(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
4 April 2013 (04.04.2013)

WIPO | PCT

(10) International Publication Number
WO 2013/045642 A1

(51) International Patent Classification:
G06F 17/30 (2006.01) *G06F 17/27* (2006.01)
G06F 3/023 (2006.01)

(21) International Application Number:
PCT/EP2012/069233

(22) International Filing Date:
28 September 2012 (28.09.2012)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11183544.3 30 September 2011 (30.09.2011) EP

(71) Applicant: **HARMAN BECKER AUTOMOTIVE SYSTEMS GMBH** [DE/DE]; Becker-Göring-Straße 16, 76307 Karlsbad (DE).

(72) Inventors: **SPINDLER, CC**; Nelkenstr. 7a, 76135 Karlsruhe (DE). **HEITMANN, Marcus**; Starenweg 7, 85386 Eching (DE). **BAPTIST, Stefan**; Franz-Fihl-Str. 4a, 80992 München (DE). **WELSCHER, Jürgen**; Burgerfeld 31c, 85570 Markt Schwaben (DE).

(74) Agent: **MEIER, Florian**; Kraus & Weisert, Thomas-Wimmer-Ring 15, 80539 München (DE).

(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, BN, 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, PA, 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, 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: METHOD OF GENERATING SEARCH TREES AND NAVIGATION DEVICE

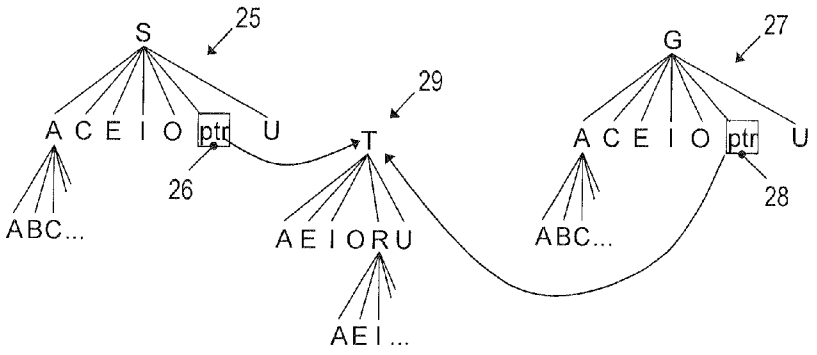

Fig. 4

(57) Abstract: A method of generating search trees (25, 27) indicating next valid characters for an input interface of a navigation device includes determining a search sub-tree (29) which indicates next valid characters for both a subset of a first set of character strings and for a different second set of character strings. A first search tree (25) is generated based on information on the first set of character strings, and a second search tree (27) is generated based on information on the second set of character strings. The first search tree (25) is generated such that a node (26) of the first search tree (25) references the search sub-tree (29). The second search tree (27) is generated such that another node (28) of the second search tree (27) references the search sub-tree (29).

WO 2013/045642 A1

Method of generating search trees and navigation device

Technical Field

Embodiments of the invention relate to data base search for use in navigation devices. Embodiments of the invention relate in particular to a method of generating search trees which indicate next valid characters for inputting names at an input interface of a navigation device, and to a navigation device using such search trees.

Background

Navigation devices are known which perform functions such as route searches between two locations. Modern navigation devices also may provide additional functionalities, such as serving as a travel guide that outputs, upon demand, information on points of interest (POI). Such information may include names of streets or POIs, but may also include additional textual or multimedia information. For illustration, some navigation devices may include travel guide functions to output detailed explanations, in textual and/or multimedia form, on objects, in response to receiving a name input.

In order to aid a user in inputting names, navigation devices may be configured to provide information on all potential next valid characters. This reduces or eliminates the risk that the user inputs a name which is not regarded valid by the navigation device. For illustration, by providing information on next valid characters and/or by limiting possible character inputs based on information on next valid characters, the inputting of a destination city or destination street may be prevented if the navigation

device does not store information on the location of the respective city or street.

Information on next valid characters should desirably be provided without substantial delays, i.e., while the user is still in the process of inputting a name. In order to allow information on next valid characters to be retrieved quickly, dedicated search structures may be used. Such search structures may be specific search trees, which are also referred to as next valid character (NVC) trees. NVC trees may be used for different search paths. For illustration, there may be a NVC tree for valid city names in one country, such as Germany, and another NVC tree for valid city names in another country. There may be a NVC tree for valid road names in one city, and another NVC tree for valid road names in other cities. Leafs of the NVC tree may point to data base objects, for example. Alternatively, leafs of the NVC trees may also merely indicate that there are no valid names which include more characters, or may point to other data structures. For illustration, a leaf of NVC tree which indicates next valid characters for country names may point to a root node of another NVC tree which stores next valid characters for all cities in the respective country, or similar.

When a large number of possible names must be accommodated, storage space requirements for storing the various NVC trees may be significant. The addition of ever more functionalities, such as travel guide functionalities, which may also call for a verification of the validity of name input during the ongoing inputting of characters, may further increase such storage space requirements.

Summary

Accordingly, there is a need for methods and navigation devices which allow storage space requirements for next valid character (NVC) trees to be reduced.

According to exemplary embodiments, a method and a navigation device as recited in the independent claims are provided to address these needs. The dependent claims define further embodiments.

According to an embodiment, a method of generating search trees which indicate next valid characters for an input interface of a navigation device is provided. Information on a first set of character strings and information on a second set of character strings is retrieved. The first and second sets are different from each other, but do not need to be disjoint. Based on the information on the first set and the information on the second set, a search sub-tree is determined which indicates next valid characters for both a subset of the first set of character strings and another subset of the second set of character strings. A first search tree is generated based on the information on the first set. A second search tree is generated based on the information on the second set. The first search tree and the second search tree are generated and stored in a data base such that a node of the first search tree references the search sub-tree, and that another node of the second search tree references the search sub-tree.

By identifying the search sub-tree which indicates next valid characters for both a subset of the first set of character strings and another subset of the second set of character strings, redundancies may be reduced. Portions of NVC trees which are identical do not need to be stored plural times. Rather, one and the same sub-tree may be referenced both by a node of the first search tree and by a node of the second search tree.

The search sub-tree indicates next valid characters and is referenced by both a node of the first search tree and a node of the second search tree. This allows search trees to be stored in a way which is efficient in terms of storage space. The search sub-tree may include plural nodes, with at least one of the nodes referencing another one of

the nodes of the search sub-tree, so as to indicate next valid characters. The search sub-tree may include plural nodes, with a plurality of the nodes respectively referencing at least two other nodes of the search sub-tree, so as to indicate next valid characters. The search sub-tree may indicate next valid characters and may be referenced by both a search tree for next valid characters of city or road names and by another search tree for next valid characters of points of interest.

To determine the search sub-tree, a first provisional search tree may be generated based on the information on the first set, wherein the first provisional search tree has a leaf node for each character string included in the first set. A second provisional search tree may be generated based on the information on the second set, wherein the second provisional search tree has a leaf node for each character string included in the second set. The first provisional search tree and the second provisional search tree may be compared to each other to determine the search sub-tree. The first provisional search tree may be a conventional NVC tree for the first set of character strings. The second provisional search tree may be a conventional NVC tree for the second set of character strings. By comparing such complete NVC search trees to each other, identical portions of the different search trees may be identified easily and reliably.

To determine the search sub-tree, a portion of the first provisional search tree may be identified which is identical to a portion of the second provisional search tree. Thereby, redundant information may be reduced or eliminated efficiently.

