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

(11) EP 1 638 080 A2

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
22.03.2006 Bulletin 2006/12

(51) Int Cl.:
G10L 13/08 *(2006.01)*

(21) Application number: **05107389.8**

(22) Date of filing: **11.08.2005**

(84) Designated Contracting States:
AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HU IE IS IT LI LT LU LV MC NL PL PT RO SE SI SK TR
Designated Extension States:
AL BA HR MK YU

(30) Priority: **11.08.2004 EP 04300531**

(71) Applicant: **International Business Machines Corporation**
Armonk, N.Y. 10504 (US)

(72) Inventors:
- **Amato, Christel**
  78550 Bazainville (FR)
- **Crepy, Hubert**
  92100 Boulogne (FR)
- **Revelin, Stephane**
  94160 Saint Mande (FR)
- **Waast-Richard, Claire**
  78140 Velizy-Villacoublay (FR)

(74) Representative: **de Pena, Alain et al**
IBM France
Intellectual Property Dept.
Le Plan du bois
06610 La Gaude (FR)

(54) A text-to-speech system and method

(57) A system and method for generating synthetic speech is disclosed. The invention operates in a computer implemented Text-To-Speech system comprising at least a speaker database that has been previously created from user recordings, a Front-End system to receive an input text and a Text-To-Speech engine. Particularly, the Front-End system generates multiple phonetic transcriptions for each word of the input text, and the TTS engine is using a cost function to select which phonetic transcription is the more appropriate for searching the speech segments within the speaker database to be concatenated and synthesized.

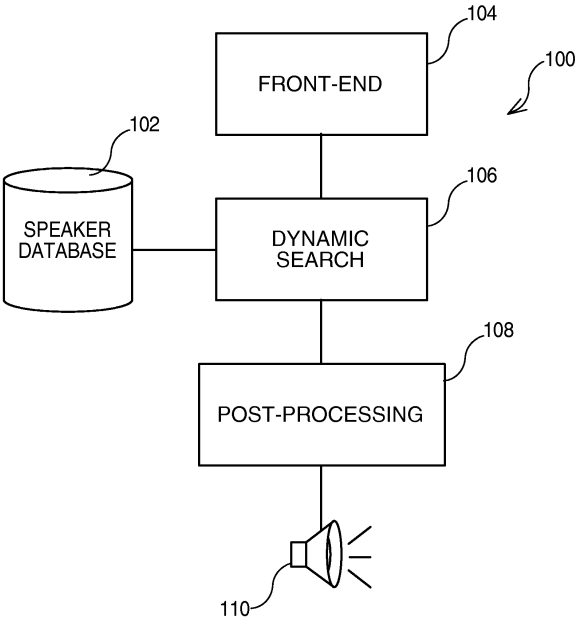

FIG. 1

EP 1 638 080 A2

Description

Technical field

[0001] The present invention relates generally to Text-To-Speech system and method, and more particularly to such system and method based on concatenative technology.

Background

[0002] Text-To-Speech (TTS) systems generate synthetic speech simulating natural speech from an input text. TTS systems based on concatenative technology usually comprise three components: a Speaker Database, a TTS Engine and a Front-End.

[0003] The Speaker Database is firstly created by recording a large number of sentences that are uttered by a speaker, the speaker utterances. Those utterances are transcribed into elementary phonetic units that are extracted from the recordings as speech samples (or segments) that constitute the speaker database of speech segments. It is to be appreciated that each database created is speaker-specific.

[0004] The Front-End that is generally based on linguistic rules and is the first component used at runtime. It takes an input text and normalizes it to generate through a phonetizer one phonetic transcription for each word of the input text. It is to be appreciated that the Front-End is speaker independent.

[0005] The TTS engine then selects for the complete phonetic transcription of the input text the appropriate speech segments from a speaker database and concatenates them to generate synthetic speech. The TTS engine may use any of the available speaker databases (or voices), but only using one at a time.

[0006] As mentioned above, the Front-End is speaker independent and generates the same phonetic transcriptions even if databases of speech segments from different speakers (i.e. different "voices") are being used. But in reality, speakers (even professional ones) do differ in their way of speaking and pronouncing words, at least because of dialectal or speaking style variations. For example, the word "tomato" may be pronounced [tom ah toe] or [tom hey toe].

[0007] Current Front-End systems predict phonetic forms using speaker-independent statistical models or rules. Ideally, the phonetic forms output by the Front-End should match the speaker's pronunciation style. Otherwise, the target phonetic forms prescribed by the Front-End don't find good matches in the speaker database and this is resulting in a degraded output signal.

