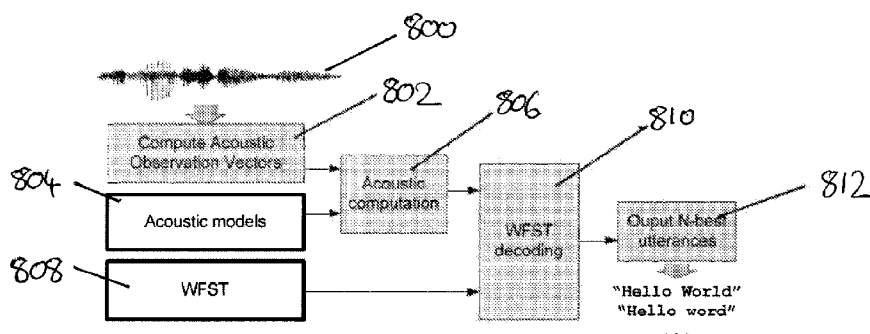




- (51) **International Patent Classification:**
G10L 15/08 (2006.01)
- (21) **International Application Number:**
PCT/GB2011/052436
- (22) **International Filing Date:**
8 December 2011 (08.12.2011)
- (25) **Filing Language:** English
- (26) **Publication Language:** English
- (30) **Priority Data:**
1020771.0 8 December 2010 (08.12.2010) GB
- (71) **Applicant (for all designated States except US):** **THE QUEEN'S UNIVERSITY OF BELFAST** [GB/GB]; University Road Belfast, Belfast Antrim BT7 5NN (GB).
- (72) **Inventors; and**
- (75) **Inventors/Applicants (for US only):** **MCALLISTER, John** [GB/GB]; 13 Sunnyside Drive, Belfast BT7 3DX (GB). **WOODS, Roger** [GB/GB]; 12 Knockracken Manor, Belfast BT8 6WQ (GB). **VEITCH, Richard** [GB/GB]; 81 Lime Grove, Shepherds Bush, London W12 8EE (GB). **MCCOURT, Paul** [GB/GB]; 52 Oakland Avenue, Belfast BT4 3BX (GB). **AUBERT, Louis-Marie** [FR/GB]; 27 High Street, Merstham, Redhill Surrey RH1 3BA (GB). **FISCHABER, Scott** [US/GB]; 40 Sunnyside Street, Belfast BT7 3EX (GB). **WOJCIESZAK, Laurent** [FR/GB]; 52 Greenwood Hill, Belfast BT8 7WF (GB).
- (74) **Agent:** **MCBRIDE, Peter**; Scotland House, 165-169 Scotland Street, Glasgow Strathclyde G5 8PL (GB).
- (81) **Designated States (unless otherwise indicated, for every kind of national protection available):** AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.
- (84) **Designated States (unless otherwise indicated, for every kind of regional protection available):** ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— with international search report (Art. 21(3))

(54) **Title:** PATTERN RECOGNITION**Fig. 8**

(57) **Abstract:** The present disclosure relates to improvements in or relating to pattern recognition, and in particular to a novel systems and methods for pattern recognition, and to devices incorporating such systems and methods. In particular, a spectral analysis is performed on a candidate pattern to obtain a set of observation vectors. The observation vectors are decoded to produce a candidate match between the observation vectors and entries from a knowledge source. The step of decoding the vectors comprises modelling the knowledge source as a weighted finite state transducer (WFST) and the step of decoding comprises modelling the propagation of tokens through nodes of the WFST for observation vectors derived from successive frames of the candidate pattern. For each successive propagation, a list of tokens is sorted in order of their associated cost, and a pruning step is applied to remove tokens from said sorted list which go beyond a pre-determined cost threshold, before performing the next iteration.

PATTERN RECOGNITION

Field of the invention

- 5 The present invention relates to improvements in or relating to pattern recognition, and in particular to a novel systems and methods for pattern recognition, and to devices incorporating such systems and methods.

Background of the invention

10

Pattern recognition is an important field of computer science and engineering. A candidate pattern is compared with a knowledge source, such as entries in a database, and a pattern recognition algorithm is applied to select the entry in the knowledge database that corresponds or that most closely corresponds to the candidate pattern.

15

The present invention has applicability for pattern recognition in many diverse fields, however as an example it is useful to consider one of the possible application areas to which the invention could be applied – that of speech recognition. In that case, the candidate pattern will be a segment of speech, and the knowledge sources will comprise information about phonemes, words and grammar.

20

There has always been an interest to interact more directly with computer technology rather than using keyboards or a mouse, and speech has been viewed as a highly attractive option. Speech recognition software has become available with modern high end PCs and servers; the latest releases achieve good performances with large vocabulary and no longer require initial training by users.

25

30

However, such software solutions are complex and place heavy demands on a computer's CPU and memory, which it turn increases the power

consumption of a system running speech recognition software. For example, the DRAGON NaturallySpeaking speech recognition software available from Nuance Communications, Inc. requires a processor running at a speed higher than 1 GHz and occupies 512MB RAM when running on a Windows® XP
5 operating system, or 1GB when running on a Windows® Vista operating system. This represents a significant proportion of CPU resources, particularly in the home computing environment.

These demands are on a computer's CPU and memory are a deterrent in the
10 uptake of sophisticated speech recognition in consumer desktop environments, but present even more difficulties for the adoption of speech recognition in embedded systems including for example mobile communications devices such as mobile telephones, and other handheld devices. The increasing algorithmic complexity needed to achieve high
15 accuracy even in noisy environments in conjunction with the tight energy constraints of battery operated devices has proved a barrier to implementing sophisticated speech recognition on hand-held devices and other consumer electronic devices where power consumption, processor load, and memory use are key concerns.

As a result, sophisticated speech recognition has to date been limited in uptake, particularly in embedded computing platforms such as mobile telephones, smart phones, netbooks and tablet computers. By way of illustration, it has been shown that the open source speech recognition engine
20 Sphinx II™ runs seven times slower than real-time when compiled on a 206 MHz StrongARM™ with 64 MB of RAM, a typical embedded system configuration.
25

Reliability for speech recognition in a realistic context requires the system to
30 manage a large vocabulary and include noise robustness and speaker adaptation algorithms. Ease-of-use implies a good integration without the need for training, hence a speaker independent system. Also, the speech

recognition system takes natural continuous speech as input, and outputs text or triggers commands in real-time.

Ideally a speech recognition system should deliver an accurate text transcription in real-time, be speaker independent (i.e. no initial training required) and be capable of handling natural continuous speech with a rich vocabulary. At present, the complexity of these tasks exceeds the processing capabilities currently forecast for embedded processor technology.

At present, in order to implement speech recognition in systems with limited resources, such as embedded devices, either word recognition accuracy or system vocabulary have to be sacrificed. These limitations are inherent to current software approaches. Achieving even acceptable levels of performance using a software approach require extremely fast, powerful, large, expensive, and energy-hungry processors.

A more recent approach to speech recognition, seen in the field of mobile telephones, is based on the use of servers to run heavy computations (e.g. Nuance's VSuite™ or Google voice search). Thus a device (such as a mobile telephone) has only simple calculations to carry out before sending acoustic features to servers through a wireless connection. Subsequently, the servers return the transcribed text to the device. However, this network-dependent service relies on the availability of a data connection and creates unnecessary traffic with a potentially long delay, which in turn reduces drastically the range of possible applications.

An object of the present invention is to mitigate or solve some of the problems identified with the prior art and to provide pattern recognition systems and methods that are faster and/or more efficient than existing solutions.

Summary of the invention

According to a first aspect of the present disclosure, there is provided a method of pattern recognition, comprising performing a spectral analysis on a candidate pattern to obtain a set of observation vectors; decoding said observation vectors to produce a candidate match between said observation
5 vectors and entries from a knowledge source; wherein said step of decoding the observation vectors comprises modelling the knowledge source as a weighted finite state transducer (WFST) and the step of decoding comprising modelling the propagation of tokens through nodes of the WFST for observation vectors derived from successive frames of the candidate pattern,
10 and wherein, for each successive propagation, a list of tokens is sorted in order of their associated cost, and a pruning step is applied to remove tokens from said sorted list which go beyond a pre-determined cost threshold, before performing the next iteration.

15 Preferably, the sorting comprises binning a candidate token into one of a set of bins, each of which is associated with a range of cost values; and the pruning step comprises ignoring the tokens in the bin associated with the highest costs.

20 Preferably, the pruning step comprises ignoring the tokens in a plurality of bins, being the set of bins associated with the highest costs.

Preferably, the cost thresholds for each bin scale non-linearly, with lower cost bins being larger than higher cost bins.

25 Preferably, the token propagation is carried out by a dynamic programming algorithm, most preferably using a Viterbi decoding algorithm, which implements the token propagation method.

30 Preferably, transitions between nodes of the WFST correspond to Hidden Markov Model (HMM) states.

Preferably, the propagation of tokens through the WFST comprises, for each transition, the calculation of a cost that includes a contribution from the coefficient of the determined observation vector and a fixed arc cost that represents the cost associated with a transition to a given HMM state.

5

Preferably, a set of said arc costs is stored as part of the knowledge source.

Preferably, said candidate pattern is a speech utterance, and said knowledge source comprises one or more of a phoneme lexicon, a word lexicon and a grammar lexicon.

10

Preferably, the knowledge source comprises all of phoneme lexicon, a word lexicon and a grammar lexicon stored in a single memory structure.

15

Preferably, token propagation is run on different threads in parallel.

According to a second aspect of the present disclosure, there is provided a pattern recognition system, comprising a spectral analyser for analysing a candidate pattern to obtain a set of observation vectors; a decoder for decoding said observation vectors to produce a candidate match between said observation vectors and entries from a knowledge source; and a memory resource having stored thereon a weighted finite state transducer (WFST) that embodies said knowledge source; and a processor arranged, for each successive propagation, to model the propagation of tokens through nodes of the WFST for observation vectors derived from successive frames of the candidate pattern, to sort a list of tokens in order of their associated cost; and to apply a pruning step to remove tokens from said sorted list which go beyond a pre-determined cost threshold.

20

25

30

Preferably, the sorting comprises binning a candidate token into one of a set of bins, each of which is associated with a range of cost values; and the

pruning step comprises ignoring the tokens in the bin associated with the highest costs.

5 Preferably, the pruning step comprises ignoring the tokens in a plurality of bins, being the set of bins associated with the highest costs.

Preferably, the cost thresholds for each bin scale non-linearly, with lower cost bins being larger than higher cost bins.

10 Preferably, the token propagation is carried out by a dynamic programming algorithm, most preferably using a Viterbi decoding algorithm, which implements the token propagation method.

15 Preferably, transitions between nodes of the WFST correspond to Hidden Markov Model (HMM) states.

20 Preferably, the propagation of tokens through the WFST comprises, for each transition, the calculation of a cost that includes a contribution from the coefficient of the determined observation vector and a fixed arc cost that represents the cost associated with a transition to a given HMM state.

Preferably, a set of said arc costs is stored as part of the knowledge source.

25 Preferably, said candidate pattern is a speech utterance, and said knowledge source comprises one or more of a phoneme lexicon, a word lexicon and a grammar lexicon.

30 Preferably, the knowledge source comprises all of phoneme lexicon, a word lexicon and a grammar lexicon stored in a single memory structure.

Preferably, token propagation is run on different threads in parallel.