The first search tree may be generated based on the first provisional search tree. To this end, the first provisional search tree may be truncated by removing search paths included in the search sub-tree from the first provisional search tree. A reference to the search sub-tree may be added at a node of the truncated first provisional search tree. With such a structure, portions of search paths which correspond, for example, to name terminations (such as “...street”, “...strasse”, “...platz”), may be removed

from the first provisional search tree to generate the first search tree. By adding the reference to the search sub-tree, a NVC search path which starts in the first search tree may be continued in the search sub-tree, which is different from the first search tree.

The second search tree may include the search sub-tree. The node of the first search tree may references a node of the second search tree which is included in the search sub-tree. Thereby, the second search tree may still represent, of its own, a complete NVC search tree. A portion of the second search tree, which represents sub-strings of some of the character strings included in the first set of character strings, may be referenced by the first search tree to reduce redundancy.

Alternatively, the search sub-tree may be separate from both the first search tree and the second search tree. In this case, generating the second search tree may comprises truncating the second provisional search tree by removing search paths included in the search sub-tree from the second provisional search tree. A reference to the search sub-tree may be added at a node of the truncated second provisional search tree.

The first search tree, the second search tree and the search sub-tree may be stored separately in a data base. Alternatively, the second search tree may be stored as a portion of one of the first and the second search trees. The data base may be a relational data base. The first search tree, the second search tree and, if applicable, the search sub-tree may be stored as binary large objects (BLOBs).

The first set of character strings and the second set of character strings may be selected from country names, city names, street names, or names of points of interest (POI). For illustration, the different sets of character strings may correspond to valid street names for different cities, or city districts, in one country.

The method is performed automatically by an electronic computing device. The information on the first set of character strings may be the set of character strings itself. The information on the second set of character strings may be the set of character strings itself. Alternatively, the information may already have undergone some processing. For illustration, to generate a set of search trees which have reduced redundancy, conventional NVC trees for the first set of character strings and for the second set of character strings may be retrieved as input, and may be processed to identify portions of search paths which are identically included in at least two of the conventional NVC trees search trees.

The method may be applied to more than two sets of character strings. For illustration, a search sub-tree may be referenced by nodes of more than two search trees. There may also be more than one search sub-tree. For illustration, there may be a search sub-tree which is referenced by nodes of a pair of search trees, and another search sub-tree which is referenced by nodes of another pair of search trees. A search tree may also reference plural different search sub-trees.

According to another embodiment, a navigation device data base is provided. The data base stores plural search trees which respectively indicate next valid characters for a name input received at an input interface of the navigation device, wherein a first search tree has a first node referencing a search sub-tree and a second search tree has a second node referencing the same search sub-tree. The search sub-tree may indicate next valid characters both for a name search which starts at a root node of the first search tree, and for a name search which starts at a root node of the second search tree. The data base may be a data base generated with a method of any one aspect or embodiment described herein.

According to another embodiment, a navigation device is provided. The navigation device comprises an input interface configured to receive character input in a sequential manner. The navigation device comprises a data base storing plural search

trees which respectively indicate next valid characters for a name input received at the input interface, wherein a first search tree has a first node referencing a search sub-tree and a second search tree has a second node referencing the same search sub-tree. The navigation device comprises a processing device coupled to the input interface and the data base to perform a next valid character search. The processing device is configured to determine, when at least one character input is received at the input interface, next valid characters using at least one of the plural search trees stored in the data base. The processing device is also configured to selectively continue the next valid character search in the search sub-tree when reaching either one of the first node of the first search tree or the second node of the second search tree.

By configuring the processing device such that it may continue a NVC search in the search sub-tree, use may be made of the search trees which have reduced redundancy.

The search sub-tree indicates next valid characters and is referenced by both a node of the first search tree and a node of the second search tree. This allows search trees to be stored in a way which is efficient in terms of storage space. The search sub-tree may include plural nodes, with at least one of the nodes referencing another one of the nodes of the search sub-tree, so as to indicate next valid characters. The search sub-tree may include plural nodes, with a plurality of the nodes respectively referencing at least two other nodes of the search sub-tree, so as to indicate next valid characters. The search sub-tree may indicate next valid characters and may be referenced by both a search tree for next valid characters of city or road names and by another search tree for next valid characters of points of interest.

The data base may store plural search sub-trees, each of which is respectively referenced by nodes of at least two different search trees. Thereby, storage space requirements may be reduced further.

The processing device may be configured to pre-load the search sub-tree prior to reaching the first node or the second node in the next valid character search. The processing device may be configured such that the search sub-tree is pre-loaded when performing a NVC search starting at a root node of one of the first and second search trees, and to utilize the pre-loaded search sub-tree when subsequently performing a NVC search starting at a root node of the other one of the first and second search trees.

The search sub-tree may be included in one of the first search tree and the second search tree. The search sub-tree may be stored in the data base separately from the first search tree and the second search tree. The effects respectively attained thereby correspond to the effects described in the context of methods of embodiments.

Each one of the first and second search trees may indicate next valid characters for country names, city names, street names, or names of POIs.

The navigation device may be a navigation device for vehicular navigation. The navigation device may be fixedly installed in a vehicle. The navigation device may be a portable navigation device.

It is to be understood that the features mentioned above and those to be explained below can be used not only in the respective combinations indicated, but also in other combinations or in isolation.

Brief description of the drawings

The foregoing and other features of embodiments will become more apparent from the following detailed description of embodiments when read in

conjunction with the accompanying drawings. In the drawings, like reference numerals refer to like elements.

Fig. 1 is a schematic block diagram of a navigation device.

Fig. 2 is a schematic representation of search trees for performing a next valid character search.

Fig. 3 is a schematic representation of data entries of a search tree for performing a NVC search.

Fig. 4 illustrates a modification of the search tree structure of Fig. 3 according to an embodiment.

Fig. 5 is a flow chart of a method of an embodiment.

Fig. 6 is a schematic view illustrating data processing in the method of Fig. 5.

Figs. 7-10 respectively are schematic representations of search trees for performing a next valid character search according to embodiments.

Detailed description

While embodiments of the invention will be described with reference to the drawings in specific contexts hereinafter, it is to be understood that various modifications and alterations may be made to the embodiments.

Fig. 1 schematically illustrates a vehicle navigation device 1 according to an embodiment. The navigation device 1 comprises a processing device 2 controlling the operation of the navigation device 1, e.g. according to control instructions stored in a memory. The processing device 2 may comprise a

central processing unit, for example in form of one or more processors, digital signal processing devices or application-specific integrated circuits. The navigation device 1 further includes a data base 3. The data base 3 may comprise any one, or any combination, of various types of storage or memory media, such as random access memory, flash memory or a hard drive, but also removable memories such as a compact disk (CD), a DVD, a memory card or the like. The navigation device 1 also includes an output interface 4 for outputting information to a user. The output interface 4 may include an optical output device, an audio output device, or a combination thereof. The navigation device 1 also includes an input interface 5 which allows a user to input characters in a sequential manner. The input interface 5 may allow a user to input textual information or voice information.

The navigation device may include additional components, such as a position sensor and/or a wireless receiver and/or a vehicle interface. The position sensor may be adapted to determine the current position of the vehicle in which the navigation device 1 is installed. The position sensor may comprise a GPS (Global Positioning System) sensor, a Galileo sensor, a position sensor based on mobile telecommunication networks and the like. The wireless receiver may be configured to receive information for updating the data base stored in the storage device 3. The vehicle interface may allow the processing device 2 to obtain information from other vehicle systems or vehicle status information via the vehicle interface. The vehicle interface may for example comprise CAN (controller area network) or MOST (Media Oriented devices Transport) interfaces.