[0008] In the case of a rule-based Front-End, the rules are in most cases created by expert linguists. For speaker adaptation, each time a new voice (i.e. a TTS system with a new speaker database) is created, the expert would have to adapt manually the rules to the speaker's speaking style. This may be very time consuming.

[0009] In the case of a statistical Front-End, a new one dedicated to the speaker must be trained, which is time consuming too.

[0010] Thus, the current speaker-independent Front-End systems force pronunciations which are not necessarily natural for the recorded speakers. Such mismatches have a very negative impact on the final signal quality, by causing a lot of concatenations and signal processing adjustments.

[0011] Thus it would be desirable to have a Text-To-Speech system that do not impact the quality of the final signal due to mismatches between the Front-End phonetic transcriptions and the recorded speech segments. The present invention offers such solution.

Summary of the invention

[0012] Accordingly, the main object of the invention is to provide a Text-To-Speech system and to achieve a method which highly improves the quality of the synthesized speech generated, by reducing the number of artefacts between speech segments, thereby saving a lot of time processing.

[0013] To summarize, when a sequence of phones is prescribed by the Front-End, there are different sequences of speech segments that can be used to synthesize this phonetic sequence, i.e. several hypotheses. The TTS engine selects the appropriate segments by operating a dynamic programming algorithm which scores each hypothesis with a cost function based on several criteria. The sequence of segments which gets the lowest cost is then selected. When the phonetic transcription provided by the Front-End to the TTS engine at runtime matches well the recorded speaker's pronunciation style, it is easier for the engine to find a matching segment sequence in the speaker database. There is less signal processing required to smoothly splice the segments together. In this setup, the search algorithm evaluates several possibilities of phonetic transcription for each word instead of only one, then computes the best cost for each possibility. In the end, the chosen phonetic transcription will be the one which yields the lowest concatenative cost. For example, the Front-End may phonetize "tomato" into the two possibilities [tom ah toe] or [tom hey toe]. The one that matches the recorded speaker's speaking style is likely to bear a lower concatenation cost, and will therefore be chosen by the engine for synthesis.

[0014] In a preferred embodiment, the invention operates in a computer implemented Text-To-Speech system comprising at least a speaker database that has been previously created from user recordings, a Front-End system to receive an input text and a Text-To-Speech engine. Particularly, the Front-End system generates multiple phonetic transcriptions for each word of the input text, and the TTS engine is using a cost function to select which phonetic transcription is the more appropriate for searching the speech segments within the speaker database to be concatenated and synthesized.

[0015] More generally, a computer system for generating synthetic speech comprises:

(a)a speaker database to store speech segments;
(b)a front-end interface to receive an input text made of a plurality of words;
(c)an output interface to audibly output the synthetic speech; and
(d)computer readable program means executable by the computer for performing actions, including:

(i)creating a plurality of phonetic transcriptions for each word the input text;
(ii)computing a cost score for each phonetic transcription by operating a cost function on the plurality of speech segments; and
(iii)sorting the plurality of phonetic transcriptions according to the computed cost scores.

[0016] In a commercial form, the computer readable program means is embodied on a program storage device that is readable by a computer machine.

[0017] Another object of the invention is to provide a method as defined in the method claims.

Brief description of the drawings

[0018] The above and other objects, features and advantages of the invention will be better understood by reading the following more particular description of the invention in conjunction with the accompanying drawings wherein :

- Figure 1 is a general view of the system of the present invention;
- Figure 2 is a flow chart of the main steps to generate a synthetic speech as defined by the present invention;
- Figure 3 shows an illustrative curve of the cost function;
- Figures 4-a and 4-b exemplify the preferred segments selection in a first-pass approach;
- Figure 5 exemplifies the preferred segments selection in a one-pass approach.

Detailed description of the invention

[0019] A Text-To-Speech (TTS) system according to the invention is illustrated in Figure 1. The general system 100 comprises a speaker database 102 to contain speaker recordings and a Front-End block 104 to receive an input text. A cost computational block 106 is coupled to the speaker database and to the Front-End block to operate a cost function algorithm. A post-processing block 108 is coupled to the cost computational block to concatene the results issued from the cost computational block. The post-processing block is coupled to an output block 110 to produce a synthetic speech.

The TTS system preferably used by the present invention is a concatenative technology based one. It requires a speaker database built from the recordings of one speaker. However, without limitation of the invention, several speakers can record sentences to create several speaker databases. In application, for each TTS system, the speaker database will be different but the TTS engine and the Front-End engine will be the same.