According to a third aspect of the invention, there is provided an embedded system comprising a pattern recognition system, comprising a spectral analyser for analysing a candidate pattern to obtain a set of observation vectors; a decoder for decoding said observation vectors to produce a candidate match between said observation vectors and entries from a knowledge source; and a memory resource having stored thereon a weighted finite state transducer (WFST) that embodies said knowledge source; and a processor arranged, for each successive propagation, to model the propagation of tokens through nodes of the WFST for observation vectors derived from successive frames of the candidate pattern, to sort a list of tokens in order of their associated cost; and to apply a pruning step to remove tokens from said sorted list which go beyond a pre-determined cost threshold.

According to a fourth aspect of the present invention, there is provided a portable device comprising the embedded system of the third aspect.

The portable device may be a hand held mobile communications device, such as a mobile telephone, smart phone or netbook, for example.

According to a further aspect of the present disclosure, there is provided a computer programme product a WFST and instructions for carrying out the method of the first aspect.

Brief Description of the Figures

Figure 1 shows general principles of pattern recognition;

Figure 2 shows the principles of operation of a decoder which is part of the system of figure 1;

Figure 3 shows the grammatical representation of a phrase;

Figure 4 shows an example representation of various words in a word lexicon that may be used in the decoder of figure 2;

5 Figure 5 shows an example of how phonemes are defined;

Figure 6 shows the grammatical representation of figure 3 with word lexicon data represented;

10 Figure 7 shows the grammatical representation of figure 3 with each of the phoneme representations illustrated;

Figure 8 shows an overview of a speech recognition system in accordance with the present invention; comprising a weighted finite state transducer (WFST);
15

Figure 9 illustrates a WFST graph with HMM states shown as the arcs of the graph;

20 Figure 10 illustrates the on-chip memory loading of WFST portions;

Figure 11 illustrates schematically the general principle of a token propagation algorithm with pruning;

25 Figure 12 illustrates a prior art token propagation method which comprises both beam and histogram pruning;

Figure 13 illustrates a token propagation method according to the disclosure which includes a sorting step;

30

Figure 14 illustrates the general principle of a doubly-linked list for token sorting in beam subdivisions;

Figure 15 illustrates a decision flowchart for a coarse token sorting method, showing the process for deciding whether a token is stored or rejected;

5 Figure 16 shows an example block diagram of architecture for a token propagation algorithm;

Figure 17 shows an example of hardware implementation of a token propagation technique according to the present disclosure;

10

Figure 18 shows a prior art parallel token propagation technique; and

Figure 19 shows a parallel and pipelined token propagation technique according to the disclosure.

15

Detailed Description

20

Figure 1 shows the components of a pattern recognition system according to the invention. In a preferred embodiment, the pattern recognition system is a speech recognition system. A candidate pattern, in this case a speech waveform, 100 is recorded and encoded as a bit-stream, for example in a .wav file format or any other suitable format. The speech wave form 100 is fed to a spectral analyser 102 which decomposes the wave form 100 into a plurality of acoustic vectors 106. Each acoustic vector 106 is sampled over a time-slice 104, referred to as a frame. A typical frame size might, for example be 10 milliseconds. The spectral analyser 102 operates in a known manner using processes such as Fast Fourier Transforms (FFT), discrete cosine transforms (DCT) and other digital filtering techniques to break down each sample into the acoustic vectors 106 which form an input to a decoder 108.

25

30

The decoder 108 performs Gaussian calculations and graph searching (using, for example, Viterbi decoding) to produce a final transcription 110 that corresponds to the interpretation of the speech waveform 100.

- 5 The basic components of the decoder 108 of figure 1 are illustrated in figure 2. The decoder 108 receives the acoustic vectors 106 and parses them using multiple processing layers. The first layer is a phoneme lexicon 200. This comprises a database of available phonemes, being the atomic units that make up speech. The phoneme lexicon 200 may be specific to a given
- 10 language for which the speech recognition system is configured, or it may be contain a set of phonemes related to multiple languages. For the purposes of the present disclosure, phonemes of the English language shall be referred to although it will be understood that the principles of the disclosure may be applied equally to phonemes of other languages, or to other atomic units
- 15 corresponding to the pattern being recognised.

- The phoneme lexicon 200 processes the acoustic vectors 106 and identifies a string 202 of phonemes from within the lexicon which are deemed present in the speech wave form 100. The phonemes thus identified form the input to a
- 20 word lexicon 204. The word lexicon 204 analyses strings of phonemes and parses them to match with a database of words contained within the lexicon. The detected words 206 form the input to a grammar module 208 which checks the order of the words before outputting the final transcription 110.

- 25 The present invention performs pattern recognition using a type of finite state machine known as a weighted finite state transducer (WSFT). A finite state transducer acts to convert input speech into an output, i.e. the final transcription shown in Figures 1 and 2. In a weighted finite state transducer, transitions between states are labelled with a weight, that provides information
- 30 about the inherent likelihood of the transition, so embodying for example knowledge of sentence, word or utterance composition, as will be described below.

A finite state machine (FSM) is a behavioural model of an automated process composed of a finite number of states, transitions between those states, and outputs which are dependant on the state of the machine and optionally the inputs to the system. It has finite internal memory, an input feature that reads symbols in a sequence, one at a time without going backward; and an output feature, which may be in the form of a user interface, once the model is implemented. The operation of an FSM begins from one of the states (called a start state), goes through transitions, depending on input to different states, and can end in any of those available - however only a certain set of states mark a successful flow of operation (called accept states).

The WFST is based on a token propagation model. Components of patterns are modelled as a network of states, with transitions between states having an associated cost, the values of which provide the “weights” of the weighted FST. Each possible sequence of states through the model represents one possible alignment of the model with the candidate pattern. The cost of a sequence of states is derived from the combined cost of each transition along the path of the model. The “cost” is an inversely proportional representation of the likelihood of candidate state transitions or the similarity of candidate pattern components. Progression through state transitions is implemented according to a token propagation algorithm.

The WFST can be modelled by a WFST graph, which shows the states and the transitions between them. Various aspects of the structure of the novel WFST are shown by the WFST graphs of figures 3 to 7. These graphs are for illustration purposes, showing how a speech recognition system according to various embodiments might operate to interpret a spoken command to either “load data” or to “save data”. It will be appreciated that a fully-specified WFST for practical use will be very complex, but that it can be built using the principles set out herein.

Figure 3 shows an example WFST structure embodying grammar knowledge and forming part of the grammar module 208 shown in Figure 2. First of all, the possible commands are stored as a grammar rule as shown by the possible paths in the graph of figure 3. The final command 300 to either load or save data can be thought of as the propagation of a token 302 along a path. The grammar of the phrase is defined by restricting the paths that are possible, so the word “data” in this example can only be preceded by the words “load” or “save”. The grammar rules mean that incompatible strings such as “load load”, or “data save” will not be recognised. Of course, more complex grammar rules can be constructed that will enable similar commands to be phrased in different manners. It will be appreciated that this is a simplified example for the purposes of illustration.

Figure 4 shows an example WFST structure embodying word knowledge and forming part of the word lexicon 204 shown in Figure 2. The graph of Figure 4 illustrates an exemplary word lexicon for the words “load” 400, “save” 402 and “data” 404. Each word 400, 402, 404 is defined as a combination of phonemes 406, which follow in a defined order. The word “data” 404 is defined to take account of two alternative pronunciations of the word “data”, the upper path representing daetah and lower path representing deytah (emphasis added).

Each of the phonemes 406 in figure 4 is in turn represented by a phoneme lexicon 200. Figure 5 shows an example WFST structure embodying phoneme knowledge and forming part of the phoneme lexicon 200 shown in Figure 2. The graph of Figure 5 shows a phoneme for “ah” 500 and “ae” 502. The phonemes for “ah” and “ae” are formed from subcomponents ({ah1, ah2, ah3} and {ae1, ae2, ae3}), each subcomponent corresponding to a Hidden Markov Model (HMM) state. Each state is evaluated against one frame by a Gaussian probability calculation (a measure of the distance from an acoustic model to the speech frame). States ah1, ae1 and ah3, ae3 can be regarded as the start and the end of the corresponding phonemes, respectively. The

middle states ah2, ae2 correspond to the steady state of the phoneme pronunciation. Self loops on each HMM state account for the variable duration of each state (several frames of 10 ms can stay on the same state using the self-loop arcs). Each phoneme in the lexicon may be modelled by a Hidden

5 Markov Model because states between the different components of phonemes cannot be observed. Only the probability of each state being the representation of the speech frame can be evaluated by Gaussian calculations. It is to be appreciated that other models other than HMMs may be used, for example dynamic time warping (DTW) although HMM will have

10 an advantage over DTW for the example of speech recognition because DTW does not handle the stochastic nature of speech.

The new WFST can be formed from a combination of some or all of the knowledge sources (i.e. grammar, word and phoneme structures). Figure 6

15 shows the grammar rule of figure 3 implementing the word lexicon of figure 4, and figure 7 illustrates the same grammar rule with the composition of individual phonemes represented; and Figure 7 shows a WFST that comprises a single network combining all the knowledge sources. As a result, the decoder is agnostic to various knowledge sources. If changes are made

20 to the knowledge sources before the WFST network is compiled, modifications to the decoder are not required. The decoder does not need to know what the WFST graph is made of. The decoding rules are embedded in the graph, not in the decoder (as in traditional approaches). It makes the decoding very regular, faster, because it requires less computation.

Figure 8 shows an overview of a speech recognition system in accordance with an embodiment of the present invention. Speech information 800 (i.e. information representing speech) is input to the system. The speech

25 information 800 may comprise a data file in .wav or any other suitable audio file format. The speech information is broken down into acoustic observation vectors by an acoustic vector computer 802, at a predetermined frame rate, for example 10 ms. The information in each time frame comprises a small

30

portion of the input speech information 800 converted into acoustic observation vectors, the coefficients for which may be derived from techniques such as mel-frequency cepstral coefficients (MFCC) extraction, linear prediction coding (LPC) and its derivatives; or similar.

5

An acoustic models store 804 is used to store information representing acoustic models of the smallest differentiable sounds (called phonemes) produced during speech. Each acoustic model is associated with a tag which identifies the acoustic model and a score, which will be described below in more detail. The acoustic models may for example be Gaussian Mixture Models (GMM) which model speech phonemes as a combination of multi-dimensional Gaussian functions.

10

An acoustic comparator 806 scores each acoustic model based on a comparison with the acoustic observation vectors of the current frame. This comparison may be performed through resolving the mixtures (typically Gaussian) representing the models for the acoustic observation vectors of the current frame.

15

The scored acoustic models are then converted into a number of transcribed utterances using a weighted finite state transducer (WFST) 808 and a WFST decoder 810. The WFST decoder 810 produces an output 812 that comprises a number of candidate transcription matches for the speech information 800. These candidate matches can be ranked in order of probability. In one embodiment, a single transcription match, corresponding to the most probable candidate, is output.