The data base 3 includes data 10 which represent search trees for performing next valid character (NVC) searches. Some or all of the search trees for NVC searches may correspond to names of various objects. Examples include names of valid destinations, such as destination countries, destination cities, or destination streets. Some of the search trees for performing the NVC searches may also correspond to other names, such

as names of POIs or names for which there are entries in an electronic travel guide.

The processing device 2 may access and utilize the data 10 when a user starts inputting a name. For illustration, the processing device 2 may determine possible next valid characters in response to a sub-string which was already entered by a user at the input interface 5. The processing device 2 may control the optical output device 4 to output the next valid characters. Alternatively or additionally, the processing device 2 may control the input interface 5 so as to prevent, or otherwise bar, the inputting of characters which do not correspond to next valid characters. When performing a NVC search, the processing device 2 may start at a root node of a search tree. The search tree may be selected based on the type of name which is to be input, such as country or city or road. As will be described in more detail in the following, the navigation device 1 uses search trees for performing a NVC search which have reduced redundancy as compared to conventional NVC trees. This is attained by allowing several search trees to reference, i.e. point to, one and the same search sub-tree. Thereby, sub-strings which are identically encountered in different sets of character strings may be accommodated, without requiring the corresponding search paths to be always stored plural times.

Fig. 2 is a schematic representation of data 10 which include plural search trees for performing NVC searches. The data 10 may be included in different portions of the data base 3. For illustration, if the data base 3 includes separate sections, or layers, for names or for POIs, the data 10 may be included in the respective section or layer.

The data 10 includes a first search tree 11 and a second search tree 12. For illustration rather than limitation, the search tree 11 may represent names of streets in one geographical area and the search tree 12 may represent names of streets in another geographical area different from the one

geographical area. Both the first search tree 11 and the second search tree 12 represent next valid characters for name input at the input interface 5 of the navigation device. For illustration, depending on the country and city which was previously input, one of the search trees 11, 12 is selected for performing a NVC search. As more characters of a name, e.g. a street name, are successively entered, a search path is traversed in the respective search tree 11, 12. The search trees 11, 12 may respectively be organized such that the child nodes of the current node represent the next valid characters. The processing device 2 may readily retrieve these next valid characters and may output them via the optical output device 4 and/or may control the input interface 5 based on the next valid characters retrieved from the respective search tree 11, 12. Different search trees may be associated with different geographic areas, so that the various paths which may be taken along the search tree represent the names of all objects known to the navigation device and located in the respective area.

Contrary to conventional NVC trees, a node 14 of the first search tree 11 points to a search sub-tree 13. As illustrated in Fig. 2, the search sub-tree 13 may be included in the second search tree 12. The search sub-tree 13 may include name terminations which are valid name terminations both for some names of the set of names represented by the first tree 11 and for some strings of the set of names represented by the second tree 12. For illustration, the name terminations “...erplatz” or “...erstrasse” may be valid endings both when a user inputs the initial characters “Schill...”, included in the set of names represented by the search tree 11, and when the user inputs the initial characters “Bruckn...”, included in set of names represented by the search tree 12. Rather than storing the associated portions of the search path which correspond to this name termination both in the first tree and the second tree, the node 14 of the first search tree 11 references the search sub-tree 13.

In operation, when the processing device accesses different nodes of the first search tree 11, the search may traverse a path 16 which ends at node 14. The reference to the search sub-tree stored at the node 14 then causes the processing device 2 to continue the search at a node of the search sub-tree 13, to identify valid characters for the remainder of the name which has not yet been input.

References to a separate search sub-tree may be provided only along some of the paths of the first search tree 11. There may be search paths 15 which can be traversed in the first search tree 11 without being referred to any separate search sub-tree.

The search sub-tree 13 is not only referenced by the node 14 of the first search tree 11, but is also referenced by a parent node 19 in the second search tree 12. This reflects that the search sub-tree 13 represents possible sub-strings for names starting with a string represented by node 14 of the first search tree 11, but also for names starting with a string represented by node 19 of the second search tree 12. When a user inputs initial characters of a string while the processing device uses the second search tree 12 to identify next valid characters, a search path 18 may be traversed. When node 19 is reached, the NVC search continues in the search sub-tree 13.

It will be appreciated that the search sub-tree 13 which is referenced by nodes of plural search trees 11, 12 may have any number of branching nodes and/or levels. In particular, the search sub-tree 13 may include a plurality of branching nodes, which are parent nodes for respectively at least two child nodes. For illustration, when the first set of character strings includes the names “Schillerstrasse”, “Schillerplatz”, “Schillerallee”, and “Schillermuseum”, and the second set of character strings includes the names „Brucknerstrasse”, „Brucknerplatz”, „Bruckneralleee”, and „Brucknermuseum”, the search sub-tree 13 may have branching nodes which correspond to the different name terminations “...erstrasse”,

"...erplatz", "...erallee" and "...ermuseum", all of which are valid when the user either starts by inputting "Schill..." when the search is started in one search tree or when or starts by inputting "Bruckn..." when the search is started in another search tree.

If the leaf nodes of the search trees 11, 12 point to an object in the data base 3, several pointers may be stored in each leaf node of the common search sub-tree 13. For illustration, the leaf node reached in path 17 may contain two different pointers. One of the pointers may be used when the leaf node of path 17 is reached when starting the NVC search in the first search tree 11. Another one of the pointers may be used when the leaf node of path 17 is reached when starting the NVC search in another search tree, such as search tree 12. The processing device 2 may keep track of the search history. A disambiguation between different pointers stored at one leaf node of the common search sub-tree 13 may be performed based on the search history. If the leaf nodes of the search trees merely indicate that there are no names which include additional characters, but do not point to other objects in the data base 3, such a disambiguation may not be required.

Fig. 3 illustrates a search tree 20 for performing a NVC search in greater detail. Nodes of the search tree 20 may correspond to different characters. The characters may be alphanumerical characters. Each node represents a sequence of characters input at the input interface 5. The child nodes of a node represent the possible next valid characters, once the respective node has been reached. A search sub-tree 21 which would normally be included in plural different search trees may be stored only once. There may be plural parent nodes which reference, i.e. point to, the search sub-tree 21.

Fig. 4 illustrates search trees for performing NVC searches. The search trees include a first search tree 25, a second search tree 27, and search sub-tree 29. The search sub-tree 29 may be stored separately from both the first search tree 25 and the second search tree 27. The search sub-tree 29

represents valid sub-strings of names which have initial characters identified by the first search tree 25, and valid sub-strings of names which have initial characters identified by the second search tree 27. Correspondingly, the first search tree 25 has a node 26 which contains a pointer to the search sub-tree 29. The second search tree 27 has another node 28 which contains a pointer to the search sub-tree 29. By allowing the search sub-tree 29 to be referenced by parent nodes in plural different search trees 25, 27, redundancy may be reduced. The plural different search trees 25, 27 may correspond to different geographical areas, such as different countries, different cities, or different city districts.

Referring to Figs. 5 and 6, a method of generating search trees for performing NVC searches according to embodiments will be described. Generally, a search sub-tree is determined which represents sub-strings of next valid characters for different sets of character strings. For illustration, one set of character strings may correspond to the names of cities in on country, and the other set of character strings may correspond to the names of cities in another country. For further illustration, one set of character strings may correspond to the names of streets in on city of a given country, and the other set of character strings may correspond to the names of cities in another city of the same country. For further illustration, one set of character strings may correspond to POIs in one city, and the other set of character strings may correspond to POIs in another city.

One or plural search sub-tree(s) may be determined in various ways. For illustration, as will be explained with reference to Figs. 5 and 6, conventional NVC trees may be generated and compared to identify an identical portion. Other implementations are possible, which may be based, for example, on directly parsing the character strings of different sets of characters to identify common groups of possible sub-strings.