However, different speakers may pronounce a given word in different ways, even in a specific context. In the following two examples, the word "tomato" may be pronounced [tom ah toe] or [tom hey toe] and the French word "fenêtre" may be pronounced [f e n è t r e] or [f e n è t r] or [f n è t r]. If the Front-End predicts the pronunciation [f e n è t r] while the recorded speaker has always pronounced [f n è t r], then it will be difficult to find the missing [e] in this context for this word in the speaker database. On the other hand, if the speaker has used both pronunciations, it could be useful to choose one or the other depending on the others constraints which can be different from one sentence to another. Then, the Front-End provides multiple phonetic transcriptions for each word of the input text and the TTS engine will choose the preferred one when searching the speech segments recorded in order to achieve the best possible quality of the synthetic speech.

[0020] As already mentioned, the speaker database used in the TTS system of the invention is built in a usual way from a speaker recording a plurality of the sentences. The sentences are processed to associate to each of the recorded word an appropriate phonetic transcription. Based on the speaker speaking style, the phonetic transcriptions may differ for each occurrence of the same word. Once the phonetic transcription of every recorded word is done, each audio file is divided into units (so called speech samples or segments) according to these phonetic transcriptions. And the speech segments are classified according to several parameters like the phonetic context, the pitch, the duration or the energy. This classification constitutes the speaker database from which the speech segments will be extracted by the cost computational block 106 during runtime as it will be explained later and then will be concatenated within the post-processing block 108 to finally produce synthetic speech within the output block 110.

[0021] Referring now to figure 2, the main steps of the overall process 200 to issue an improved synthetic speech as defined by the present invention is described.

[0022] The process starts on steps 202 with the reception of an input text within the Front-End block. The input text may be in the form of a user typing a text or of any application transmitting a user request.

[0023] On step 204, the input text is normalized in an usual way well known by those skilled in the art.

[0024] On next step 206, several phonetic transcriptions are generated for each word of the normalized text. It is to be appreciated that the way the Front-End generates multiple phonetic forms is not critical as long as all

the alternate forms are correct for the given sentence. Thus a statistical or rule-based Front-End may be indifferently used or any Front-End based on any other methods. The person skilled in the art would find complete information on statistical Front-End system in « Optimisation d'arbres de décision pour la conversion graphèmes-phonèmes », H.Crépy, C.Amato-Beaujard, J.C. Marcadet and C. Waast-Richard, Proc. of XXIVèmes Journées d'Étude sur la Parole, Nancy, 2002 and more complete information on rule-based Front-End systems in « Self-learning techniques for Grapheme-to-Phoneme conversion », F.Yvon, Proc. of the 2nd Onomastica Research Colloquim, 1994.

Whatever the Front-End system used, it has to disambiguate non-homophonic homographs by itself (e.g. "record" [r ey k o r d] and "record" [r e k o r d]) and it has to propose phonetic forms that are valid for the word usage in the sentence.

To illustrate this on the previous example of the word "fenêtre" which can be pronounced [f e n è t r e], [f e n è t r] or [f n è t r], depending on speaking style, the chosen Front-End block may generate these three phonetic forms.

By contrast, the French word "président" has two possible pronunciations depending on its grammatical class: [p r é z i d an] if it is a noun or [p r é z i d] if it is a verb. The choice of one or the other is totally depending on the sentence syntax. In this case the Front-End must not generate multiple phonetic transcription for the word "président".

[0025] On step 208, the Front-End produces a prediction of the overall pitch contour of the input text (and so produces incidentally the pitch values), the duration and the energy of the speech segments, the well-known prosody parameter. Doing so, the Front-End defines targeted features that will be then used by the search algorithm on next step 210.

[0026] Step 210 allows to operate a cost function for each phonetic transcription provided by the Front-End. A speech segment extraction is made, and given a current segment, this search algorithm aims at finding the next best segments among those available, to be concatenated to the current one. This quest takes into account the features of each segment and the targeted features provided by the Front-End. The search routine allows to evaluate several paths in parallel as it is illustrated in figure 3.

For each unit selection as pointed by a different letter in the example of figure 3, several segments are costed and selected given the previous selected candidates (if any). For each segment a concatenated cost is computed by the cost function and the ones that have the lowest costs are added to a grid of candidate segments. The cost function is based on several criteria which are tunable, (e.g. they can be weighted differently). For instance, if phonetic duration is deemed very important, a high weight to this criterion will penalize the choice of segments which have duration very different from the targeted duration.

[0027] Next, on step 212, the best/preferred path is selected, which is in the preferred embodiment the one that yields the overall lowest cost. The segments aligned to this path are then kept. Once the algorithm has found the best path among the several possibilities, all selected speech samples are concatenated on step 214 using standard signal processing techniques to finally produce synthetic speech on step 216. The best possible quality of the synthetic speech is achieved when the search algorithm successfully limits the amount of signal processing applied to the speech samples. If the phonetic transcriptions used to synthesize a sentence are the same as those that were actually used by the speaker during recordings, the dynamic programming search algorithm will likely find segments in similar contexts and ideally contiguous in the speaker database. When two segments are contiguous in the database, they can be concatenated smoothly as almost no signal processing is involved in joining them. Avoiding or limiting the degradation introduced by signal processing, leads to better signal quality of the synthesized speech. Providing several alternate candidate phonetic transcriptions to the search algorithm increases the chances of selecting best-matching speaker's segments, since those will exhibit lower concatenation costs.