20

25

The WFST 808 comprises a precompiled network of all available knowledge sources stored in a memory structure. The process of compiling and optimizing the different knowledge sources (the multiple WFST graphs representing the phoneme lexicon, the word lexicon and the grammar/language model) into one single static WFST graph can be

30

performed off-line by separate computers. This saves the host system (that is, the system that runs the speech recognition software) the task of on-the-fly composition of knowledge sources to form a dynamic graph based on the input, which can represent up to 50% of computations. Moreover, by enabling
5 network creation separate from the system, an efficient regularised decoding architecture can be generated with real-time fixed processing and minimal delay. This eases the development of the application layer and improves the user experience. This approach also leads to a much simplified algorithm, which gives an advantage in terms of power consumption.

10 Speech recognition systems and the information stored in them require significant memory resources, and so efforts are made to either reduce the size in memory of the speech recognition systems as a whole, or to adjust the underlying architecture of the relevant components. Different approaches to
15 deal with the large size of Wastes have been proposed, including for example, two-pass algorithm and on-the-fly composition.

In two-pass decoding, two WFST graphs are used. The first WFST graph is based on a simplified language model, a pass through which generates
20 several hypotheses. These hypotheses in turn provide the input to a second WFST. This second WFST comprises a grammar module that is compiled from a more complex language model only. Because it does not include neither the phoneme lexicon nor the word lexicon, it is smaller in size and so the second pass is usually much faster than the first.

25 In an on-the-fly composition, instead of combining all knowledge sources into a single WFST graph, the phoneme lexicon can be kept as a separate WFST graph from the main WFST graph comprising the word and grammar knowledge sources. The phoneme lexicon needs to be composed with the
30 word and grammar lexicons on-the-fly during the decoding.

The present disclosure allows for the on-chip storage of portions of the WFST 808, as shown in Figure 10. The WFST 808 may be stored on a different chip than the WFST decoder 810. Required nodes and arcs from the WSFT are first selected and copied (at step 1100) and then stored (at step 1102) at a local memory.

The system of the present disclosure is flexible enough to be adapted to WFSTs of different natures. For example, the core architecture for a decoder in line with the present disclosure can work on WFSTs that are composed on-the-fly by another part of the system.

If an on-the-fly composition algorithm is used, the block in charge of the computation would interface with the system of the present disclosure at this stage. For example, consider a case where the phoneme lexicon is stored separately from another memory resource that stores both the word and grammar lexicons. Block 1100 would select and copy nodes and arcs from both the phoneme lexicon and word and grammar lexicons. An additional block 1104 could be provided that would compose nodes and arcs from the first WFST graph comprising the phoneme lexicon with nodes and arcs from the second WFST graph comprising the word and grammar lexicons to form a local WFST sub-graph that contains nodes where tokens are currently located. This local WFST sub-graph needs to be updated at every frame according to tokens' displacements.

The WFST decoder 810 decodes the input to by searching the WFST 808 to find the best hypotheses according to the input speech. The method used is a time synchronous Viterbi decoding method which is implemented using a token propagation algorithm.

Figure 9 shows an example WFST graph. Each node 900 in the graph (labelled from 0 to 6) represents a WFST state. The HMM states are represented by the arcs (or lines – the term “arc” will be used as a

generalisation) between the nodes. The arcs of the graphs are labelled with a notation identifying “HMM state”:”output”/”cost”, that is, for example, the arc labelled “i3:A/0.5” represents the HMM state “i3”, the output “A”, and a cost of 0.5.

5

For any given HMM state, the graph only allows subsequent transitions to a subset of HMM states. In this illustrative example, there are four available HMM states, i1, i2, i3 and i4. It can be seen from node 0 for example that only i1 or i2 is allowed as an initial HMM state, and then if i1 occurs, it can then only be followed by i3. On the other hand, if i2 occurs, it can be followed either by i3 or i4.

10

The cost represented here (i.e. at the end of the “HMM state”:”output”/”cost” notation) is the fixed cost associated with the transition between the relevant nodes of the WFST, that is, in moving from a given HMM state to another. The particular cost values of the transitions, and the availability of given transitions, effectively embodies the knowledge sources. That is, in the example of a speech recognition, the words and grammar play a part in defining the sequences of HMM states that are allowable, and the relative probabilities of the states where there are a number of allowed transitions.

15

20

Outputs are then associated with various HMM states (or arcs in the graph). Each of the outputs corresponds to a sequence of HMM states, or sometimes a single HMM state. As mentioned above, in one speech recognition embodiment, a phoneme is actually comprised of three HMM states. Therefore, if a sequence of those three HMM states occurs, that phoneme can be presented as an output. In figure 9, “-” represents a null output. This may mean that a particular HMM state is not at the end point of an HMM state sequence that means an output can be uniquely defined, or that there is no sound at that point. Other outputs in the figure are represented by “A”, “B”, and “D”.

25

30

Initially, node 0 will be set to a value of zero while the other nodes will be set to infinity. The values will be updated by the propagation of tokens through the graph. At each time frame, an acoustic observation vector is extracted by the spectral analyser and a Gaussian calculation is carried out to obtain a value for each HMM state, corresponding to the coefficients for each dimension of the vector. The input Gaussian results are then fed to the token propagation algorithm. The Gaussian result forms the basis of a Gaussian cost which is added to the fixed arc cost. The (additive) combination of these two costs gives the total cost for each arc, i.e. for transition to each of the permitted HMM states.

For example, if a first frame of data yields Gaussian results for HMM states i1-i4 of 1.7, 0.8, 1.5 and 0.3 respectively, then the total cost for the transition between nodes zero and one is given by the combined Gaussian result (1.7) plus the fixed arc cost (0.2), giving a total cost of 1.9. That is, the cost of a transition representing a null output is 1.9. Similarly, the cost of a transition representing a "B" output, that is, in the token propagating from node zero to node two, is given by the Gaussian result for i2, 0.8, plus the fixed arc cost, 0.6, so totalling 1.4. So, at that stage, the WFST is telling us that the output of a "B" is more likely than a null output. The costs of each path is represented by the propagation of a token along the graph. The token is a data structure that records the path taken, the cumulative cost of that path, and the previously output values as traceback entries.

At the next frame, a fresh set of Gaussian results is obtained, and the tokens that were propagated to nodes 1 and 2 are propagated further in the graph. The costs associated with each token accumulates, and the new Gaussian portion of the cost is calculated according to the newly derived HMM states. So, for example, if the next frame of data yields Gaussian results for HMM states i1-i4 of 2.4, 1.7, 0.2 and 1.3 respectively, then the cost associated with the propagation from node one to node three is the sum of the previously accumulated cost, the new Gaussian cost, and the fixed arc cost associated

with that HMM state, i.e. in this example, $1.9+0.2$ (the result for HMM state i3) $+0.5 = 2.6$.

In this way, tokens propagate through the graph, and so yield a number of hypotheses of what information the input speech information 800 represents (represented by the different paths through the graph) are created using the WFST decoder 810. Each hypothesis is associated with a token which is used to identify a hypothesis, store the running cost of the hypothesis, and store the location of the hypothesis in the WFST 808. The token with the lowest overall cost represents the match that is most likely to be correct.

As shown in figure 11, at every frame tokens are propagated 1200 along the arcs of the WFST 808. The number of possible hypotheses will grow exponentially as more information is provided to the WFST decoder 810 and pruning 1202 is used to maintain a list of tokens 1204 associated viable hypotheses and discarding the tokens associated with the least probable hypotheses.

Pruning 1202 represents one of the main challenges of any token propagation method; performances in term of recognition accuracy, speed and energy consumption heavily depend on its implementation.

Figure 12 shows a classic token propagation method. As noted above, the hypotheses represent the possible utterances pronounced which may be sequences of words output during the tokens propagation. So each token in the token list 1300 is also linked to a record of the word history associated with that token so that the best possible transcription can be displayed at the end of the utterance. Word histories are stored in the Word Link Record (WLR) memory 1302 and each token is linked to an entry in this memory. An entry in the WLR memory 1302 corresponds to a token's hypothesis and the corresponding utterance to the hypothesis can be recovered by tracing the WLR back.

The token selection technique shown in figure 12 uses a combination of beam pruning 1304 and histogram pruning 1306. Beam pruning 1304 is a breadth-first search process that discards each token having a cost higher than the current best cost plus a fixed beam width. Histogram pruning 1306 limits the number of tokens propagated during the subsequent frame, to a fixed maximum number of tokens. For each frame, a histogram of the token costs is built. The maximum cost of the tokens which are selected to be propagated during the next frame is chosen so that the cumulative number of tokens does not exceed a predefined limit (for example, 4000 tokens). Pruning is then applied on the input tokens. Before being propagated, token costs are compared to the maximum cost determined in the previous step: tokens whose cost are higher than the threshold are pruned (ignored).

The technique shown in figure 12 has the drawback of requiring two passes through all the tokens. Moreover, the fact that more tokens/hypotheses are kept alive than actually propagated in the subsequent frames creates new entries in the WLR memory 1302 that are unusable. This in turns causes the WLR memory 1302 to fill up with unusable entries. As a result, the WLR memory 1302 needs frequent 'garbage collection' 1308 (i.e. the WLR memory 1302 needs to be searched to identify and remove the unusable entries) to reallocate the memory slots of unusable entries to new entries.

Figure 13 shows a token propagation method in accordance with the present disclosure. As for the classical token propagation method, the hypotheses represent the possible utterances pronounced which are sequences of words output during the tokens propagation. So each token in the token list 1400 is also linked to a record of the word history associated with that token so that the best possible transcription can be displayed at the end of the utterance. Word histories are stored in the Word Link Record (WLR) memory 1402 and each token is linked to an entry in this memory. An entry in the WLR memory

1402 corresponds to a token's hypothesis and the corresponding utterance to the hypothesis can be recovered by tracing the WLR back.

However, in the token propagation method of figure 13, pruning is achieved by sorting tokens in the token list in cost order, preferably in increasing cost order. This is a surprising development because it has previously been considered too computationally expensive to successfully implement sorting whilst at the same time achieving an acceptable response time.

The method shown in figure 13 operates as follows. The tokens are sorted 1404 in order of cost (preferably, in order of increasing cost). The tokens having cost over a certain threshold are rejected. The remaining tokens are stored in the token list 1400 in order of cost. These tokens are then propagated 1406 in the order they have been sorted, starting with the token having the lowest cost and ending with the token having the highest cost. As part of propagation, the cost of each token is recalculated. Beam pruning 430 then discards each token having a cost higher than the current best cost plus a fixed beam width. The tokens are then re-sorted 1404 and the process is continued for every frame until the end of the utterance thereby obtaining the most likely results.

The use of a sorting step in the token propagation algorithm marks a clear departure from traditional approaches where sorting is a very expensive task. The sorting is performed on-the-fly and offers several advantages over the conventional beam and histogram pruning (as performed either alone or in combination).

Firstly, the propagation of the tokens having the lowest cost in advance of higher cost tokens guarantees optimum performance and makes histogram pruning straightforward.

Secondly, because best tokens (i.e. the lowest cost tokens) are propagated first, there is a high probability that these tokens will still be the best tokens after they have been propagated and their new cost calculated. Thus, at the next frame there is high probability that the best tokens will be received at an early stage in the propagation of all the tokens stored in the token list 1400. Since the best token is the reference used for determining which tokens are pruned during beam pruning, identifying the best token at an early stage in the propagation improves the performance of the system.

Furthermore, the sorting approach enables the system to selectively store only the most relevant tokens (i.e. those having a cost below a certain threshold). This drastically reduces the number of unusable tokens / hypotheses stored in WLR memory 1402. Thus the frequency that garbage collection 1410 needs to be performed is in turn greatly reduced thereby improving the performance of the system.