Fig. 5 is a flow chart of a method 30 of an embodiment. The method may be performed automatically by an electronic computing device when preparing a data base for use in a navigation device. The method 30 may be performed off board a vehicle.

At 31, plural sets of character strings are retrieved. The sets may be retrieved from electronic map data, electronic cadastral maps, or from a user input.

At 32, a conventional, complete NVC tree is generated for each set. This may be done by parsing the set of character strings, for each set, to identify character strings which have common initial sub-strings. The complete NVC trees serve as provisional search trees which are processed further to generate search trees having reduced redundancy.

At 33, the complete NVC trees are compared to each other to identify one or several sub-tree(s) which are identically included in at least two of the different complete NVC trees. Not all sub-tree(s) contained in at least two of the complete NVC trees may be of interest as far as reducing storage space requirements are concerned. For illustration, only sub-trees which have a certain size may be taken into account. For illustration, the search at 33 may be performed to identify sub-trees which have a number of branching nodes which is greater than a threshold, and/or which have a total number of child nodes which is greater than a threshold, and/or which have a total number of nodes which is greater than a threshold.

At 34, plural search trees are generated from the complete NVC trees. To this end, at least some of the complete NVC trees are truncated by removing therefrom the search paths which are included in one of the sub-tree(s) identified at 33. A pointer to the sub-tree is added at a node of the respective truncated NVC tree.

The plural search trees may be stored in a data base. If a search sub-tree is stored separately from the search trees, it is also stored in the data base. The data base may subsequently be deployed to navigation systems.

The plural search trees and the one or several search sub-tree(s) may be generated in various ways. For illustration, if two or more of the conventional, complete NVC trees generated at 32 share a sub-tree, only one of these complete NVC trees may be truncated. The search sub-tree may remain a part of the other complete NVC tree, which thus functions as search tree. Entries at leaf nodes of the search sub-tree may be adapted to account for the fact that such leaf nodes may be reached in NVC searches starting at different search trees. For illustration, each leaf nodes of the search sub-tree may store plural pointers or other data, with each pointer or data being associated with a different search history.

For further illustration, if two or more of the conventional, complete NVC trees generated at 32 share a sub-tree, all of these NVC trees may be truncated. The identified sub-tree may be stored separately. Entries at leaf nodes of the thus generated search sub-tree may be adapted to account for the fact that such leaf nodes may be reached in NVC searches starting at different search trees. Pointers to the search sub-tree are added at nodes of each one of the different truncated NVC trees.

Referring to Fig. 6, the method of Fig. 5 will be illustrated further. A first set of character strings 41 and a second set of character strings 51 are processed to generate search trees for a NVC search. The two sets 41, 51 are different. The two sets 41, 51 may even be disjoint in the sense that no character string of set 41 is included in character set 51 and vice versa. However, the two sets 41, 51 do not need to be disjoint.

Conventional complete NVC trees 42, 52 are generated. The NVC trees 42, 52 may be generated by parsing the respective sets of character strings.

Character strings may be grouped in accordance with sub-strings found at the beginning of the respective character strings. In other implementations, the method may start using the NVC trees 42, 52 as input.

The two complete NVC trees 42, 52 are compared to each other. An identical sub-tree portion 43 which is identically included in both NVC trees 42, 52 is identified.

The NVC tree 42 may be truncated by removing the sub-tree portion 43. The parent node of the sub-tree portion 43 in the complete NVC tree 42 may be replaced by a node 14 which stores a pointer, or otherwise references, the search sub-tree 13 which remains included in the second NVC tree. The parent node of the search sub-tree 43 is referenced both by node 14 of the first search tree 11 and by the parent node of the search sub-tree 43 in the second search tree 12. The second complete NVC tree 52 may be used as second search tree without any further modification in some implementations. In other implementations, the second search tree 12 may be generated from the second complete NVC tree 52 by adapting leaf nodes, so as to account of the fact that leaf nodes of the search sub-tree 13 may be reached both when starting in the first search tree 11 and when starting in the second search tree 12.

In other embodiments, both NVC trees 42, 52 may be truncated to generate the first and second search trees 11, 12, and the search sub-tree 13 may be generated based on the sub-tree portion 43 and may be stored separately from the first and second search trees 11, 12.

In still other embodiments, it is not required to generate the conventional, complete NVC trees 42, 52. For illustration, the search sub-tree as well as the first search tree and the second search tree which reference the search sub-tree may be determined by parsing the character strings of the first set 41 and of the second set 51.

In still other embodiments, the method may start using the complete NVC trees 42, 52 as input. Such complete NVC trees 42, 52 may be provided by vendors of electronic maps, for example.

The search trees and search sub-tree(s) may have any one of a variety of configurations, as will be illustrated further with reference to Figs. 7-10. Each one of the various configurations of search trees may alternatively or additionally be used in the data 10 to store search trees for NVC searches.

Fig. 7 is a schematic illustration of data 60 representing search trees 11, 62 and a search sub-tree 13 for performing NVC searches. In the data 60, the search sub-tree 13 is stored separately from the both the first search tree 11 and the second search tree 62. For illustration, the first search tree 11, the second search tree 62, and the search sub-tree 13 may be stored as separate binary large objects (BLOBs).

The first search tree 11 has a node 14 which references the search sub-tree 13. The second search tree 62 has a node 64 which references the search sub-tree 13. When node 14 is reached along search path 16, when a user inputs a sub-string of characters, the search continues in the search sub-tree 13 to identify the following next valid characters. Similarly, when node 64 is reached along search path 18, when a user inputs a sub-string of characters, the search continues in the search sub-tree 13 to identify the subsequent next valid characters of a name input.

When the data 60 are used in NVC searches, the search sub-tree 13 may be buffered for subsequent use. For illustration, the processing device 2 of the navigation device 1 may load the search sub-tree 13 when performing a NVC search starting at the second search tree 62. When another NVC search is subsequently performed starting at the first search tree 11, this buffered data may be used. This may help speed up NVC searches.

A search sub-tree may be referenced by more than two search trees. For illustration, there may be three or even more search trees, which respectively have a node that has a pointer to one and the same search sub-tree, as will be explained with reference to Fig. 8.

Fig. 8 is a schematic illustration of data 65 representing search trees 11, 62, 66 and a search sub-tree 13 for performing NVC searches. The first search tree 11 has a node 14 which references the search sub-tree 13. The second search tree 62 has a node 64 which references the search sub-tree 13. The third search tree 66 has a node 67 which references the search sub-tree 13. When node 14 is reached in a NVC search starting at the first search tree 11, or when node 64 is reached in a NVC search starting at the second search tree 62, or when node 67 is reached in a NVC search starting at the third search tree 64, the NVC search continues in the search sub-tree 13 to identify the following next valid characters.

While the data 65 illustrate the search sub-tree 13 as being stored separately from the first search tree 11, the second search tree 62, and the third search tree, the search sub-tree 13 may be included in either one of these trees.

The data base 3 may include plural search sub-trees. A search tree may have different nodes which each reference a different search sub-tree, as illustrated in Fig. 9.

Fig. 9 is a schematic illustration of data 70 representing search trees 71, 72, 73 and first and second search sub-tree 74, 75 for performing NVC searches. A first search tree 71 has a node 76 which references the first search sub-tree 74. A second search tree 72 has a node 77 which also references the first search sub-tree 74. The first search sub-tree 74 represents next valid characters both for a NVC search which starts at the first search tree 71 and

reaches node 76, and for a NVC search which starts at the second search tree 72 and reaches node 77.

