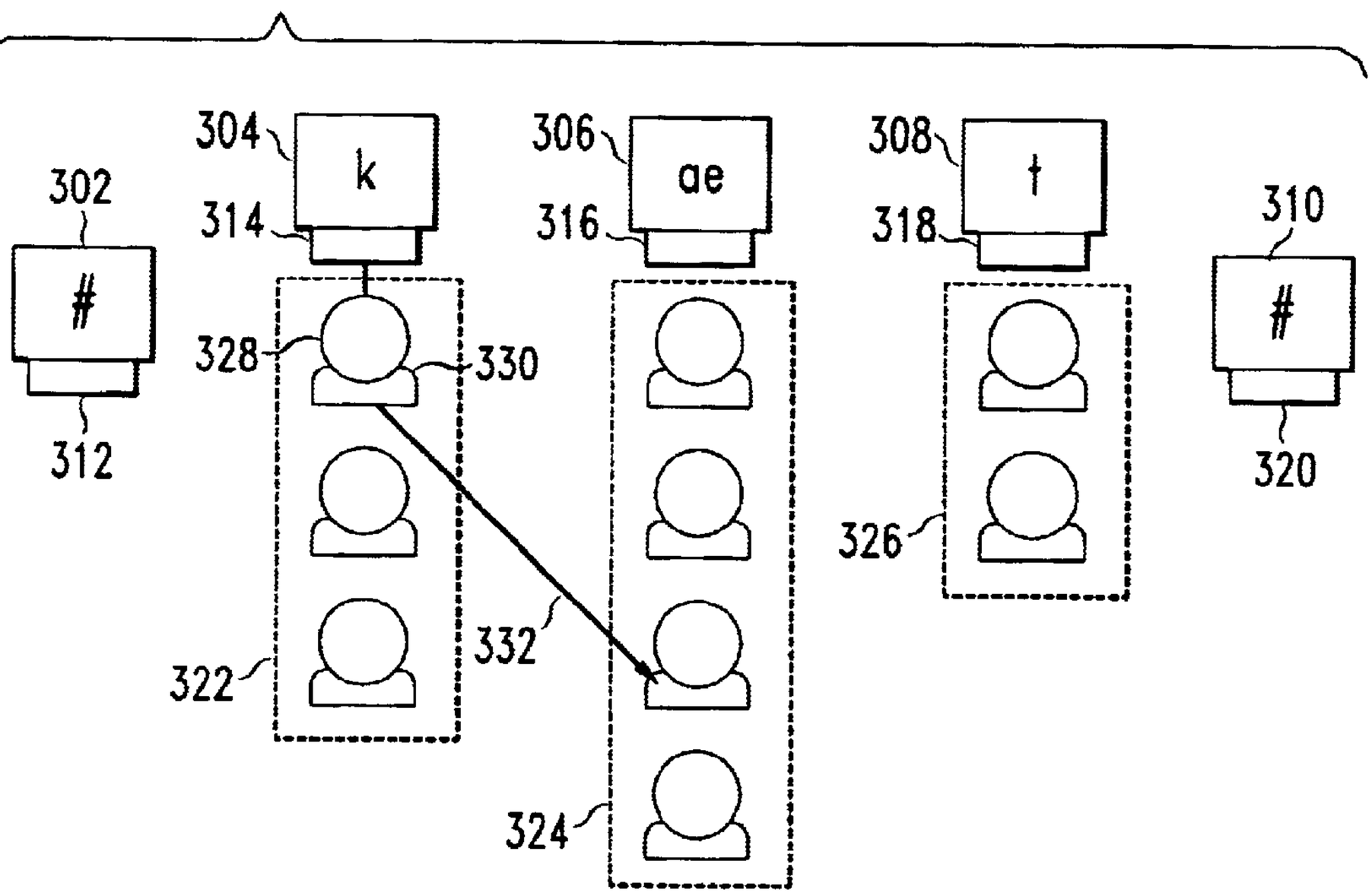

302
#
312
304
k
314
306
ae
316
308
t
318
310
#
320
328
330
322
332
324
326

FIG. 4

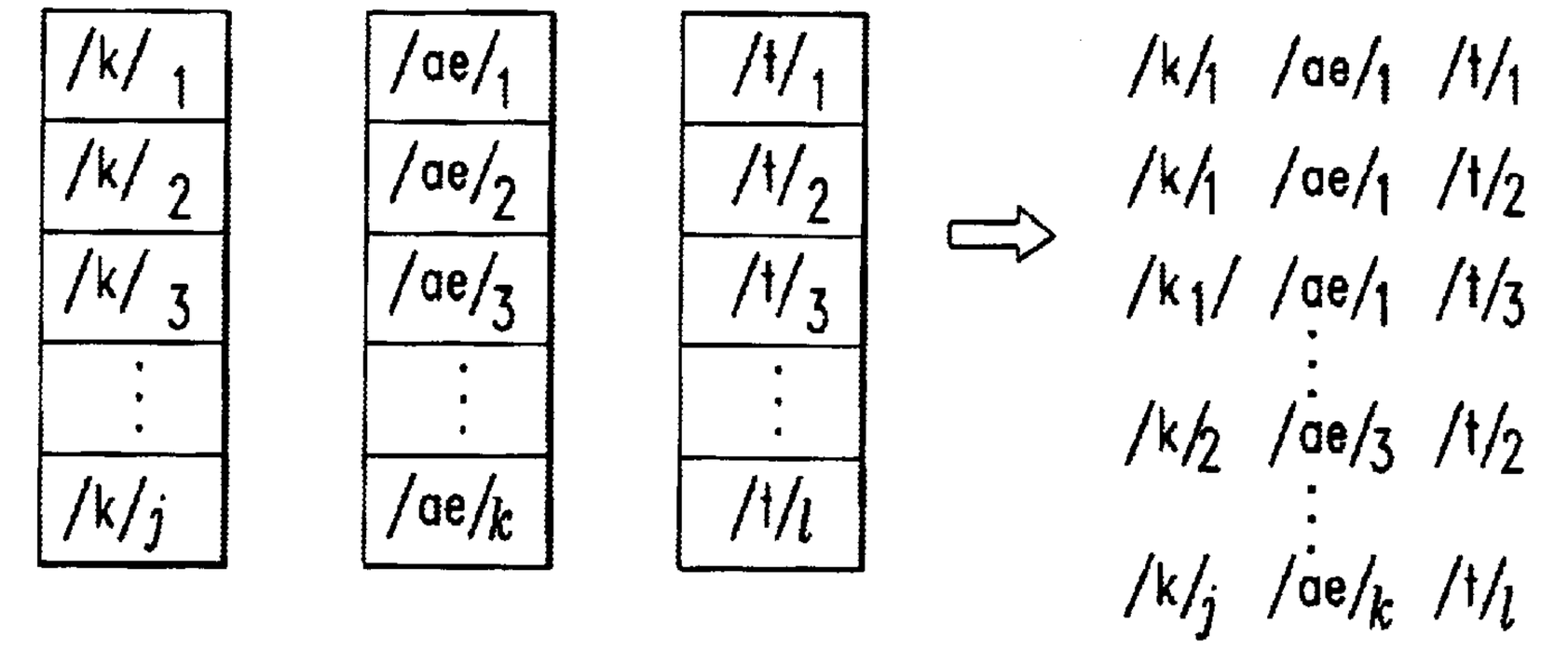

/k/ 1
/k/ 2
/k/ 3
/k/j
/ae/1
/ae/2
/ae/3
/ae/k
/t/1
/t/2
/t/3
/t/l
/k/1 /ae/1 /t/1
/k/1 /ae/1 /t/2
/k1/ /ae/1 /t/3
/k/2 /ae/3 /t/2
/k/j /ae/k /t/l