At every frame, tokens are propagated, sorted and discarded “live”, which allows the system to be dimensioned to store only the number of tokens that are actually propagated during the next frame; as opposed to the necessity to propagate and store all tokens, and then apply the pruning (which needs a lot more memory).

As noted above, a problem with implementing sorting as part of a token propagation method is that sorting is a computationally expensive operation, and will require a lot of resource from the processor. Despite being capable of more complex tasks, processors have to put on hold the rest of the system to deal with the sorting. Generally processors are designed for sequential tasks and complex arithmetic operations but they handle tasks such as sorting very poorly indeed.

This problem may be overcome or at least partially addressed by using parallel hardware accelerators and distributed memory to effectively deal with

the repetitive task of sorting. This represents a clear departure from existing work as this creates a novel architecture which can be suitable for embedded applications.

- 5 The parallel hardware accelerators are tailored to the task of sorting and achieve it very efficiently in term of speed and energy consumption. They also free up the rest of the system, especially the arithmetic unit, so that the token propagation can continue to run in parallel with the sorting.
- 10 A number of different sorting algorithms can be used with the speech recognition system of the present disclosure. Among those, hardware implementation of two techniques, namely skip lists and parallel sorting by sort-trees, have been found particularly suitable.
- 15 However, the inventors have performed experiments and have verified that a non-exhaustive, or “coarse”, sorting algorithm can offer very similar recognition performances to skip lists and parallel sorting by sort-trees, but with an even more efficient implementation.
- 20 The coarse sorting algorithm operates as follows with reference to figure 14, which shows the general principle of the use of a doubly-linked list for token sorting in beam subdivisions. A singly-linked list could also be used.
- 25 In order to minimize the amount of sorting that needs to be done, tokens are sorted into an ordering list 1500 that is divided up into fractions (for example, this may comprise 64 subdivisions). The beam subdivisions are bins that divide the beam into fractions. Say, if the beam pruning threshold is equal to 160, and we have 4 subdivisions, then: $f = 0$ for costs from 0 to 40; $f = 1$ for costs from 40 to 80; $f = 2$ for costs from 80 to 120; $f = 3$ for costs from 120 to
- 30 160. It is also possible to scale the thresholds for each level to ensure that the lower subdivisions are much larger as there are fewer tokens which will fall into these categories; the idea is to have a relatively even distribution

between subdivisions with the highest threshold being the beam threshold. Such a scaling could be a non-linear scaling, for example a logarithmic scaling. For example, again assuming that a beam pruning threshold is equal to 160, and we have 4 subdivisions, it would be possible to set $f = 0$ for costs from 0 to 70; $f = 1$ for costs from 70 to 110; $f = 2$ for costs from 110 to 140 and $f = 4$ for costs from 140 to 160. This is because far more tokens will have a cost between 140 and 160 than between 0 and 20; and it is desirable that the number of tokens is evenly spread between beam subdivisions.

10 Note that in practice, more subdivisions are used (typically 16 with a 5000-word speech recognition task where we keep up to 4000 tokens alive at every frame; for a 50000-word task, 60000 tokens and 64 subdivisions may be necessary).

15 As there is no way to determine which subdivisions will be filled when, each entry in the doubly linked ordering list 1500 is associated with a pointer to the preceding value in the unordered token list 1502 as well as a pointer to the next value in the doubly linked list 1500. This allows the doubly linked list 1500 to be read from the beginning to the end once propagation is complete
20 and it permits data to be inserted and removed from each beam subdivision independently. A separate entry list (not shown) is maintained which contains a pointer to the last entry for each beam subdivision.

25 A binary search is used to determine which beam subdivision to place a new token. With this method, the number of comparisons made will only be $\log_2$ of the number of beam subdivisions. It is also possible to implement a solution based on arithmetic methods.

30 The next step is to decide whether to store a new hypothesis as a new token, as a replacement of a previously stored token, or drop it completely. This is based on how full the beam subdivisions are and the cost of the new token. Figure 15 shows a flowchart for making this decision.

At step 1600, a particular bin, identified by a bin index f , is determined on the basis of the cost of the token that is input. Each bin corresponds to a predetermined beam subdivision, that is, a range of costs. In a preferred embodiment the bin index, f , increments with the costs of the subdivisions, that is, a low index corresponds to a low cost subdivision.

At step 1602, the total number, N , of sorted tokens is checked against a predetermined maximum number, $N_{\max}$, of tokens that the system (as a whole) can store. If the number of tokens N is less than the maximum $N_{\max}$ then the new token is inserted in the allocated bin at step 1604. If, however, the total number of tokens N is not less than the maximum number of tokens $N_{\max}$ that the system can store, then the bin index, f , is compared at step 1606 with f_{top} , which is the bin index corresponding to the current sub-division whose cost boundaries are highest. If the bin index f of the identified bin is less than f_{top} , then one token is removed from the f_{top} bin (at step 1608) and then a token is inserted into bin f (at 1604). If, however, the identified bin index f is not less than f_{top} , then the token is discarded (at step 1610).

An example of an actual implementation of the token propagation algorithm including the token sorting can be represented by the block diagram of figure 16.

In step 1700, the outgoing arcs from the nodes where tokens are located in the WFST stored in memory 1702 are loaded into local memory 1704. In step 1706 the cost of each token is incremented by the cost of propagation on every corresponding outgoing arc. In step 1708 a beam subdivision, f , is determined for each propagated token based on the updated cost.

In step 1710, a decision is taken as to what is to be done with a particular token – i.e. is it to be inserted in the token list, is to replace an existing token in the token list or is it to be discarded? This decision is based on the

allocated beam subdivision, f , and performed as previously described with reference to figure 15.

If a token is not discarded in step 1710 the symbol associated with the token
5 is copied in step 1712 from memory 1704 into a word link record (WLR)
memory 1714. In addition, in step 1716, the token is stored in the token list
1718 in order of cost as calculated in step 1706. Each token stored in the
token list 1718 is associated with a node of the WFST graph and a traceback
10 entry to the word link record 1714 as previously described with reference to
figure 13.

Note that figure 16 also includes the management of N-best hypotheses. In
step 1720 the N-best hypotheses for a particular node are checked. The list
of nodes for the N-best hypotheses are recorded in memory 1722, each entry
15 in the list of nodes for the N-best hypotheses is associated with a list of the
Nbest tokens located on the particular node. This enables the decoder to
associate up to N hypotheses per node in the WFST graph, each of them
identified by a distinct token. As tokens are propagated through the WFST
network, two or more tokens can reach the same node at the same time
20 through different paths, offering different transcription hypotheses. The
system described in the present disclosure allows the display of these
different transcriptions.

At the start of a new frame, the best tokens (i.e. the tokens having the lowest
25 cost) from the last frame are read in order of cost in step 1730 and used by
step 1700 in order of cost (i.e. lowest cost tokens first) to load outgoing arcs of
the WFST into local memory 1704.

In step 1732 the N-best utterances are output. This is done by tracing back
30 entries in the WLR memory 1714 and the Lexicon word link (WL) memory
1734. These different transcription hypotheses can also be fed to another
layer of the system. The N-best outputs of the speech recognition core

engine can be used at a higher application level, for example, multiple hypotheses displayed to the user; multiple hypotheses used as input to a more complex language model, to name a few uses.

5 It is also worth noting the number and distribution of memory blocks 1702, 1704, 1714, 1718, 1722, 1726, and 1734 which highlights that this architecture is particularly suited for hardware implementation where distributed memories can be used to solve problems related to memory bandwidth and locality. As an illustration, figure 17 depicts the non-trivial
10 implementation on dedicated hardware of the technique outlined.

In a further embodiment, the system of the present disclosure may also provide for parallel token propagation, as shown in figure 18. Running the token propagation on different threads in parallel is an efficient way to
15 accelerate the overall token propagation procedure.

As shown in figure 19, in a pruning approach in accordance with the present disclosure, sorting is scalable and can be distributed over parallel threads without the need for frequent synchronization. Moreover the parallel system
20 does not have to wait that all tokens are sorted. The first tokens with best costs can be propagated without having to wait for the last one to be ready. This enables the system to be implementable with parallel pipelined processes on each thread. Moreover, the need for communication between threads is reduced and as a result the length of each pipeline process can be
25 increased, which reduces the bandwidth and energy used by the system and also increases the speed of the system.

In the system of the present disclosure, the use of threads to implement a pipeline process also offers a quasilinear improvement in performances which
30 increases with the number of threads used. Dedicated hardware can be used to implement the system of the present disclosure for embedded applications

thereby making this performance gain achievable even with a large number of parallel hardware blocks.

This is not achievable in other approaches where tokens are not sorted.

5 Performance evaluation of the parallel implementation using traditional pruning approaches shows that as the number of parallel threads increases performance improvement tend to drop off. This is due to the need for all threads to synchronize and communicate data through shared memory. The requirement of frequent barriers to synchronize threads and the limited
10 bandwidth of the shared memory limit the achievable speed.

Thus the present disclosure allows the development of hardware implementations of speech recognition systems and thereby allows speech recognition to be performed on embedded systems, meaning that
15 implementation on handheld devices or mobile communications devices is possible, or as an FPGA-based implementation. Further, such systems can use a word count that provides utility for functionality in a wide range of applications, for example a word count of around 50,000 words, or more, could be accommodated.

20

Products in the embedded system market which can use the system of the present disclosure includes for example mobile battery operated devices like mobile phones, smartphones, multimedia devices, satellite navigation systems, low cost mini laptop (netbooks), personal digital assistants (PDAs),
25 mp3 players, videogame consoles, digital cameras, DVD players, and printers.

It is to be appreciated that, while there has been some discussion of the applicability of the invention to speech recognition, the principles of the
30 invention could applied to pattern recognition more generally. Other applications include use with advanced HVAC systems which use networked thermostats to more accurately and efficiently control temperature that can

change by time of day and season. Home automation uses wired- and wireless-networking that can be used to control lights, climate, security, audio/visual, surveillance, etc., all of which use embedded devices for sensing and controlling.

5

Further applications include use with transportation systems from flight to automobiles which increasingly use embedded systems. New airplanes contain advanced avionics such as inertial guidance systems and GPS receivers that also have considerable safety requirements. Various electric
10 motors — brushless DC motors, induction motors and DC motors — are using electric/electronic motor controllers. Automobiles, electric vehicles, and hybrid vehicles are increasingly using embedded systems to maximize efficiency and reduce pollution. Other automotive safety systems include anti-lock braking system (ABS), Electronic Stability Control (ESC/ESP), traction control (TCS)
15 and automatic four-wheel drive.

Medical equipment is continuing to advance with more embedded systems for vital signs monitoring, electronic stethoscopes for amplifying sounds, and various medical imaging (PET, SPECT, CT, MRI) for non-invasive internal
20 inspections and can also include the system of the present disclosure.

A further possible application is motion gesture recognition. The person skilled in the art will appreciate that many other applications will be possible that are not listed herein.

25

Various improvements and modifications can be made to be above without departing from the scope of the invention.