To read more details on the concatenation and production of synthetic speech, the person skilled in the art would refer to «Current status of the IBM Trainable Speech Synthesis System», R. Donovan, A. Ittycheriah, M. Franz, B. Ramabhadran, E. Eide, M. Viswanathan, R. Bakis, W. Hamza, M. Picheny, P. Gleason, T. Rutherfoord, P. Cox, D. Green, E. Janke, S. Revelin, C. Waast, B. Zeller, C. Guenther, and S. Kunzmann, Proc. of the 4th ISCA Tutorial and Research Workshop on Speech Synthesis, Edinburgh, Scotland, 2001 and to «Recent improvements to the IBM Trainable Speech Synthesis System», E. Eide, A. Aaron, R. Bakis, P. Cohen, R. Donovan, W. Hamza, T. Mathes, J. Ordinas, M. Polkosky, M. Picheny, M. Smith, and M. Viswanathan, Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing, Hong Kong, 2003.Front-End.

[0028] It is to be noted that two methods of selecting the most appropriate phonetic transcriptions may be used: a first pass method or a one-pass selection method, now detailed.

[0029] The first pass method consists of running the search algorithm in a first pass only to perform the phonetic transcription selection. The principle is to favor the phonetic criterion in the cost function, e.g. by setting a zero (or extremely small) weight to the others criteria in order to emphasize the phonetic constraints. This method maximizes the chances of choosing a phonetic form identical or very close to the ones used by the speaker during recordings. For each phonetic form provided by the Front-End for a word, different paths are evaluated as shown on figure 3-a. The best paths of all the phonetic forms are compared and the very best one is the phonetic

transcription retained for the further speech segments selection (step 212). Once the phonetic transcription is chosen, the TTS engine goes on in a second pass with the usual speech segments search given the result of this first pass as shown on figure 3-b.

[0030] The second approach 'the one pass selection' allows to select the appropriate phonetic form amongst multiple phonetic transcriptions by introducing them into the usual search step. The principle is mainly the same as the previous method except that only one search pass is done and no parameters of the cost function are strongly favored. All parameters of the cost function are tuned to reach the best tradeoff in the segments choice between the phonetic forms and the other constraints. If a speaker has pronounced a word in different manner during recordings, the choice of the best suitable phonetic transcription may be helped by the others constraints like the pitch, duration, type of the sentence. This is illustrated on figure 4. For instance, here are two French sentences with the same word 'fenêtre' pronounced differently:

(1) La fenêtre est ouverte.
with the word 'fenêtre' pronounced [f e n è t r] , and
(2) Ferme la fenêtre !
with the word 'fenêtre' pronounced [f n è t r].

[0031] The first sentence is affirmative while the second one is exclamatory. These sentences differ in pitch contour, duration and energy. During synthesis this information may help to select the appropriate phonetic form because it will be easier for the search algorithm to find speech segments close to the predicted pitch, duration and energy in sentences of a matching type, for example.

[0032] In this implementation, the phonetic transcription selection is done at the same time as the speech units selection. Then the segments are concatenated to produce the synthesized speech.

Claims

1. A Text-To-Speech system comprising:

means (102) for storing a plurality of speech segments;
means (104) for creating a plurality of phonetic transcriptions for each word of an input text;
means (106) coupled to the storing means and to the creating means for selecting preferred phonetic transcriptions by operating a cost function on the plurality of speech segments.

2. The system of claim 1 wherein the means for selecting preferred phonetic transcriptions comprises means for computing a cost score for each phonetic transcription of the plurality of phonetic transcriptions and means for sorting the plurality of phonetic transcriptions according to the computed cost scores.

3. The system of claim 1 or 2 wherein the means for creating a plurality of phonetic transcriptions comprises rule-based means.

4. The system of claim 1 or 2 wherein the means for creating a plurality of phonetic transcriptions comprises statistical means.

5. The system of anyone of claims 1 to 4 wherein the means for creating a plurality of phonetic transcriptions further comprises means to normalize the input text.

6. The system of anyone of claims 1 to 5 wherein the means for creating a plurality of phonetic transcriptions further comprises means to generate prosody parameters.

7. The system of claim 6 wherein the prosody parameters are input to the means for selecting the preferred phonetic transcriptions.