FIG. 5

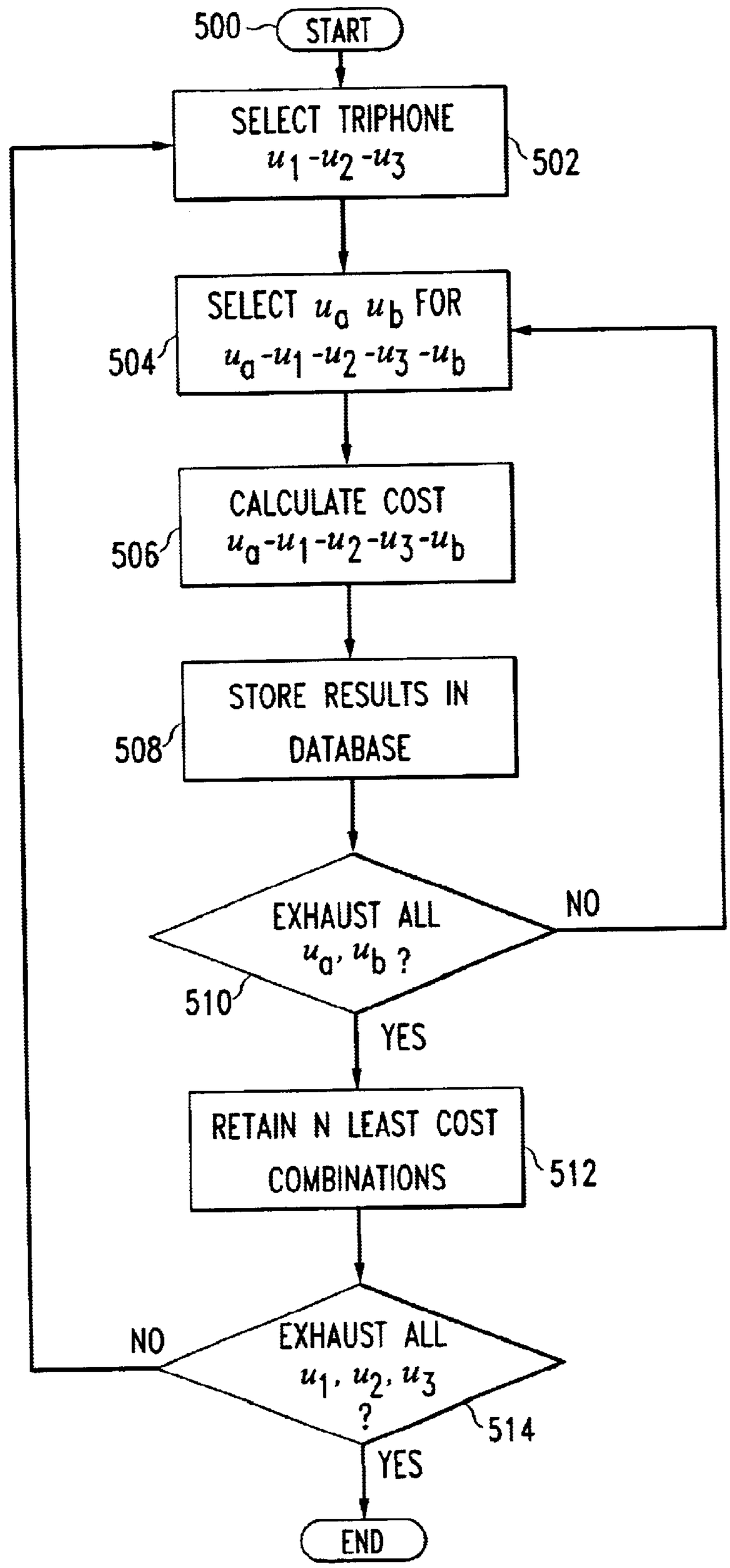
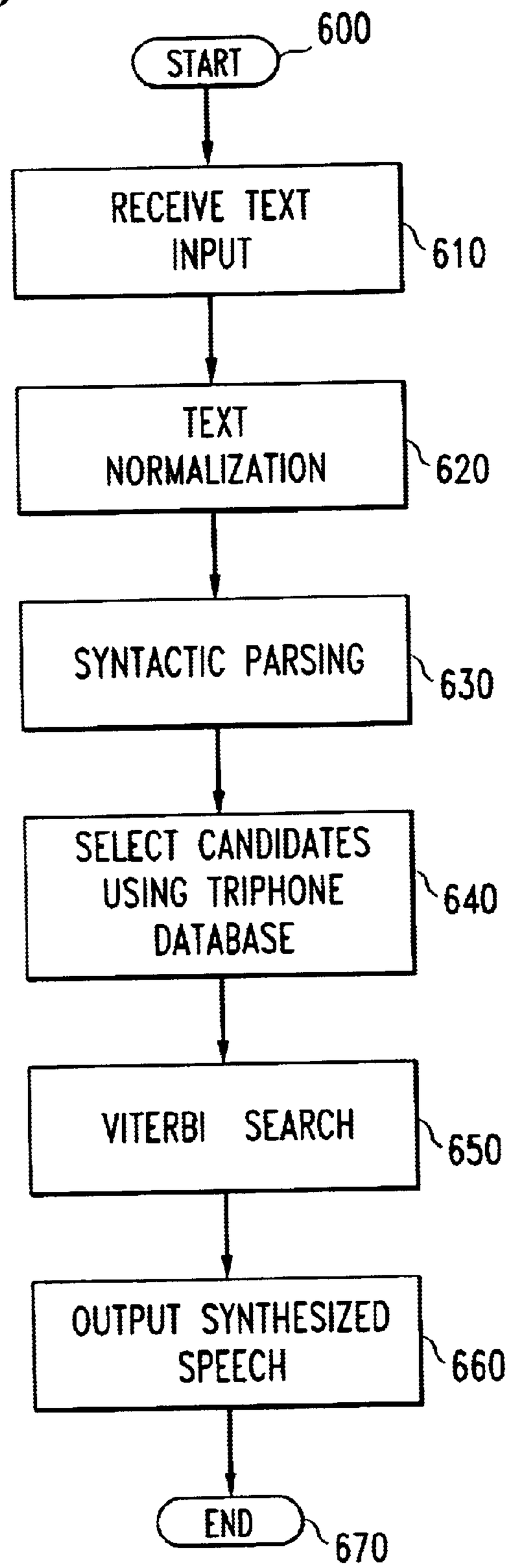