CLAIMS

1. A method of pattern recognition, comprising:

performing a spectral analysis on a candidate pattern to obtain a set of
5 observation vectors;

decoding said observation vectors to produce a candidate match
between said observation vectors and entries from a knowledge source;
wherein

said step of decoding the acoustic vectors comprises modelling the
10 knowledge source as a weighted finite state transducer (WFST) and the step
of decoding comprising modelling the propagation of tokens through nodes of
the WFST for observation vectors derived from successive frames of the
candidate pattern, and wherein,

for each successive propagation, a list of tokens is sorted in order of
15 their associated cost, and a pruning step is applied to remove tokens from
said sorted list which go beyond a pre-determined cost threshold, before
performing the next iteration.

2. The method of claim 1, wherein the sorting comprises binning a candidate
20 token into one of a set of bins, each of which is associated with a range of
cost values; and the pruning step comprises ignoring the tokens in the bin
associated with the highest costs.

3. The method of claim 2, wherein the pruning step comprises ignoring the
25 tokens in a plurality of bins, being the set of bins associated with the highest
costs.

4. The method of claim 2 or claim 3, wherein the cost thresholds for each bin
scale non-linearly, with lower cost bins being larger than higher cost bins.

5. The method of any preceding claim, wherein the token propagation is
carried out by a dynamic programming algorithm.

6. The method of claim 5, wherein said dynamic programming algorithm comprises a Viterbi decoding algorithm.

5 7. The method of any preceding claim, wherein transitions between nodes of the WFST correspond to Hidden Markov Model (HMM) states.

8. The method of claim 7, wherein the propagation of tokens through the WFST comprises, for each transition, the calculation of a cost that includes a
10 contribution from the coefficient of the determined observation vector and a fixed arc cost that represents the cost associated with a transition to a given HMM state.

9. The method of claim 8, wherein a set of said arc costs is stored as part of
15 the knowledge source.

10. The method of any preceding claim, wherein said candidate pattern is a speech utterance, and said knowledge source comprises one or more of a phoneme lexicon, a word lexicon and a grammar lexicon.

20 11. The method of claim 10, wherein the knowledge source comprises all of phoneme lexicon, a word lexicon and a grammar lexicon stored in a single memory structure.

25 12. The method of any preceding claim, wherein token propagation is run on different threads in parallel.

13. A pattern recognition system, comprising:
a spectral analyser for analysing a candidate pattern to obtain a set of
30 observation vectors;

32

a decoder for decoding said observation vectors to produce a candidate match between said observation vectors and entries from a knowledge source;

5 a memory resource having stored thereon a weighted finite state transducer (WFST) that embodies said knowledge source; and

10 a processor arranged, for each successive propagation, to model the propagation of tokens through nodes of the WFST for observation vectors derived from successive frames of the candidate pattern, to sort a list of tokens in order of their associated cost; and to apply a pruning step to remove tokens from said sorted list which go beyond a pre-determined cost threshold.

15 14. The system of claim 13, wherein the sorting comprises binning a candidate token into one of a set of bins, each of which is associated with a range of cost values; and the pruning step comprises ignoring the tokens in the bin associated with the highest costs.

20 15. The system of claim 14, wherein the pruning step comprises ignoring the tokens in a plurality of bins, being the set of bins associated with the highest costs.

16. The system of claim 14 or claim 15, wherein the cost thresholds for each bin scale non-linearly, with lower cost bins being larger than higher cost bins.

25 17. The system of any of claims 13 to 16, wherein the token propagation is carried out by a dynamic programming algorithm.

18. The system of claim 17, wherein said dynamic programming algorithm is a Viterbi decoding algorithm.

30 19. The system of any of claims 13 to 18, wherein transitions between nodes of the WFST correspond to Hidden Markov Model (HMM) states.

33

20. The system of any of claims 13 to 19, wherein the propagation of tokens through the WFST comprises, for each transition, the calculation of a cost that includes a contribution from the coefficient of the determined observation vector and a fixed arc cost that represents the cost associated with a transition to a given HMM state.

21. The system of claim 20, wherein a set of said arc costs is stored as part of the knowledge source.

22. The system of any of claims 13 to 21, wherein said candidate pattern is a speech utterance, and said knowledge source comprises one or more of a phoneme lexicon, a word lexicon and a grammar lexicon.

23. The system of claim 22, wherein the knowledge source comprises all of phoneme lexicon, a word lexicon and a grammar lexicon stored in a single memory structure.

24. The system of any of claims 13 to 23, wherein, token propagation is run on different threads in parallel.

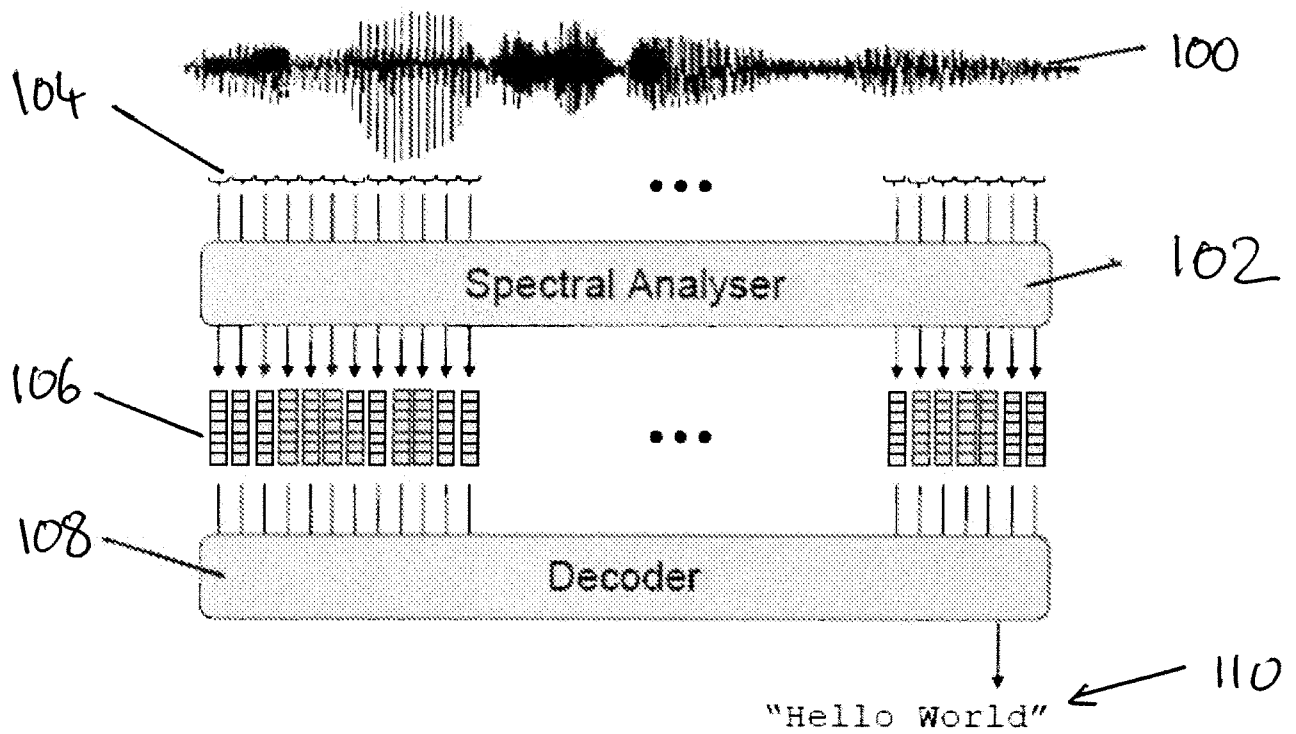
25. An embedded system comprising a pattern recognition system, comprising a spectral analyser for analysing a candidate pattern to obtain a set of observation vectors; a decoder for decoding said observation vectors to produce a candidate match between said observation vectors and entries from a knowledge source; and a memory resource having stored thereon a weighted finite state transducer (WFST) that embodies said knowledge source; and a processor arranged, for each successive propagation, to model the propagation of tokens through nodes of the WFST for observation vectors derived from successive frames of the candidate pattern, to sort a list of tokens in order of their associated cost; and to apply a pruning step to remove tokens from said sorted list which go beyond a pre-determined cost threshold.

26. A portable device comprising the embedded system of claim 25.

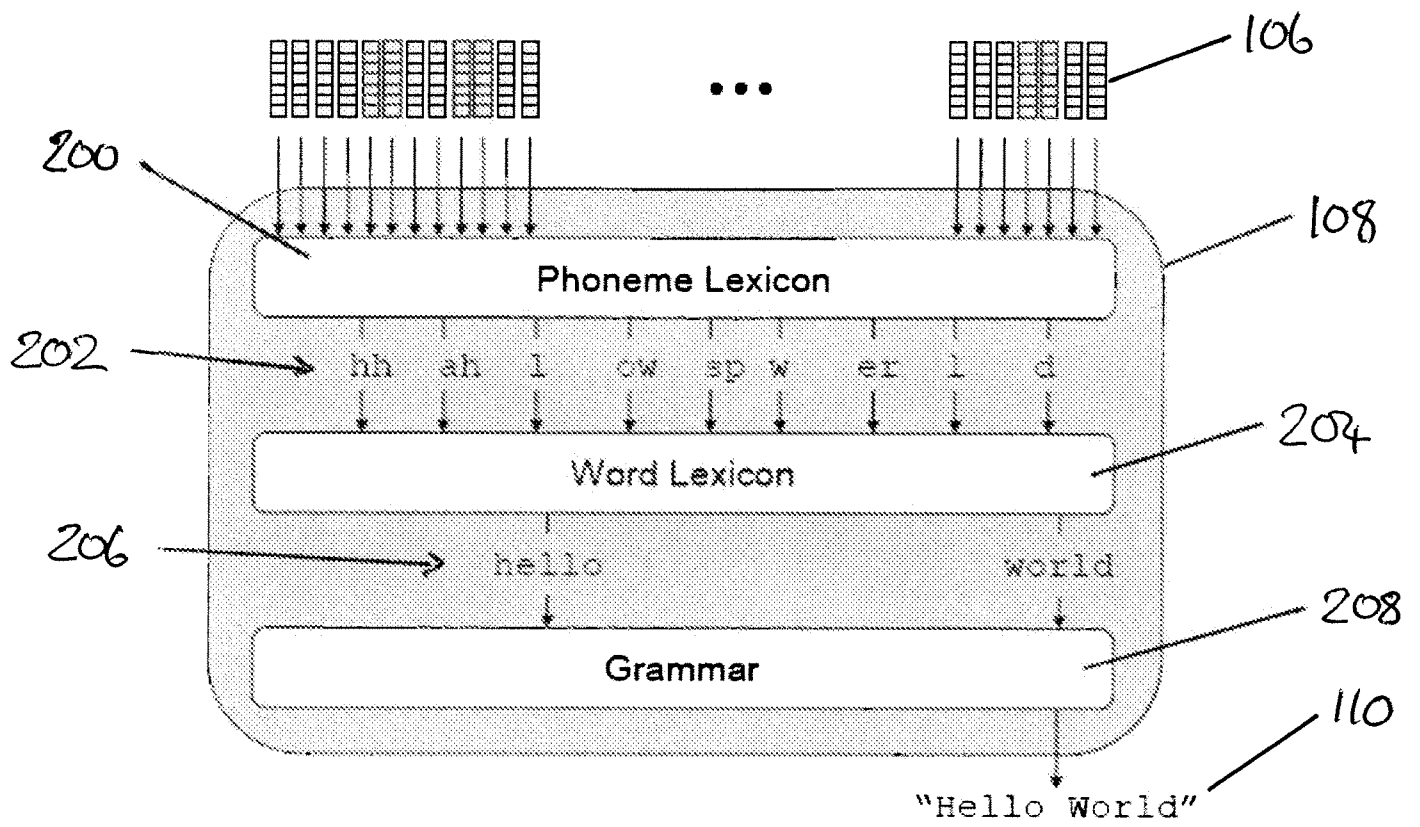
5 27. The portable device of claim 26, being a hand held mobile communications device.