The second search tree 72 has another node 78 which references the second search sub-tree 75. A third search tree 73 has a node 79 which also references the second search sub-tree 75. The second search sub-tree 75 represents next valid characters both for a NVC search which starts at the second search tree 72 and reaches node 78, and for a NVC search which starts at the third search tree 73 and reaches node 79.

While search sub-trees may correspond to name terminations, i.e., sub-strings of character strings located at the end of the respective character strings, this need not be the case. Rather, the principles of identifying a search sub-tree which represents sequences of characters that are next valid characters for different sets of character strings may be used also when the associated sub-strings are found in the center of names, for examples. This is illustrated in Fig. 10.

Fig. 10 is a schematic illustration of data 80 representing a first search tree 81 and a second search tree 82. The second search tree 82 includes a search sub-tree 13.

The search sub-tree 13 corresponds to sub-strings of next valid characters which are located in the middle of names represented by the first search tree 81 or at the end of names represented by the second search tree 82. The first search tree 81 has a node 84 which references the search sub-tree 13. A leaf node of the search sub-tree 13 may have two data, one of which references back to the first search tree 81. When a search is performed starting in the first search tree 81, node 84 may be reached along a search path 86 as a user inputs the first characters of a name. From node 84, the processing device 2 accesses the search sub-tree 13. The search may be continued along search path 87 through the search sub-tree 13 as the user continues

to input characters of the name. At a leaf node of the search sub-tree 13, a disambiguation may be made to determine whether this leaf node was reached in a search starting in the first search tree 81 or in the second search tree 82. When the leaf node of the search sub-tree 13 is reached in a search starting in the first search tree 81, the leaf node may reference back to a node of the first search tree 81, or to yet another search sub-tree. As the user continues to input characters of the name, the search for next valid characters may be continued along path 89 in the first search tree 81.

Similarly, the processing device 2 may reach a leaf node of the common search sub-tree 13 also when starting a search in the second search tree 82. When the search is started in the second search tree 82, for example because the second search tree 82 corresponds to a city previously selected by the user, a search path 88 may be traversed as a user inputs the first characters of a name. From the parent node of the root node of search sub-tree 13, the search continues in the search sub-tree 13. When a leaf node of the search sub-tree 13 is reached, disambiguation based on the search history may be performed.

In the devices and methods of various embodiments, the character strings represented by the search trees for performing next valid character searches may be strings of alphanumerical characters. In the devices and methods of various embodiments, the search sub-tree and the plural search trees which reference the same search sub-tree may be generated such that they correspond to sub-strings of names which have a certain length. For illustration, each path in the search sub-tree may correspond to name sub-strings having a length of two, of three, or of more than three characters. The nodes in the search trees which reference the search sub-tree may correspond to name sub-strings having a length of at least two characters. In other words, the search sub-tree may in particular correspond to sub-strings which represent next valid characters both for some names of a first name set and for some names of a second name set.

While devices and methods according to embodiments have been described in detail, modifications may be implemented in other embodiments. For illustration, while search trees have been described in the context of street names, city names, or country names, the search trees may represent any one of a variety of different names, such as names of POIs.

For further illustration, while exemplary structures of index trees and their nodes have been explained, any suitable data structure may be used to implement the index tree. For illustration, the index tree may be in a relational data base.

Embodiments of the invention may be used for vehicle navigation devices, which may be portable devices or devices fixedly installed in the vehicle. Embodiments of the invention are not limited to being used in this particular field.

CLAIMS

1. A method of generating search trees (11, 12; 25, 27; 11, 62; 11, 62, 66; 71, 72; 81, 82) indicating next valid characters for an input interface (5) of a navigation device (2), said method comprising:

retrieving information on a first set (41) of character strings and information on a second set (51) of character strings, said first and second sets (41, 51) being different from each other;

determining, based on said information on said first set (41) and said information on said second set (51), a search sub-tree (13; 29; 74, 75) which indicates next valid characters for both a subset of said first set (41) of character strings and another subset of said second set (51) of character strings;

generating a first search tree (11; 25; 71; 81) based on said information on said first set (41) and a second search tree (12; 27; 72, 73; 82) based on said information on said second set (51), wherein said first search tree (11; 25; 71; 81) and said second search tree (12; 27; 72, 73; 82) are generated such that a node (14; 26; 76; 84) of said first search tree (11; 25; 71; 81) references said search sub-tree (13; 29; 74, 75) and that another node (19; 28; 64; 64, 67; 77) of said second search tree (12; 27; 72, 73; 82) references said search sub-tree (13; 29; 74, 75), and storing said first search tree (11; 25; 71; 81) and said second search tree (12; 27; 72, 73; 82) in a data base (3).

2. The method of claim 1,

wherein said determining comprises:

generating a first provisional search tree (42) based on said information on said first set (41), wherein said first provisional search tree (42) has a leaf node for each character string included in said first set;

generating a second provisional search tree (52) based on said information on said second set (51), wherein said second provisional search tree (52) has a leaf node for each character string included in said second set (51);

comparing said first provisional search tree (42) and said second provisional search tree (52) to determine said search sub-tree (13; 29; 74, 75).

3. The method of claim 2,

wherein said determining comprises:

identifying a portion (43) of said first provisional search tree (42) which is identical to a portion (43) of said second provisional search tree (52).

4. The method of claim 3,

wherein said generating said first search tree (11; 25; 71; 81) comprises:

truncating said first provisional search tree (42) based on said identified portion (43); and

adding a reference to said search sub-tree (13; 29; 74, 75) at a node (14) of said truncated first provisional search tree.

5. The method of claim 4,

wherein said second search tree (12; 82) includes said search sub-tree (13),

wherein said node (14; 84) of said first search tree (11; 81) references a node of said second search tree (12; 82) which is included in said search sub-tree (13).

6. The method of claim 3 or claim 4,

wherein said generating said second search tree (12; 27; 72, 73; 82) comprises:

truncating said second provisional search tree (52) based on said identified portion (43); and

adding a reference to said search sub-tree (13; 29; 74, 75) at a node of said truncated second provisional search tree.

7. The method of claim 6,
wherein said first search tree (25; 71), said second search tree (27; 82) and said search sub-tree (29; 74) are stored as separate binary large objects.

8. The method of any one of the preceding claims,
wherein said first set (41) of character strings and said second set (51) of character strings are selected from country names, city names, street names, or names of points of interest (POI).

9. The method of any one of the preceding claims, further comprising:
retrieving information on at least one further set of character strings different from said first set (41) and said second set (51);
wherein said search sub-tree (13) is determined based on said information on said first set (41), said information on said second set (51) and said information on said at least one further set, such that said search sub-tree (13) indicates next valid characters for a subset of said first set (41), a subset of said second set (51), and a subset of said at least one further set; and
generating at least one further search tree (65) based on said information on said at least one further set, such that a node (67) of said at least one further search tree (65) references said search sub-tree (13).

10. A navigation device, comprising:
an input interface (5) configured to receive character input in a sequential manner;
a data base (3) storing plural search trees (11, 12; 25, 27; 11, 62; 11, 62, 66; 71, 72; 81, 82) which respectively indicate next valid characters for a name input received at said input interface (5), wherein a first search tree (11; 25; 71; 81) has a first node (14; 26; 76; 84) referencing a search sub-tree (13; 29; 74, 75) and a second search tree (12; 27; 72, 73; 82) has a second node (19; 28; 64; 64, 67; 77) referencing the same search sub-tree (13; 29; 74, 75); and

a processing device (2) coupled to said input interface (5) and said data base (3) to perform a next valid character search, said processing device (2) being configured

- to determine, when a character input is received at said input interface (5), next valid characters using at least one of said plural search trees (11, 12; 25, 27; 11, 62; 11, 62, 66; 71, 72; 81, 82) stored in said data base (3); and
- to selectively continue said next valid character search in said search sub-tree (13; 29; 74, 75) when reaching either one of said first node (14; 26; 76; 84) of said first search tree (11; 25; 71; 81) or said second node (19; 28; 64; 64, 67; 77) of said second search tree (12; 27; 72, 73; 82).