500
START
SELECT TRIPHONE
u_1-u_2-u_3
502
SELECT u_a u_b FOR
u_a-u_1-u_2-u_3-u_b
504
CALCULATE COST
u_a-u_1-u_2-u_3-u_b
506
STORE RESULTS IN
DATABASE
508
EXHAUST ALL
u_a, u_b ?
510
NO
YES
RETAIN N LEAST COST
COMBINATIONS
512
EXHAUST ALL
u_1, u_2, u_3
?
514
NO
YES
END

FIG. 6


START
600
RECEIVE TEXT INPUT
610
TEXT NORMALIZATION
620
SYNTACTIC PARSING
630
SELECT CANDIDATES USING TRIPHONE DATABASE
640
VITERBI SEARCH
650
OUTPUT SYNTHESIZED SPEECH
660
END
670

METHOD AND SYSTEM FOR PRESELECTION OF SUITABLE UNITS FOR CONCATENATIVE SPEECH

TECHNICAL FIELD

The present invention relates to a system and method for increasing the speed of a unit selection synthesis system for concatenative speech synthesis and, more particularly, to predetermining a universe of phonemes—selected on the basis of their triphone context—that are potentially used in speech. Real-time selection is then performed from the created phoneme universe.

BACKGROUND OF THE INVENTION

A current approach to concatenative speech synthesis is to use a very large database for recorded speech that has been segmented and labeled with prosodic and spectral characteristics, such as the fundamental frequency (F**0**) for voiced speech, the energy or gain of the signal, and the spectral distribution of the signal (i.e., how much of the signal is present at any given frequency). The database contains multiple instances of speech sounds. This multiplicity permits the possibility of having units in the database that are much less stylized than would occur in a diphone database (a "diphone" being defined as the second half of one phoneme followed by the initial half of the following phoneme, a diphone database generally containing only one instance of any given diphone). Therefore, the possibility of achieving natural speech is enhanced with the "large database" approach.

For good quality synthesis, this database technique relies on being able to select the "best" units from the database—that is, the units that are closest in character to the prosodic specification provided by the speech synthesis system, and that have a low spectral mismatch at the concatenation points between phonemes. The "best"sequence of units may be determined by associating a numerical cost in two different ways. First, a "target cost" is associated with the individual units in isolation, where a lower cost is associated with a unit that has characteristics (e.g., F**0**, gain, spectral distribution) relatively close to the unit being synthesized, and a higher cost is associated with units having a higher discrepancy with the unit being synthesized. A second cost, referred to as the "concatenation cost", is associated with how smoothly two contiguous units are joined together. For example, if the spectral mismatch between units is poor, perhaps even corresponding to an audible "click", there will be a higher concatenation cost.

Thus, a set of candidate units for each position in the desired sequence can be formulated, with associated target costs and concatenative costs. Estimating the best (lowest-cost) path through the network is then performed using a Viterbi search. The chosen units may then be concatenated to form one continuous signal, using a variety of different techniques.

While such database-driven systems may produce a more natural sounding voice quality, to do so they require a great deal of computational resources during the synthesis process. Accordingly, there remains a need for new methods and systems that provide natural voice quality in speech synthesis while reducing the computational requirements.

SUMMARY OF THE INVENTION

The need remaining in the prior art is addressed by the present invention, which relates to a system and method for

increasing the speed of a unit selection synthesis system for concatenative speech and, more particularly, to predetermining a universe of phonemes in the speech database, selected on the basis of their triphone context, that are potentially used in speech, and performing real-time selection from this precalculated phoneme universe.