28. A computer programme product a WFST and instructions for carrying out the method of any of claims 1 to 12.

1/15

**Fig. 1**

2/15

**Fig. 2**

3/15

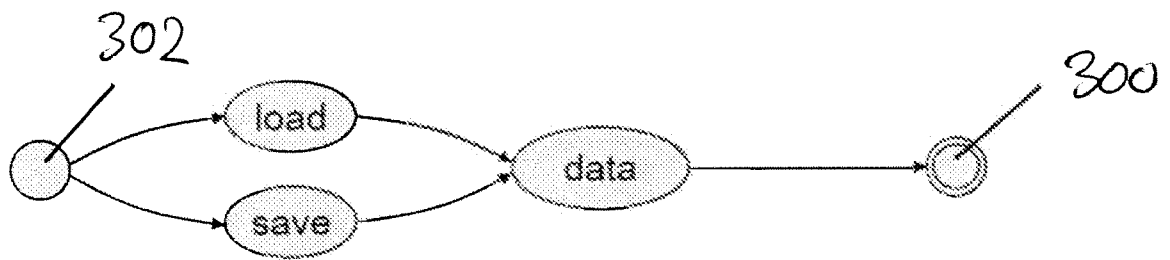


Fig. 3

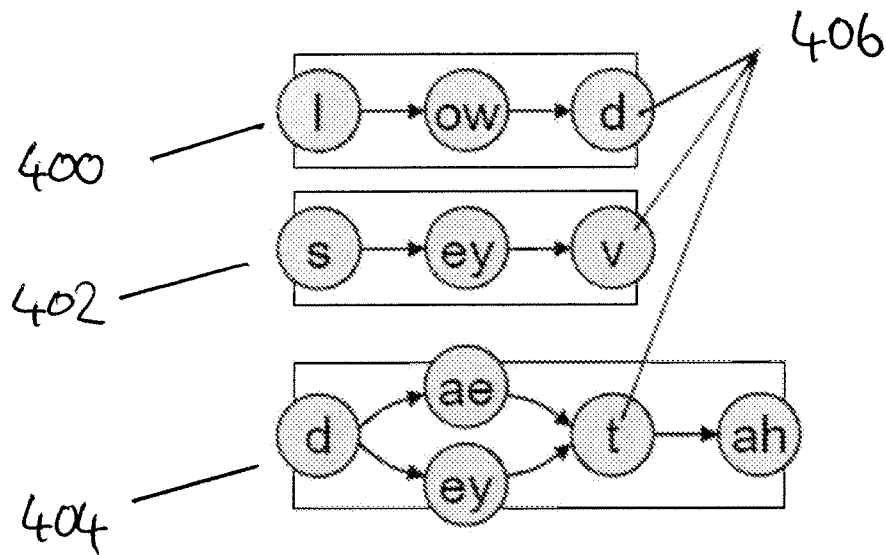


Fig. 4

4/15

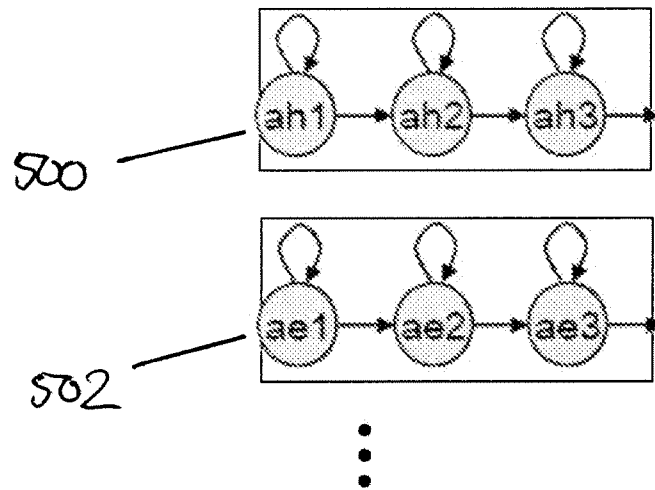


Fig. 5

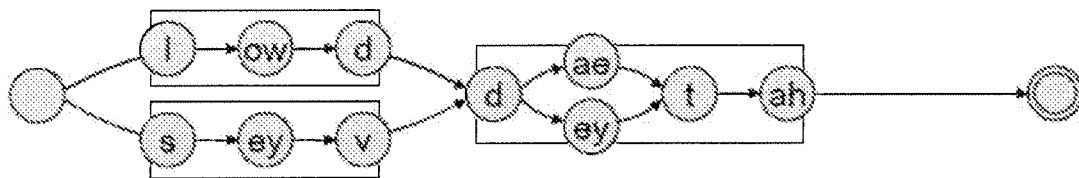
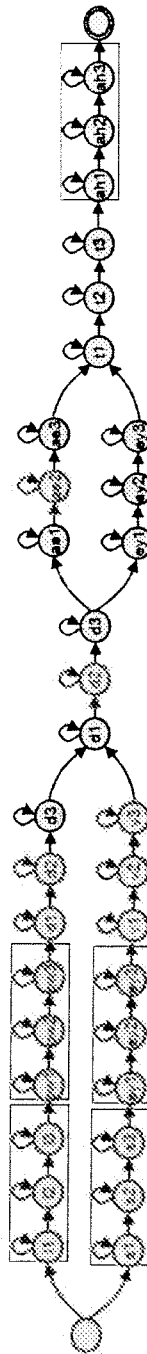
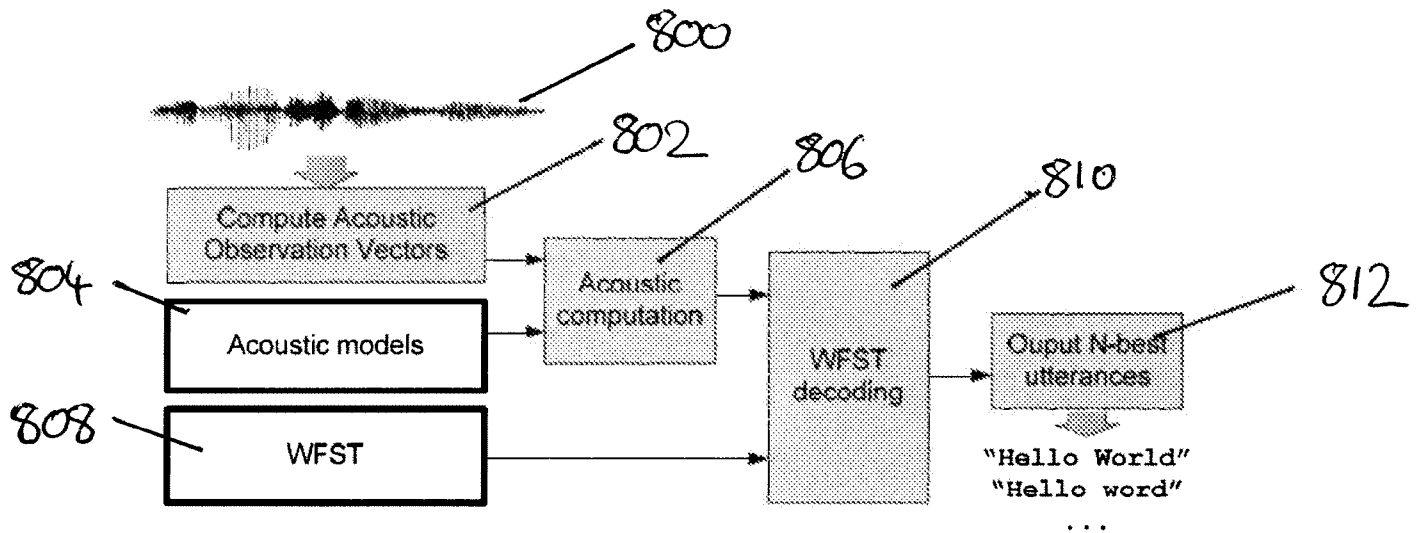