11. The navigation device (1) of claim 10,
wherein the data base (3) stores plural search sub-trees (74, 75), each of which is respectively referenced by nodes (76-79) of at least two different search trees (71-73).

12. The navigation device (1) of claim 10 or claim 11,
wherein said processing device (2) is configured to pre-load said search sub-tree (13; 29; 74, 75) prior to reaching said first node (14; 26; 76; 84) or said second node (19; 28; 64; 64, 67; 77) in said next valid character search.

13. The navigation device (1) of any one of claims 10 to 12,
wherein said search sub-tree (13; 29; 74, 75) is included in one of said first search tree (11; 25; 71; 81) and said second search tree (12; 27; 72, 73; 82).

14. The navigation device (1) of any one of claims 10 to 12,
wherein said search sub-tree (29; 74, 75) is stored in said data base (3) separately from said first search tree (25; 71) and said second search tree (27; 72, 73).

15. The navigation device (1) of any one of claims 10 to 14,

wherein each one of said first and second search trees (11, 12; 25, 27; 11, 62; 11, 62, 66; 71, 72; 81, 82) indicates next valid characters for country names, city names, street names, or names of points of interest (POI).

5

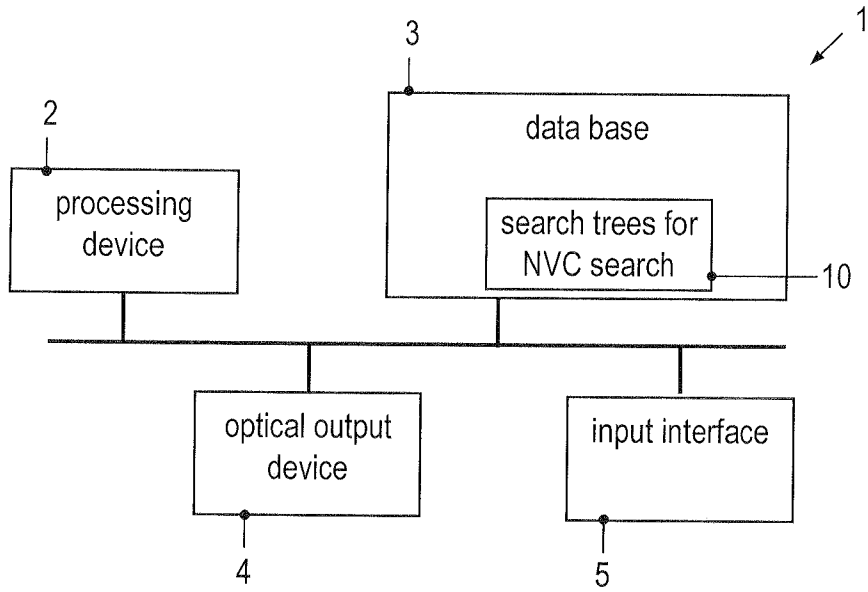
1
3
2
data base
processing device
search trees for NVC search
10
optical output device
input interface
4
5

Fig. 1

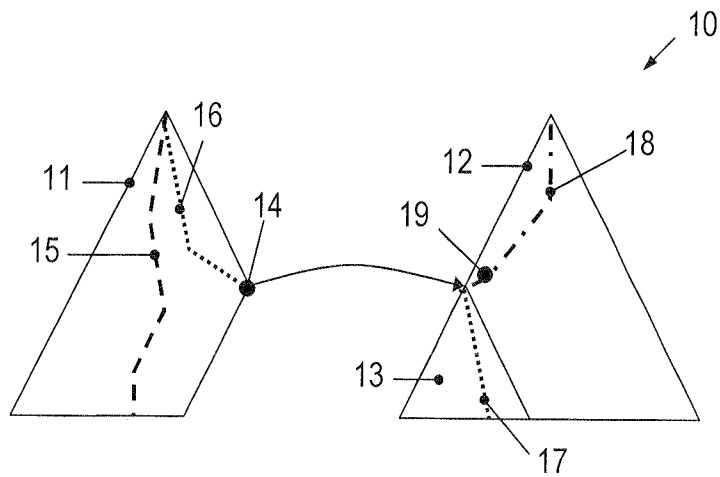
10
16
11
14
12
18
19
15
13
17

Fig. 2

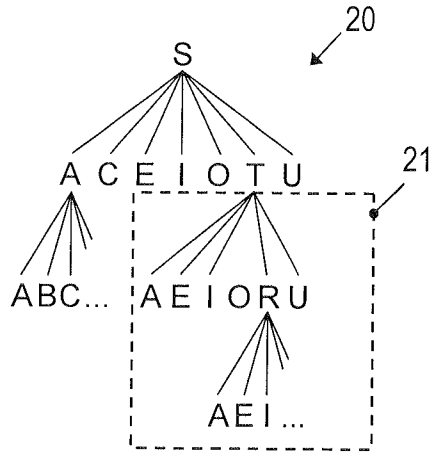
20
S
A C E I O T U
21
ABC...
A E I O R U
A E I ...

Fig. 3

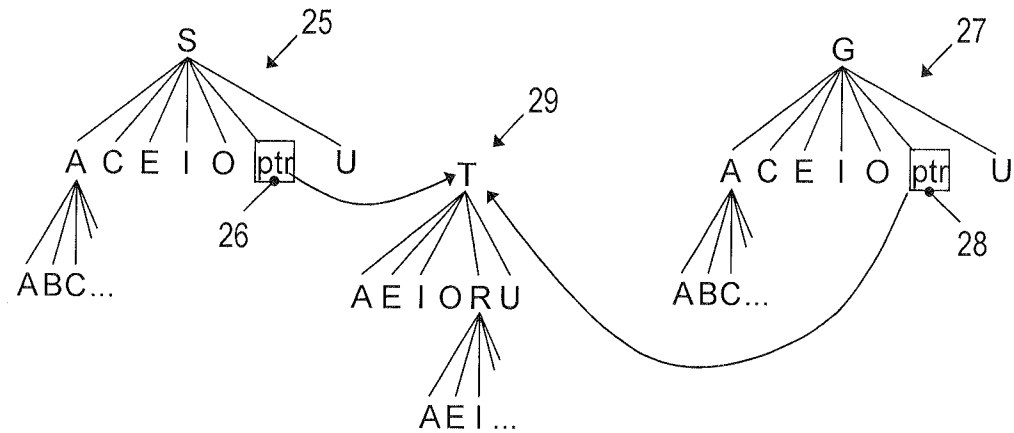
25
S
A C E I O ptr U
26
ABC...
29
T
A E I O R U
A E I ...
27
G
A C E I O ptr U
28
ABC...