In accordance with the present invention, a triphone database is created where for any given triphone context required for synthesis, there is a complete list, precalculated, of all the units (phonemes) in the database that can possibly be used in that triphone context. Advantageously, this list is (in most cases) a significantly smaller set of candidates units than the complete set of units of that phoneme type. By ignoring units that are guaranteed not to be used in the given triphone context, the selection process speed is significantly increased. It has also been found that speech quality is not compromised with the unit selection process of the present invention.

Depending upon the unit required for synthesis, as well as the surrounding phoneme context, the number of phonemes in the preselection list will vary and may, at one extreme, include all possible phonemes of a particular type. There may also arise a situation where the unit to be synthesized (plus context) does not match any of the precalculated triphones. In this case, the conventional single phoneme approach of the prior art may be employed, using the complete set of phonemes of a given type. It is presumed that these instances will be relatively infrequent.

Other and further aspects of the present invention will become apparent during the course of the following discussion and by reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Referring now to the drawings,

FIG. **1** illustrates an exemplary speech synthesis system for utilizing the unit (e.g., phoneme) selection arrangement of the present invention;

FIG. **2** illustrates, in more detail, an exemplary text-to-speech synthesizer that may be used in the system of FIG. **1**;

FIG. **3** illustrates an exemplary "phoneme" sequence and the various costs associated with this sequence;

FIG. **4** contains an illustration of an exemplary unit (phoneme) database useful as the unit selection database in the system of FIG. **1**;

FIG. **5** is a flowchart illustrating the triphone cost precalculation process of the present invention, where the top N units are selected on the basis of cost (the top 50 units for any 5-phone sequence containing a given triphone being guaranteed to be present); and

FIG. **6** is a flowchart illustrating the unit (phoneme) selection process of the present invention, utilizing the precalculated triphone-indexed list of units (phonemes).

DETAILED DESCRIPTION

An exemplary speech synthesis system **100** is illustrated in FIG. **1**. System **100** includes a text-to-speech synthesizer **104** that is connected to a data source **102** through an input link **108**, and is likewise connected to a data sink **106** through an output link **110**. Text-to-speech synthesizer **104**, as discussed in detail below in association with FIG. **2**, functions to convert the text data either to speech data or physical speech. In operation, synthesizer **104** converts the text data by first converting the text into a stream of phonemes representing the speech equivalent of the text,

then processes the phoneme stream to produce an acoustic unit stream representing a clearer and more understandable speech representation. Synthesizer **104** then converts the acoustic unit stream to speech data or physical speech. In accordance with the teachings of the present invention, as discussed in detail below, database units (phonemes) accessed according to their triphone context, are processed to speed up the unit selection process.

Data source **102** provides text-to-speech synthesizer **104**, via input link **108**, the data that represents the text to be synthesized. The data representing the text of the speech can be in any format, such as binary, ASCII, or a word processing file. Data source **102** can be any one of a number of different types of data sources, such as a computer, a storage device, or any combination of software and hardware capable of generating, relaying, or recalling from storage, a textual message or any information capable of being translated into speech. Data sink **106** receives the synthesized speech from text-to-speech synthesizer **104** via output link **110**. Data sink **106** can be any device capable of audibly outputting speech, such as a speaker system for transmitting mechanical sound waves, or a digital computer, or any combination of hardware and software capable of receiving, relaying, storing, sensing or perceiving speech sound or information representing speech sounds.

Links **108** and **110** can be any suitable device or system for connecting data source **102**/data sink **106** to synthesizer **104**. Such devices include a direct serial/parallel cable connection, a connection over a wide area network (WAN) or a local area network (LAN), a connection over an intranet, the Internet, or any other distributed processing network or system. Additionally, input link **108** or output link **110** may be software devices linking various software systems.

FIG. **2** contains a more detailed block diagram of text-to-speech synthesizer **104** of FIG. **1**. Synthesizer **104** comprises, in this exemplary embodiment, a text normalization device **202**, syntactic parser device **204**, word pronunciation module **206**, prosody generation device **208**, an acoustic unit selection device **210**, and a speech synthesis back-end device **212**. In operation, textual data is received on input link **108** and first applied as an input to text normalization device **202**. Text normalization device **202** parses the text data into known words and further converts abbreviations and numbers into words to produce a corresponding set of normalized textual data. For example, if "St." is input, text normalization device **202** is used to pronounce the abbreviation as either "saint" or "street", but not the /st/ sound. Once the text has been normalized, it is input to syntactic parser **204**. Syntactic processor **204** performs grammatical analysis of a sentence to identify the syntactic structure of each constituent phrase and word. For example syntactic parser **204** will identify a particular phrase as a "noun phrase" or a "verb phrase" and a word as a noun, verb, adjective, etc. Syntactic parsing is important because whether the word or phrase is being used as a noun or a verb may affect how it is articulated. For example, in the sentence "the cat ran away", if "cat" is identified as a noun and "ran" is identified as a verb, speech synthesizer **104** may assign the word "cat" a different sound duration and intonation pattern than "ran" because of its position and function in the sentence structure.