8. The system of anyone of claims 1 to 7 wherein the means for selecting the preferred phonetic transcriptions further comprises means for selecting preferred speech segments associated to the preferred phonetic transcriptions.

9. The system of claim 8 further comprising concatenation means to concatene the preferred speech segments.

10. The system of claim 9 further comprising means coupled to the concatenation means to output synthetic speech from the concatenated speech segments.

11. In a Text-To-Speech system, a method for selecting preferred phonetic transcriptions of an input text comprising the steps of:

storing a plurality of speech segments;
creating a plurality of phonetic transcriptions for each word of an input text;
computing a cost score for each phonetic transcription by operating a cost function on the plurality of speech segments; and
sorting the plurality of phonetic transcriptions according to the computed cost scores.

12. The method of claim 11 further comprising the step of normalizing the input text before creating the plurality of phonetic transcriptions.

13. The method of claim 11 or 12 further comprising the step of generating prosody parameters after the step of creating a plurality of phonetic transcriptions.

14. The method of anyone of claims 11 to 13 further comprising the step of selecting preferred speech segments after the step of sorting the plurality of phonetic transcriptions.

15. The method of claim 14 further comprising the step of concatenating the preferred speech segments.

16. The method of claim 15 further comprising the step of outputting synthetic speech after the concatenating step.

17. The system of anyone of claims 1 to 10 wherein the means for storing a plurality of speech segments, the means for creating a plurality of phonetic transcriptions for each word of an input text and the means for selecting preferred phonetic transcriptions are computer readable program means executable by a computer machine.

18. A program storage device readable by a computer machine, tangibly embodying the computer readable program means of claim 17.

19. A computer system for generating synthetic speech comprising:

(a)a speaker database to store speech segments;
(b)a front-end interface to receive an input text made of a plurality of words;
(c)an output interface to audibly output the synthetic speech; and
(d)computer readable program means executable by the computer for performing actions, including:

(i)creating a plurality of phonetic transcriptions for each word the input text;
(ii)computing a cost score for each phonetic transcription by operating a cost function on the plurality of speech segments; and
(iii)sorting the plurality of phonetic transcriptions according to the computed cost scores.

20. The system of claim 19 wherein the computer readable program means is embodied on a program storage device readable by a computer machine.

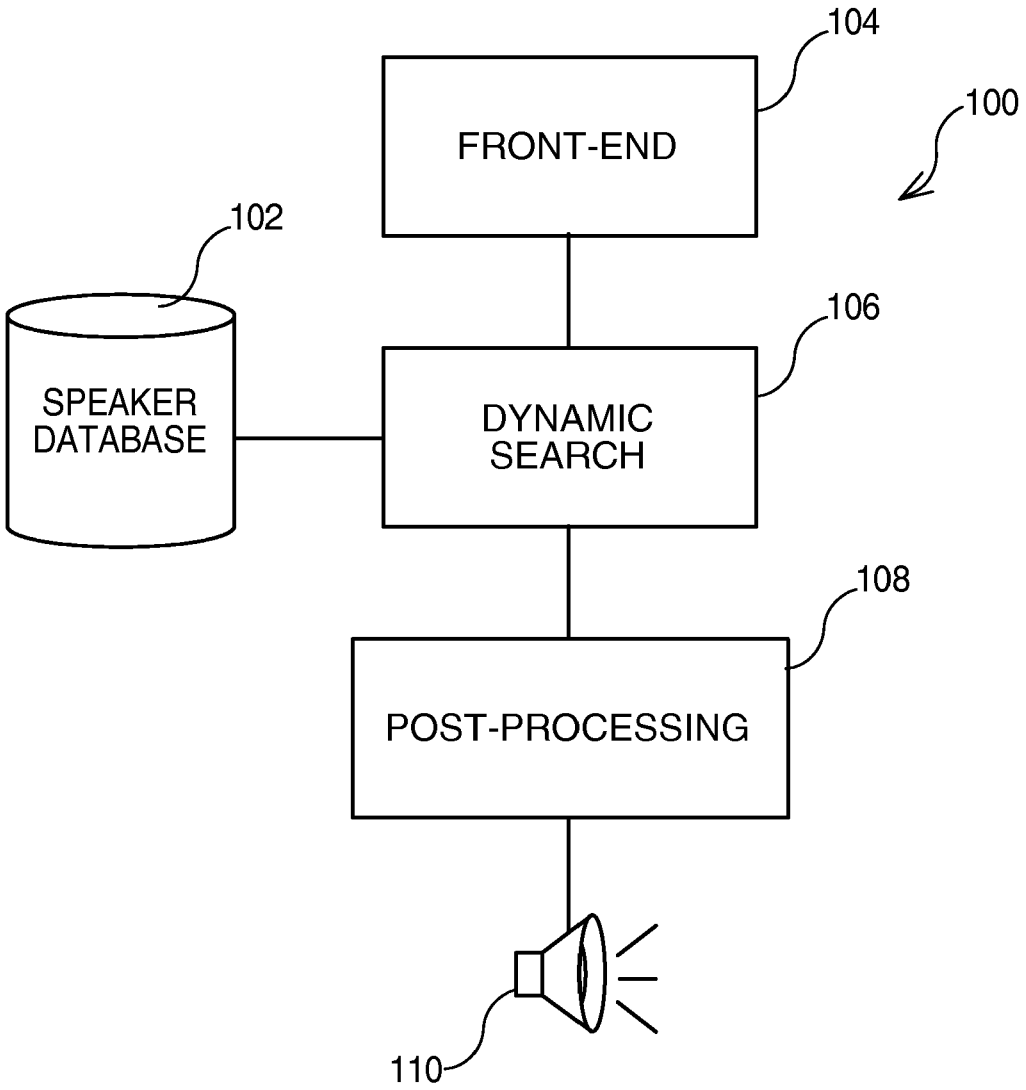
104
100
FRONT-END
102
SPEAKER DATABASE
106
DYNAMIC SEARCH
108
POST-PROCESSING
110