Fig. 4

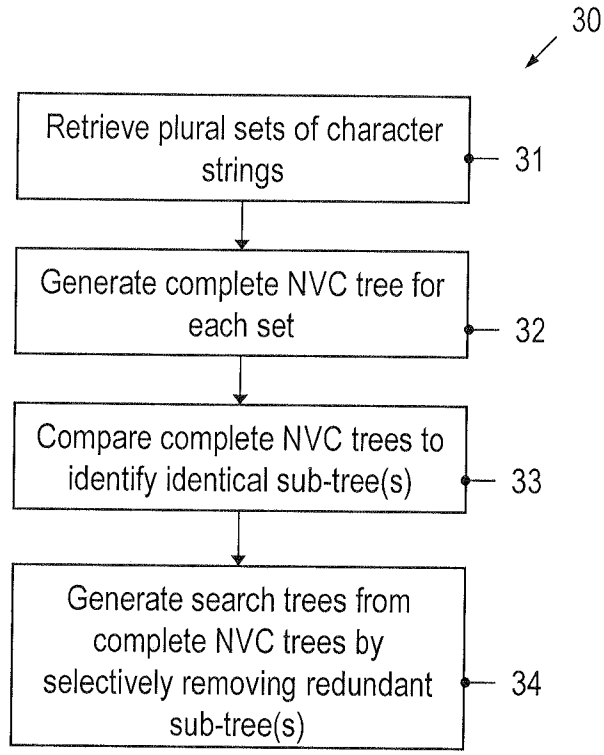
30
Retrieve plural sets of character strings
31
Generate complete NVC tree for each set
32
Compare complete NVC trees to identify identical sub-tree(s)
33
Generate search trees from complete NVC trees by selectively removing redundant sub-tree(s)
34

Fig. 5

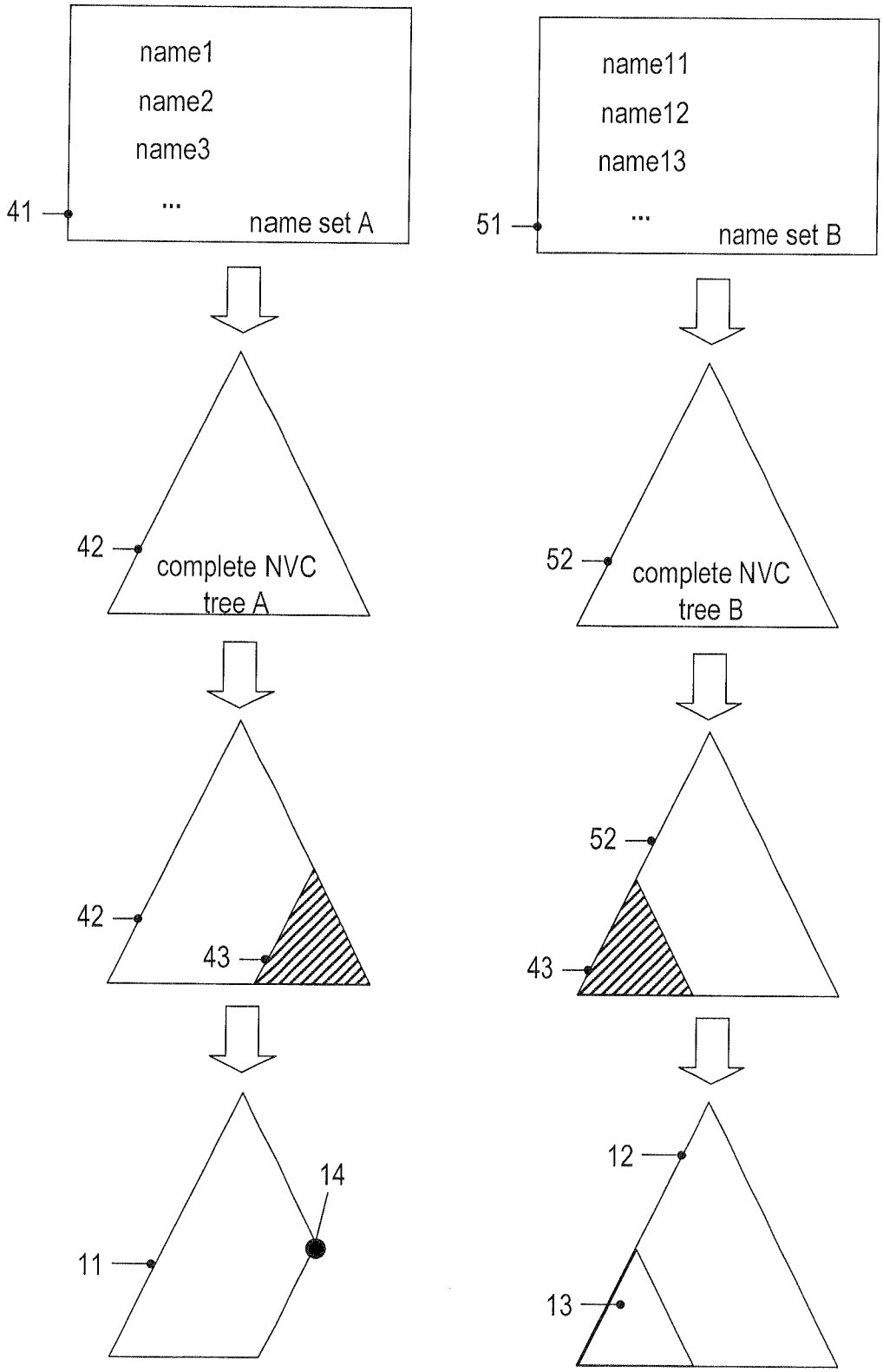
name1
name2
name3
...
41
name set A
name11
name12
name13
...
51
name set B
42
complete NVC tree A
52
complete NVC tree B
42
43
52
43
14
11
12
13