Once the syntactic structure of the text has been determined, the text is input to word pronunciation module **206**. In word pronunciation module **206**, orthographic characters used in the normal text are mapped into the appropriate strings of phonetic segments representing units of sound and speech. This is important since the same orthographic strings may have different pronunciations depending on the word in which the string is used. For example, the orthographic string "gh" is translated to the phoneme /f/ in "tough", to the phoneme /g/ in "ghost", and is not directly realized as any phoneme in "though". Lexical stress is also marked. For example, "record" has a primary stress on the first syllable if it is a noun, but has the primary stress on the second syllable if it is a verb. The output from word pronunciation module **206**, in the form of phonetic segments, is then applied as an input to prosody determination device **208**. Prosody determination device **208** assigns patterns of timing and intonation to the phonetic segment strings. The timing pattern includes the duration of sound for each of the phonemes. For example, the "re" in the verb "record" has a longer duration of sound than the "re" in the noun "record". Furthermore, the intonation pattern concerning pitch changes during the course of an utterance. These pitch changes express accentuation of certain words or syllables as they are positioned in a sentence and help convey the meaning of the sentence. Thus, the patterns of timing and intonation are important for the intelligibility and naturalness of synthesized speech. Prosody may be generated in various ways including assigning an artificial accent or providing for sentence context. For example, the phrase "This is a test!" will be spoken differently from "This is a lest?". Prosody generating devices are well-known to those of ordinary skill in the art and any combination of hardware, software, firmware, heuristic techniques, databases, or any other apparatus or method that performs prosody generation may be used. In accordance with the present invention, the phonetic output and accompanying prosodic specification from prosody determination device **208** is then converted, using any suitable, well-known technique, into unit (phoneme) specifications.

The phoneme data, along with the corresponding characteristic parameters, is then sent to acoustic unit selection device **210** where the phonemes and characteristic parameters are transformed into a stream of acoustic units that represent speech. An "acoustic unit" can be defined as a particular utterance of a given phoneme. Large numbers of acoustic units, as discussed below in association with FIG. **3**, may all correspond to a single phoneme, each acoustic unit differing from one another in terms of pitch, duration, and stress (as well as other phonetic or prosodic qualities). In accordance with the present invention, a triphone preselection cost database **214** is accessed by unit selection device **210** to provide a candidate list of units, based on a triphone context, that are most likely to be used in the synthesis process. Unit selection device **210** then performs a search on this candidate list (using a Viterbi search, for example), to find the "least cost" unit that best matches the phoneme to be synthesized. The acoustic unit stream output from unit selection device **210** is then sent to speech synthesis back-end device **212** which converts the acoustic unit stream into speech data and transmits (referring to FIG. **1**) the speech data to data sink **106** over output link **110**.

FIG. **3** contains an example of a phoneme string **302–310** for the word "cat" with an associated set of characteristic parameters **312–320** (for example, F**0**, duration, etc.) assigned, respectively, to each phoneme and a separate list of acoustic unit groups **322**, **324** and **326** for each utterance. Each acoustic unit group includes at least one acoustic unit **328** and each acoustic unit **328** includes an associated target cost **330**, as defined above. A concatenation cost **332**, as represented by the arrow in FIG. **3**, is assigned between each acoustic unit **328** in a given group and an acoustic units **332** of the immediately subsequent group.

In the prior art, the unit selection process was performed on a phoneme-by-phoneme basis (or, in more robust systems, on half-phoneme—by—half-phoneme basis) for every instance of each unit contained in the speech database. Thus, when considering the /æ/ phoneme **306**, each of its acoustic unit realizations **328** in speech database **324** would be processed to determine the individual target costs **330**, compared to the text to be synthesized. Similarly, phoneme-by-phoneme processing (during run time) would also be required for /k/ phoneme **304** and /t/ phoneme **308**. Since there are many occasions of the phoneme /æ/ that would not be preceded by /k/ and/or followed by /t/, there were many target costs in the prior art systems that were likely to be unnecessarily calculated.