FIG. 1

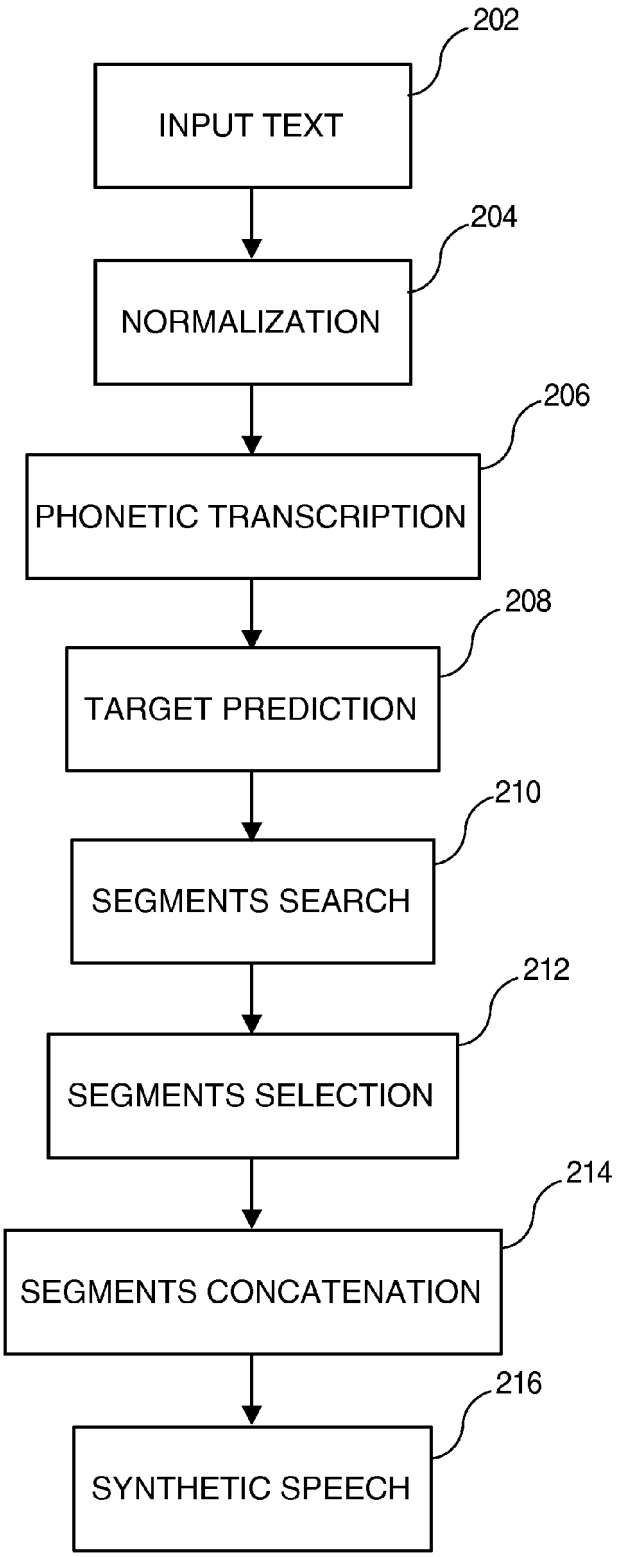
202
INPUT TEXT
204
NORMALIZATION
206
PHONETIC TRANSCRIPTION
208
TARGET PREDICTION
210
SEGMENTS SEARCH
212
SEGMENTS SELECTION
214
SEGMENTS CONCATENATION
216
SYNTHETIC SPEECH