Fig. 6

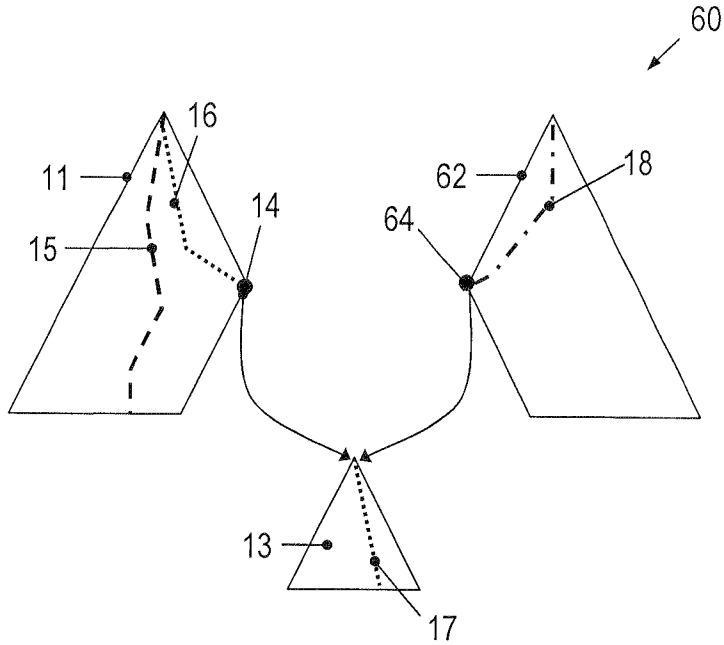

Fig. 7

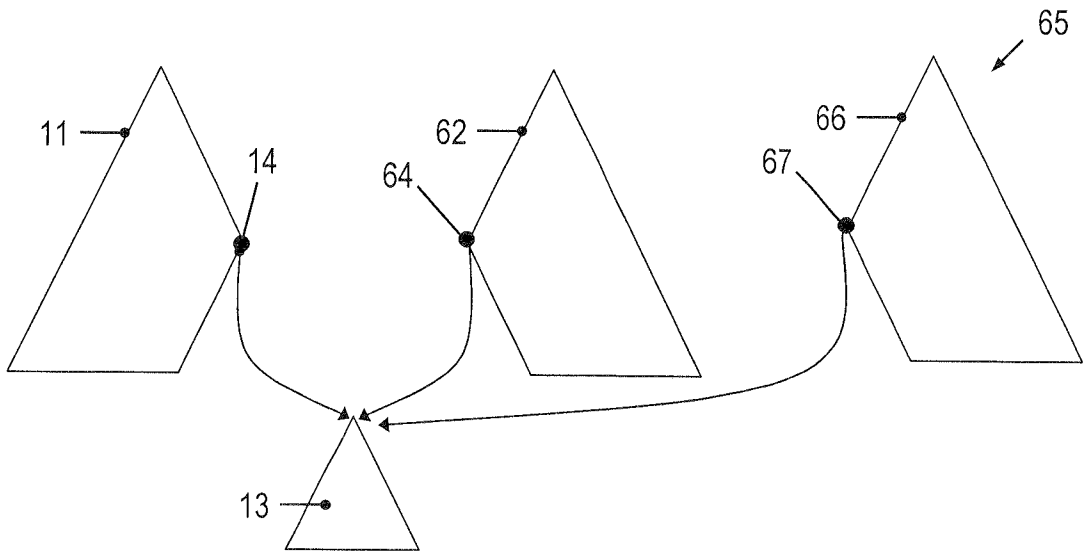

Fig. 8

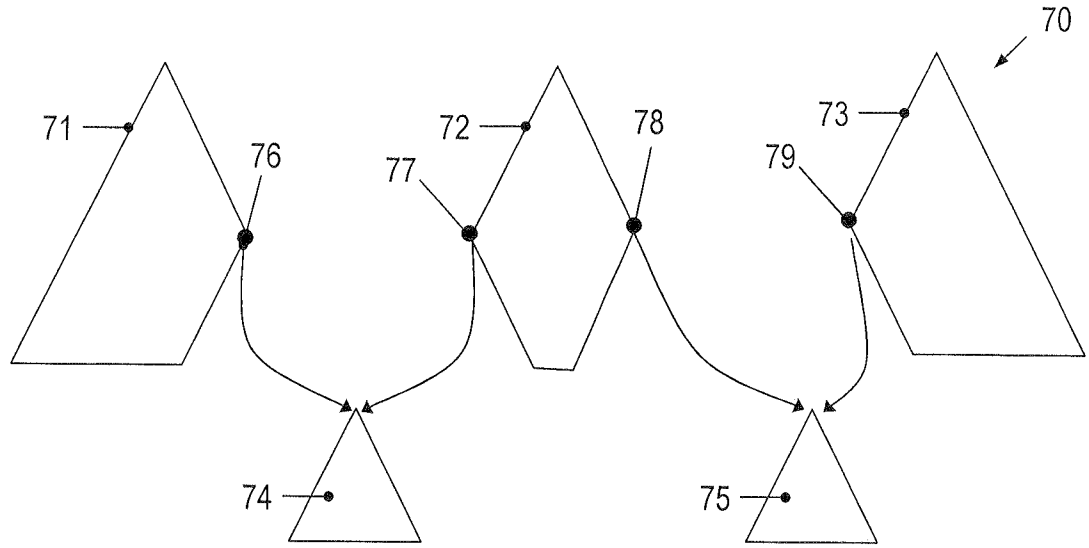

Fig. 9

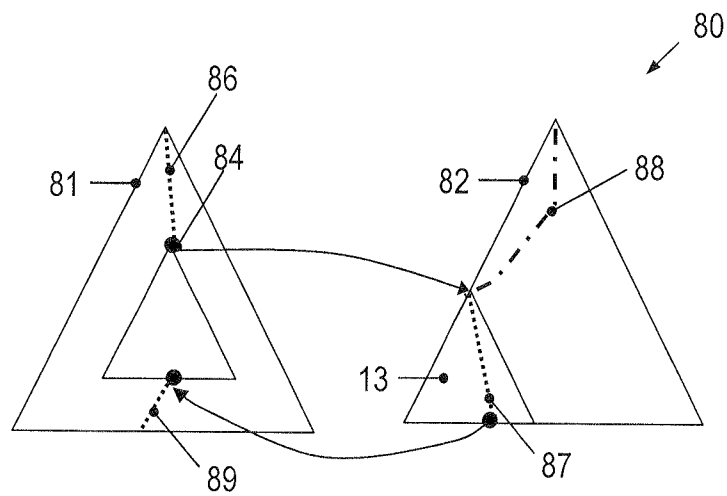

Fig. 10

INTERNATIONAL SEARCH REPORT

International application No
PCT/EP2012/069233

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F17/30 G06F3/023 G06F17/27
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)
G06F G01C

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 practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	EP 2 184 686 A1 (GUANGDONG GUOBI TECHNOLOGY CO [CN]) 12 May 2010 (2010-05-12) Reduced need for storage space.; paragraphs [0002] - [0005] -----	1-15
A	EP 1 455 163 A2 (BOSCH GMBH ROBERT [DE]) 8 September 2004 (2004-09-08) A vehicle navigation system destination input procedure selects similar sounding or spelt destinations from databank for user confirmation with spelling or word, acoustic or mechanical input in several categories including state, town, street or marked points; paragraph [0001] - paragraph [0031] ----- -/--	1-15

[X] Further documents are listed in the continuation of Box C.

[X] 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 application or patent 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	Date of mailing of the international search report
24 October 2012	02/11/2012

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Bowler, Alyssa

1

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	WO 2008/005808 A2 (TELE ATLAS NORTH AMERICA INC [US]; KUZNETSOV TSIA [US]; SANDLER ILYA M) 10 January 2008 (2008-01-10) Figure 8 shows two trees pointing to one object store. Figures 4a and 4b show how to give user an option of next valid character.; paragraph [0002] - paragraph [0062]; figure 4b	1-15
A	KOVACS L ET AL: "Implementation of sentence parser for Hungarian language in natural language processing", APPLIED MACHINE INTELLIGENCE AND INFORMATICS (SAMI), 2010 IEEE 8TH INTERNATIONAL SYMPOSIUM ON, IEEE, PISCATAWAY, NJ, USA, 28 January 2010 (2010-01-28), pages 59-63, XP031644206, ISBN: 978-1-4244-6422-7 Implementation of sentence parser for Hungarian language in natural language processing.; page 59, column 1 - page 60, column 1	1-15
A	WO 2008/106729 A1 (KANNUU PTY LTD [AU]; DINN KEVIN W [AU]) 12 September 2008 (2008-09-12) A display is generated which includes one or more parts of the information. Selection of the one or more parts of the information may be made and results in the generation of a display of further one or more parts for selection. The further one or more parts may be selected in order to add to the selected one or more parts to build a larger part or whole of words, sentences, messages, text, symbols and/or graphics.; the whole document	1-15
A	US 2004/021691 A1 (DOSTIE MARK [CA] ET AL) 5 February 2004 (2004-02-05) Character prediction visually informs user of which set of characters are most likely to have the character that the user needs next.	1-15

1

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/EP2012/069233

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 2184686 A1	12-05-2010	CN 101533403 A	16-09-2009
		EP 2184686 A1	12-05-2010
		US 2011208512 A1	25-08-2011
		WO 2010051674 A1	14-05-2010
EP 1455163 A2	08-09-2004	AT 443244 T	15-10-2009
		DE 10309948 A1	16-09-2004
		EP 1455163 A2	08-09-2004
WO 2008005808 A2	10-01-2008	NONE	
WO 2008106729 A1	12-09-2008	AU 2008222600 A1	12-09-2008
		CA 2717265 A1	12-09-2008
		EP 2135177 A1	23-12-2009
		US 2010153880 A1	17-06-2010
		WO 2008106729 A1	12-09-2008
US 2004021691 A1	05-02-2004	CA 2323856 A1	18-04-2002
		US 2004021691 A1	05-02-2004
		US 2011010655 A1	13-01-2011