In accordance with the present invention, it has been recognized that run-time calculation time can be significantly reduced by pre-computing the list of phoneme candidates from the speech database that can possibly be used in the final synthesis before beginning to work out target costs. To this end, a "triphone" database (illustrated as database **214** in FIG. **2**) is created where lists of units (phonemes) that might be used in any given triphone context are stored (and indexed using a triphone-based key) and can be accessed during the process of unit selection. For the English language, there are approximately 10,000 common triphones, so the creation of such a database is not an insurmountable task. In particular, for the triphone /k/-/æ/-/t/, each possible /æ/ in the database is examined to determine how well it (and the surrounding phonemes that occur in the speech from which it was extracted) matches the synthesis specifications, as shown in FIG. **4**. By then allowing the phonemes on either side of /k/ and /t/ to vary over the complete universe of phonemes all possible costs can be examined that may be calculated at run-time for a particular phoneme in a triphone context. In particular, when synthesis is complete, only the N "best" units are retained for any 5-phoneme context (in terms of lowest concatenation cost; in one example N may be equal to 50). It is possible to "combine" (i.e., take the union of) the relevant units that have a particular triphone in common. Because of the way this calculation is arranged, the combination is guaranteed to be the list of all units that are relevant for this specific part of the synthesis.

In most cases, there will be number of units (i.e., specific instances of the phonemes) that will not occur in the union of possible all units, and therefore need never be considered in calculating the costs at run time. The preselection process of the present invention, therefore, results in increasing the speed of the selection process. In one instance, an increase of 100% has been achieved. It is to be presumed that if a particular triphone does not appear to have an associated list of units, the conventional unit cost selection process will be used.

In general, therefore, for any unit us that is to be synthesized as part of the triphone sequence u_1-u_2-u_3, the preselection cost for every possible 5-phone combination u_a-u_1-u_2-u_3-u_b that contains this triphone is calculated. It is to be noted that this process is also useful in systems that utilize half-phonemes, as long as "phoneme" spacing is maintained in creating each triphone cost that is calculated. Using the above example, one sequence would be k_1-$æ_1$-t_1 and another would be k_2-$æ_2$-t_2. This unit spacing is used to avoid including redundant information in the cost functions (since the identity of one of the adjacent half-phones is already a known quantity). In accordance with the present invention, the costs for all sequences u_a-k_1-$æ_1$-t_1-u_b are calculated, where u_a and u_b are allowed to vary over the entire phoneme set. Similarly, the costs for all sequences u_a-k_2-$æ_2$-t_2-u_b are calculated, and so on for each possible triphone sequence. The purpose of calculating the costs offline is solely to determine which units can potentially play a role in the subsequent synthesis, and which can be safely ignored. It is to be noted that the specific relevant costs are re-calculated at synthesis time. This re-calculation is necessary, since a component of the cost is dependent on knowledge of the particular synthesis specification, available only at run time.

Formally, for each individual phoneme to be synthesized, a determination is first made to find a particular triphone context that is of interest. Following that, a determination is made with respect to which acoustic units are either within or outside of the acceptable cost limit for that triphone context. The union of all chosen 5-phone sequences is then performed and associated with the triphone to be synthesized. That is:

$$PreselectSet(u_1, u_2, u_3) = \bigcup_{a \epsilon PH} \bigcup_{b \epsilon PH} CC_n(u_a, u_1, u_2, u_3, u_b)$$

where CC_n is a function for calculating the set of units with the lowest n context costs and CC_n is a function which calculated the n-best matching units in the database for the given context. PH is defined as the set of unit types. The value of "n" refers to the minimum number of candidates that are needed for any given sequence of the form u_a-u_1-u_2-u_3-u_b.