FIG. 2

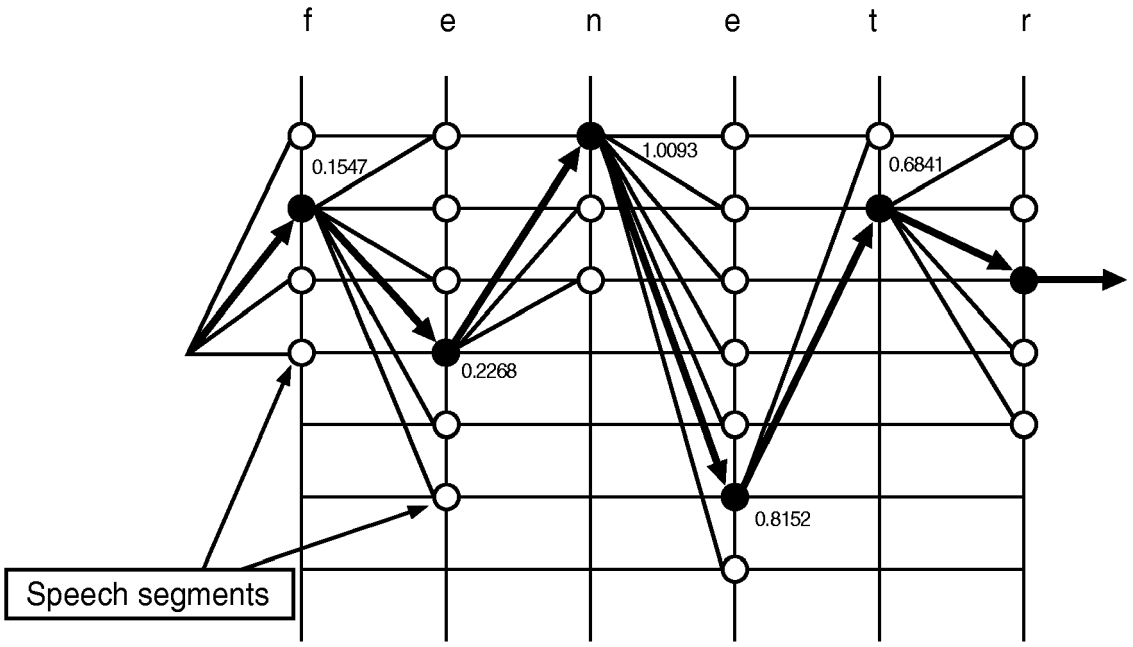

FIG. 3

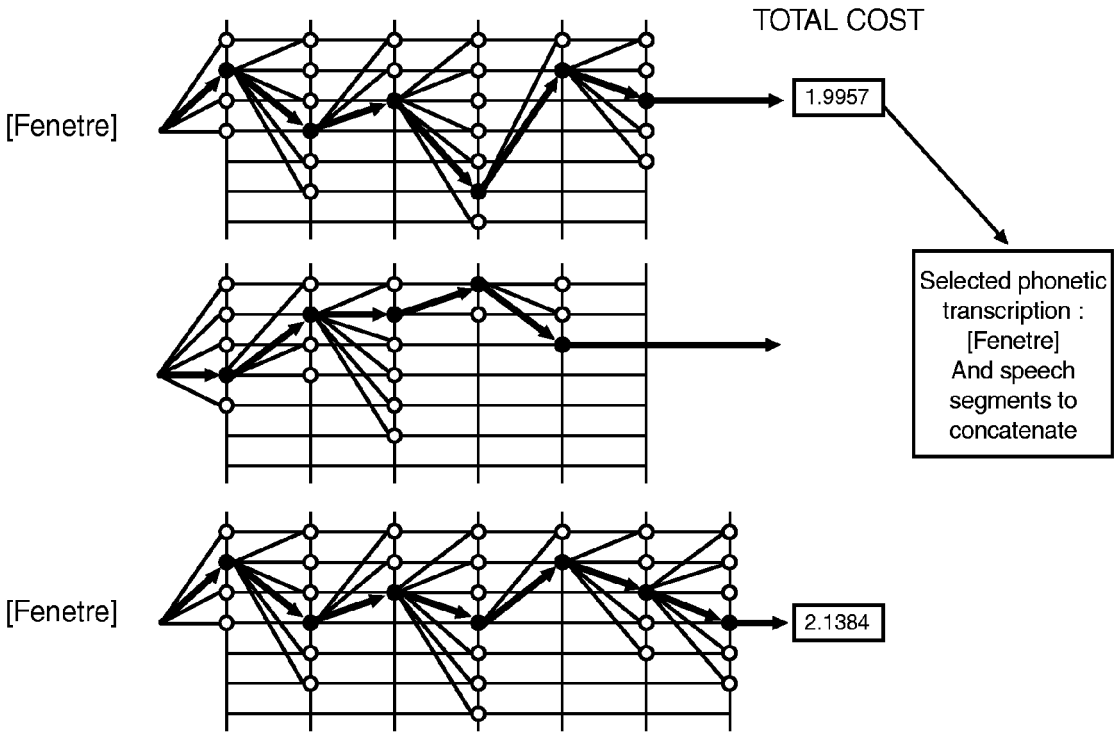
TOTAL COST
[Fenetre]
1.9957
Selected phonetic transcription : [Fenetre] And speech segments to concatenate
[Fenetre]
2.1384