Fig. 6

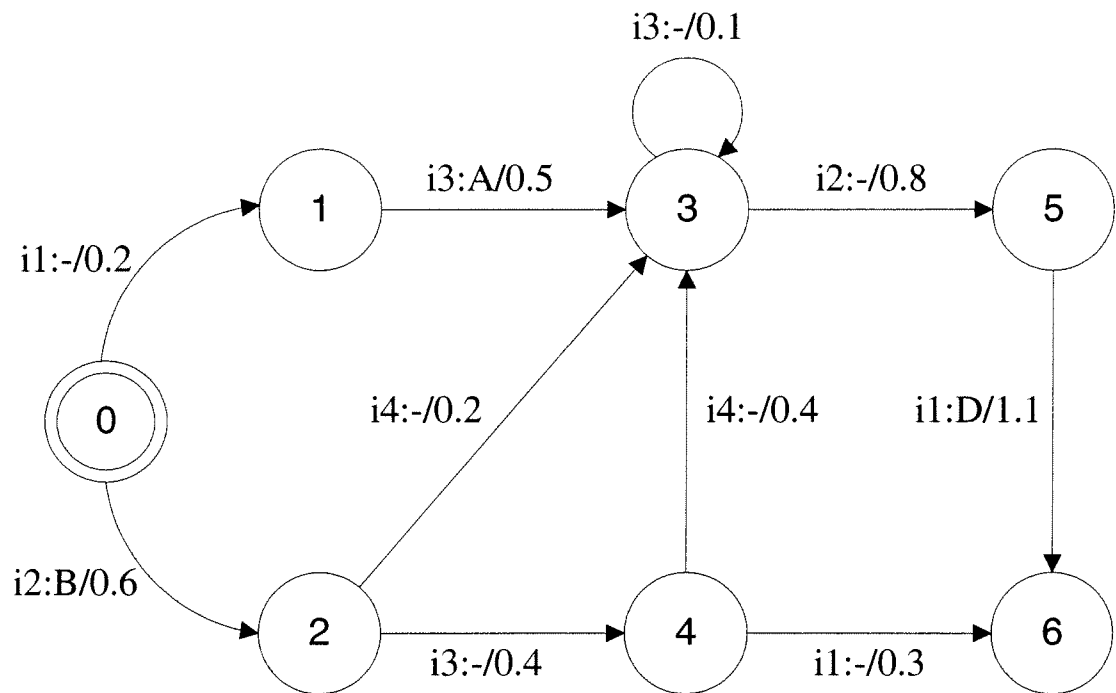
5/15

**Fig. 7**

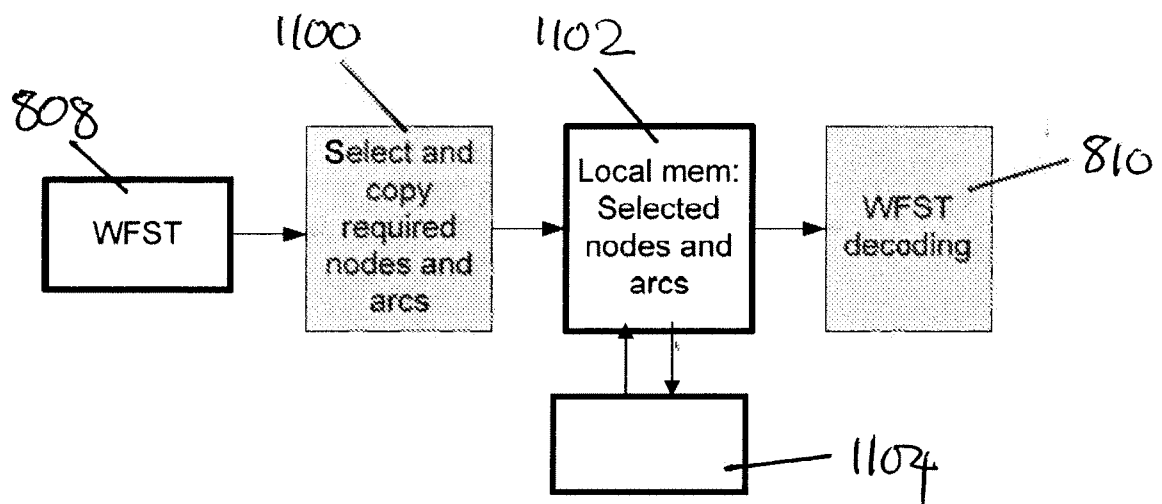
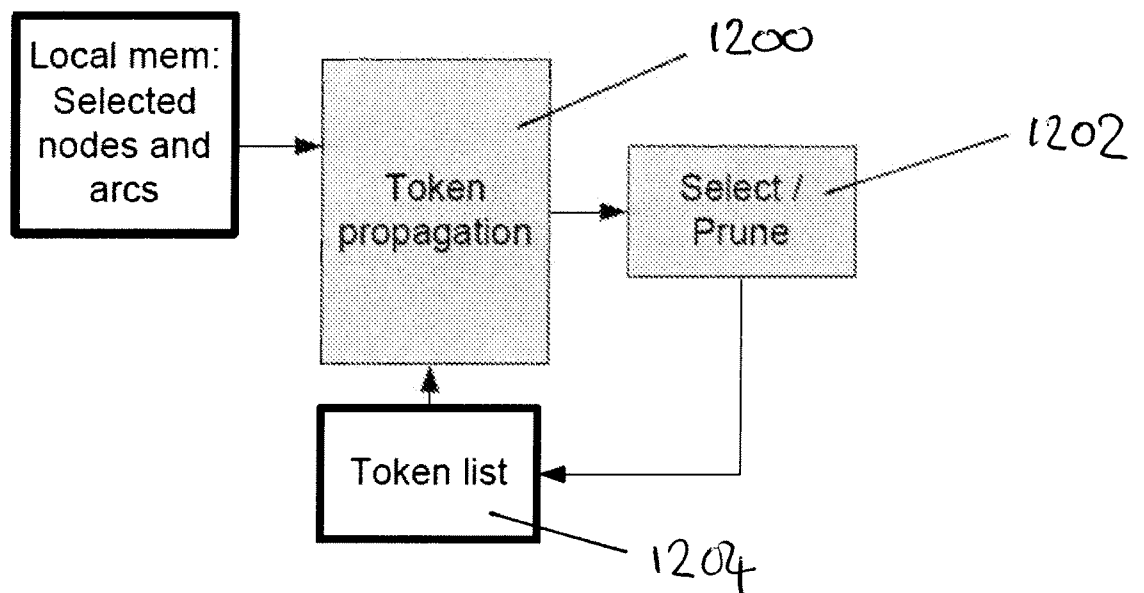
6/15