FIG. **5** shows, in simplified form, a flowchart illustrating the process used to populate the triphone cost database used in the system of the present invention. The process is initiated at block **500** and selects a first triphone u_1-u_2-u_3 (block **502**) for which preselection costs will be calculated. The process then proceeds to block **504** which selects a first pair of phonemes to be to the "left" u_a, and "right" u_b phonemes of the previously selected triphone. The concatenation costs associated with this 5-phone grouping are calculated (block **506**) and stored in a database with this particular triphone identity (block **508**). The preselection costs for this particular triphone are calculated by varying phonemes u_a and u_b over the complete set of phonemes (block **510**). Thus, a preselection cost will be calculated for the selected triphone in a 5-phoneme context. Once all possible 5-phoneme combinations of a selected triphone have been evaluated and a cost determined, the "best" are retained, with the proviso that for any arbitrary 5-phoneme context, the set is guaranteed to contain the top N units. The "best" units are defined as exhibiting the lowest target cost (block **512**). In an exemplary embodiment, N=50. Once the "top 50" choices for a selected triphone have been stored in the triphone database, a check is made (block **514**) to see if all possible triphone combinations have been evaluated. If so, the process stops and the triphone database is defined as completed. Otherwise, the process returns to step **502** and selects another triphone for evaluation, using the same method. The process will continue until all possible triphone combinations have been reviewed and the costs calculated. It is an advantage of the present invention that this process is performed only once, prior to "run time", so that during the actual synthesis process (as illustrated in FIG. **6**), the unit selection process uses this created triphone database.

FIG. **6** is a flowchart of an exemplary speech synthesis system. At its initiation (block **600**), a first step is to receive the input text (block **610**) and apply it (block **620**) as an input to text normalization device **202** (as shown in FIG. **2**). The normalized text is then syntactically parsed (block **630**) so that the syntactic structure of each constituent phrase or

word is identified as, for example, a noun, verb, adjective, etc. The syntactically parsed text is then converted to a phoneme-based representation, (block **640**), where these phonemes are then applied as inputs to a unit (phoneme) selection module, such as unit selection device **210** discussed in detail above in association with FIG. **2**. A preselection triphone database **214**, such as that generated by following the steps as outlined in FIG. **5** is added to the configuration. Where a match is found with a triphone key in the database, the prior art process of assessing every possible candidate of a particular unit (phoneme) type is replaced by the inventive process of assessing the shorter, precalculated list related to the triphone key. A candidate list of each requested unit is generated and a Viterbi search is performed (block **650**) to find the lowest cost path through the selected phonemes. The selected phonemes may be then be further processed (block **660**) to form the actual speech output.

What is claimed is:

1. A method of synthesizing speech from an input text using phonemes, the method comprising the steps:

a) creating a triphone preselection cost database including a plurality of all likely triphone combinations and generating a key to index each triphone in the database, wherein creating the triphone preselection cost database further comprises:

1) selecting a predetermined triphone sequence u_1-u_2-u_3; and

2) calculating a preselection cost for each 5-phoneme sequence u_a-u_1-u_2-u_3-u_b, where u_2 is allowed to match any identically labeled phoneme in the database and the units u_a and u_b vary over the entire phoneme universe;

b) retrieving a portion of the input text for synthesis as a phoneme sequence;

c) comparing a retrieved phoneme, in context with its neighboring phonemes, with a plurality of N least cost triphone keys stored in the triphone preselection cost database;

d) choosing, as candidates for synthesis, a list of units from the triphone preselection cost database that comprise a matching triphone key;

e) repeating steps b) through d) for each phoneme in the input text;

f) selecting the least cost path through the network of candidates;

g) processing the phonemes selected in step f) into synthesized speech; and

h) outputting the synthesized speech to an output device.

2. The method as defined in claim **1** wherein in performing step a2), the preselection cost is the target cost or an element of the target cost.

3. The method as defined in claim **1**, wherein creating a triphone preselection cost database further comprises:

3) determining a plurality of N least cost database units for the particular 5-phoneme context;

4) performing the union of the N least cost units for all combinations of u_a and u_b;

5) storing the union created in step 4) in a triphone preselection cost database; and

6) repeating steps 1)–5) for each possible triphone sequence.

4. The method as defined in claim **3**, wherein in performing step a4), N=50.

5. A method of creating a preselection cost database of triphones to be used in speech synthesis, the method comprising the steps of:

a) selecting a predetermined triphone sequence u_1-u_2-u_3;

b) calculating a preselection cost for each 5-phoneme sequence u_a-u_1-u_2-u_3-u_b, where u_2 is allowed to match any identically labeled phoneme in the database and the units u_a and u_b vary over the entire phoneme universe;

c) determining a plurality of N least cost database units for the particular 5-phoneme context;

d) performing the union of the plurality of N least cost database units determined in step c);

e) storing the union created in step d) in a triphone preselection cost database; and

f) repeating steps a)–e) for each possible triphone sequence.

6. The method as defined in claim **5** wherein in performing step d), a plurality of fifty least cost sequences and associated costs are stored.

7. The method as defined in claim **5** wherein in performing step b), the preselection cost is the target cost or an element of the target cost.

* * * * *