FIG. 5

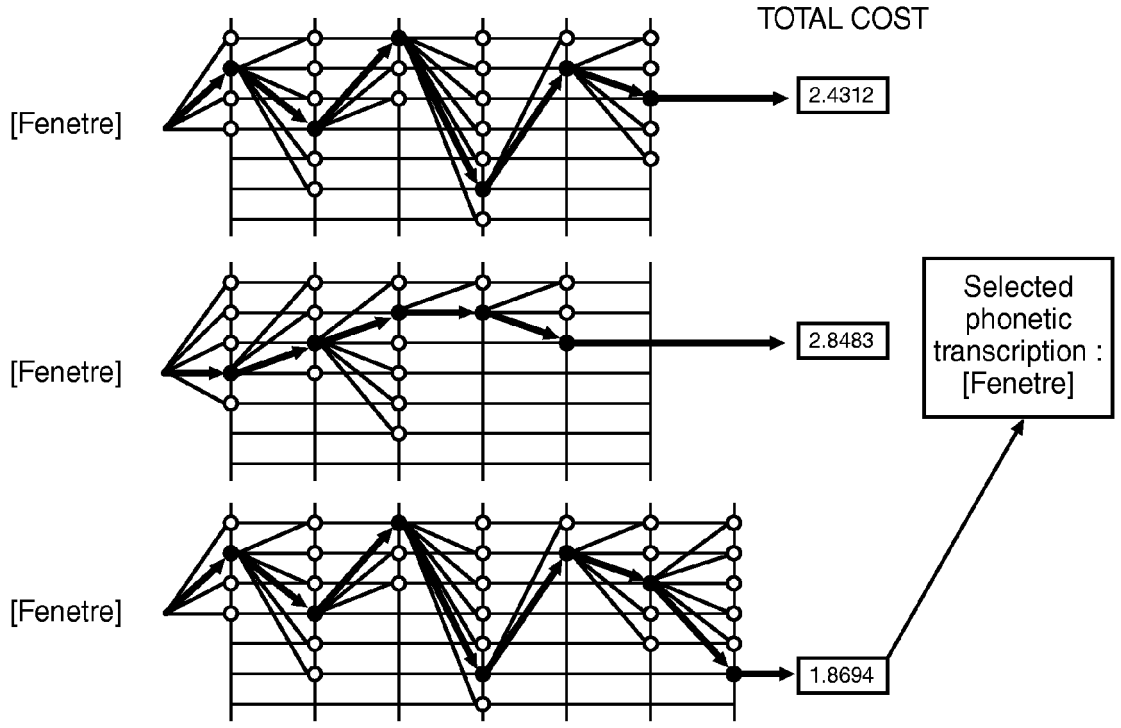
TOTAL COST
[Fenetre]
2.4312
[Fenetre]
2.8483
Selected phonetic transcription : [Fenetre]
[Fenetre]
1.8694

FIG. 4a

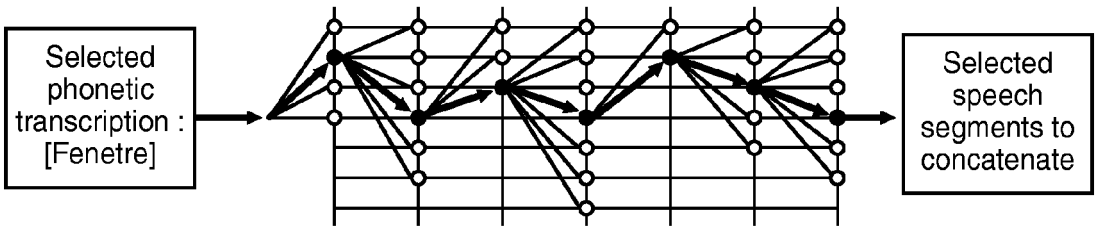
Selected phonetic transcription : [Fenetre]
Selected speech segments to concatenate

FIG. 4b