**Fig. 8**

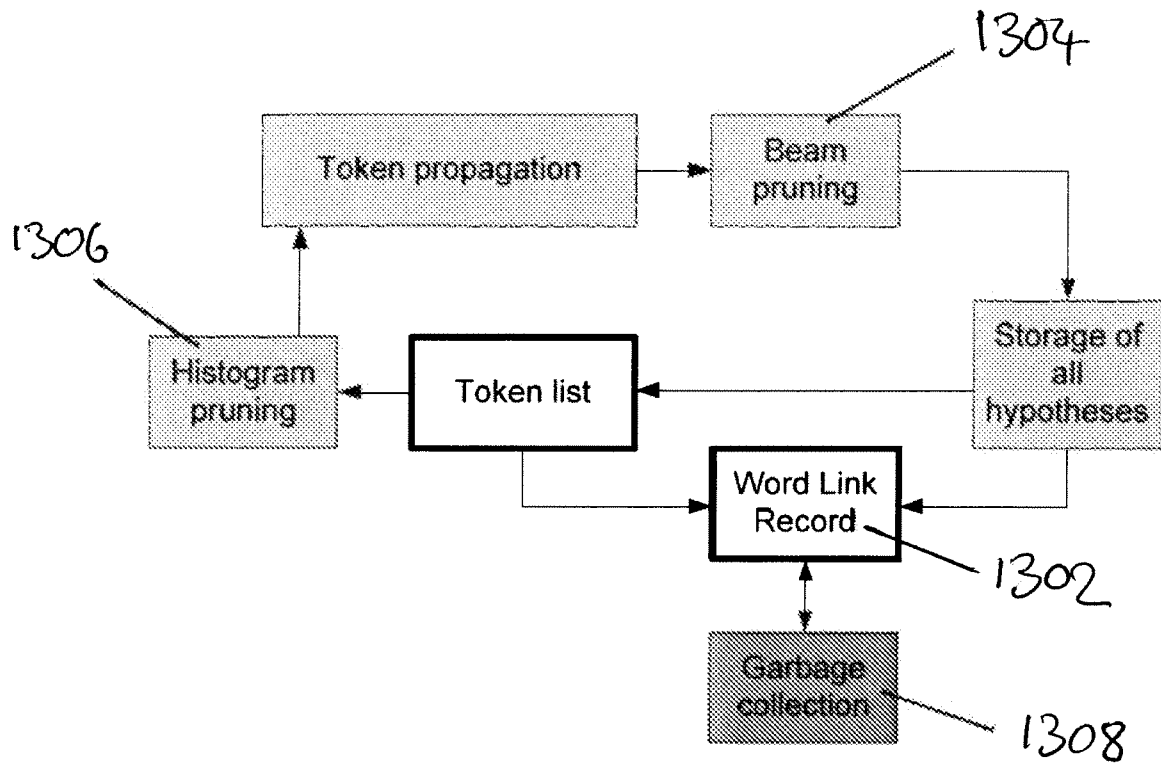
7/15

**Fig. 9**

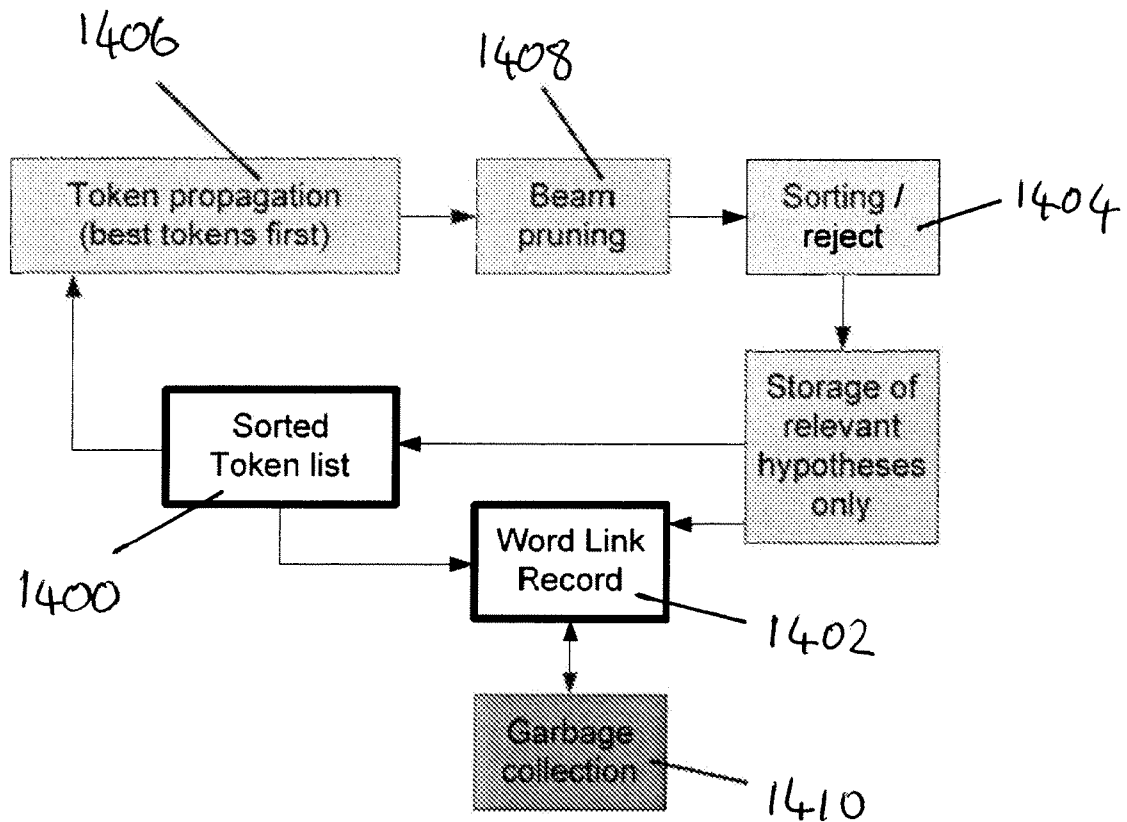
8/15

**Fig. 10****Fig. 11**

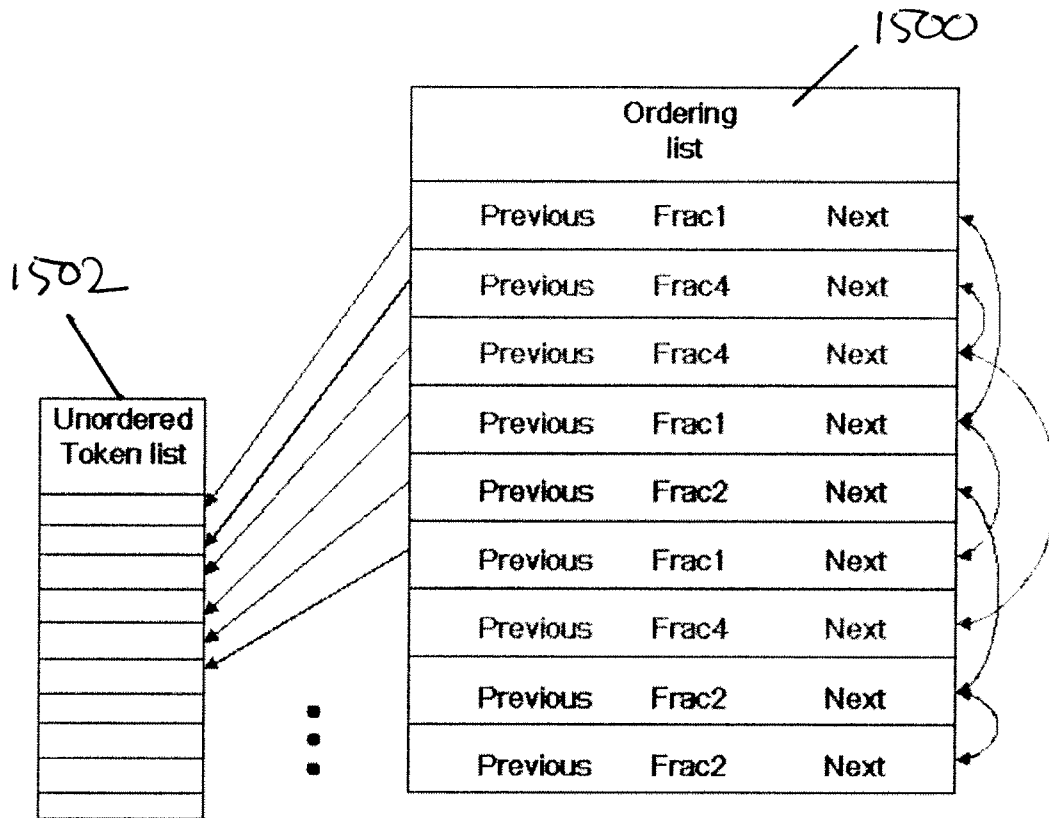
9/15

**Fig. 12**

10/15

**Fig. 13**

11/15

**Fig. 14**

12/15

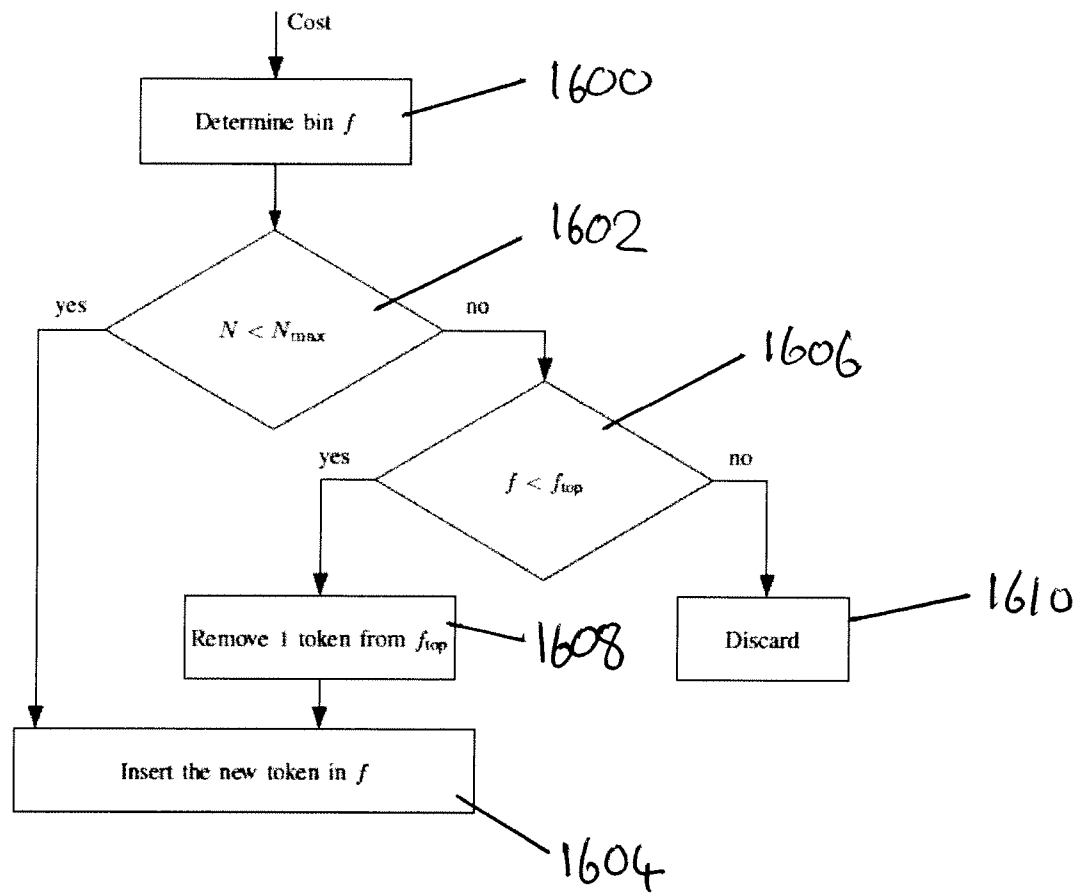


Fig. 15

14/15

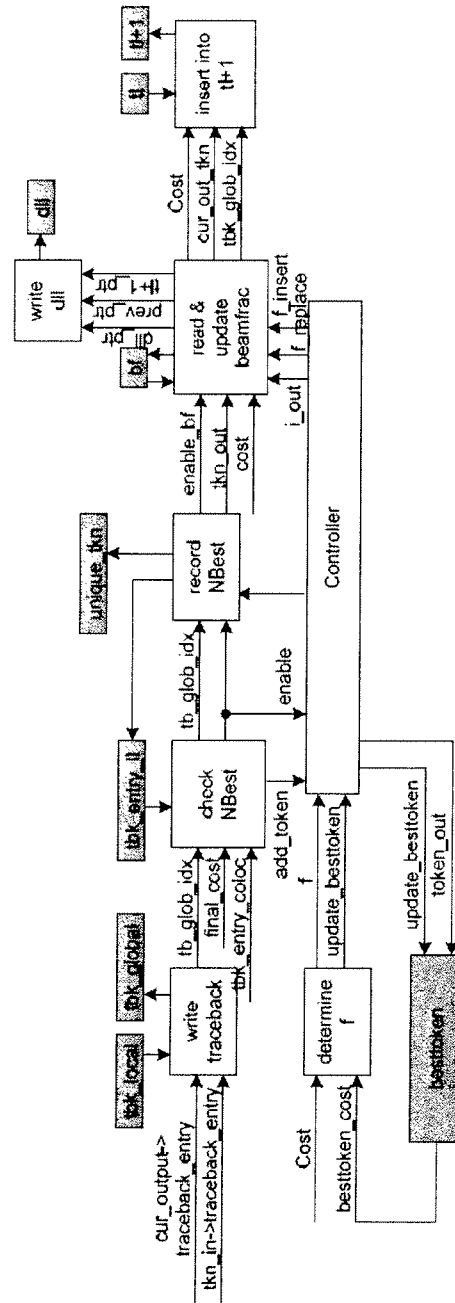
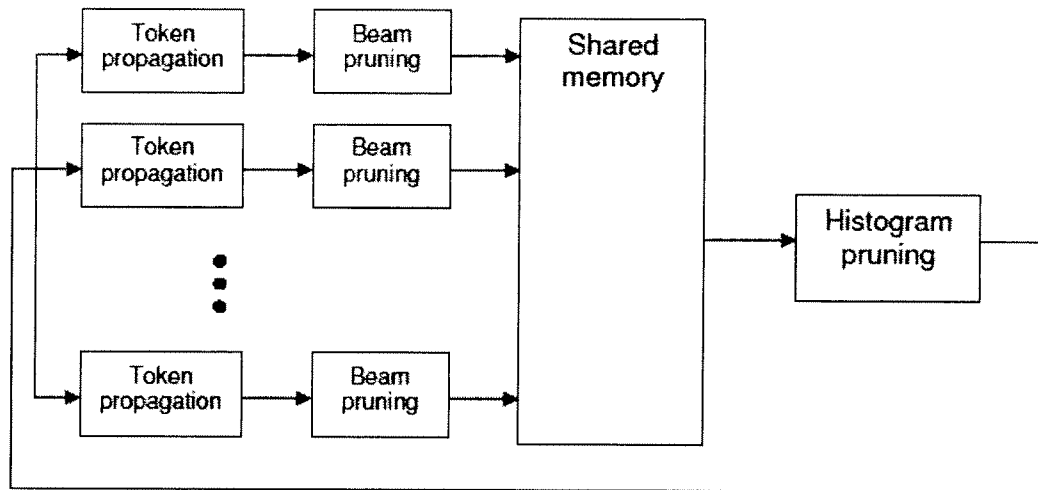
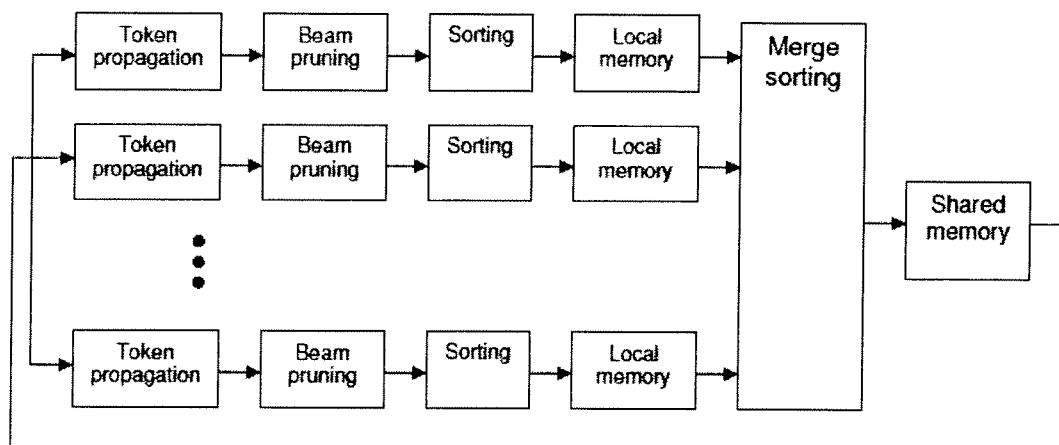


Fig. 17

15/15**Fig. 18****Fig. 19**

INTERNATIONAL SEARCH REPORT

International application No

PCT/GB2011/052436

A. CLASSIFICATION OF SUBJECT MATTER

INV. G10L15/08

ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G10L

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2007/143112 A1 (YU DONG [US] ET AL) 21 June 2007 (2007-06-21) paragraphs [0001] - [0003] paragraphs [0026] - [0028] paragraphs [0035], [0064] paragraphs [0074] - [0098] paragraphs [0107] - [0109] -----	1-28
X	BILMES J: "Dynamic Graphical Models", IEEE SIGNAL PROCESSING MAGAZINE, IEEE SERVICE CENTER, PISCATAWAY, NJ, US, vol. 27, no. 6, 1 November 2010 (2010-11-01), pages 29-42, XP011317699, ISSN: 1053-5888 page 35 ----- -/-	1-28



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search

15 February 2012

Date of mailing of the international search report

23/02/2012

Name and mailing address of the ISA/

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040,
Fax: (+31-70) 340-3016

Authorized officer

Ramos Sánchez, U

INTERNATIONAL SEARCH REPORT

International application No

PCT/GB2011/052436

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT		
Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	TSUNEO KATO ET AL: "An efficient beam pruning with a reward considering the potential to reach various words on a lexical tree", ACOUSTICS SPEECH AND SIGNAL PROCESSING (ICASSP), 2010 IEEE INTERNATIONAL CONFERENCE ON, IEEE, PISCATAWAY, NJ, USA, 14 March 2010 (2010-03-14), pages 4930-4933, XP031697068, ISBN: 978-1-4244-4295-9 last paragraph of section 2.2 on page 4932 -----	1-28

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/GB2011/052436

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2007143112	A1	21-06-2007	NONE
