

(19) World Intellectual Property Organization
International Bureau



(43) International Publication Date
31 August 2006 (31.08.2006)

PCT

(10) International Publication Number
WO 2006/090404 A1

(51) International Patent Classification:
G06K 9/00 (2006.01)

(21) International Application Number:
PCT/IN2005/000059

(22) International Filing Date:
23 February 2005 (23.02.2005)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant (for all designated States except US):
HEWLETT-PACKARD DEVELOPMENT COMPANY, L.P. [US/US]; 20555 S.H. 249, Houston, TX 77070 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **BHASKARABHATLA, Ajay** [IN/IN]; 37-12-31, Nggo Colony, PR Gardens, Visakhapatnam, AP 530007 (IN). **MADHVANATH, Sriganesh** [IN/IN]; C-704 Raheja Residency, Koramangala 3rd Block, Bangalore 560034 (IN).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

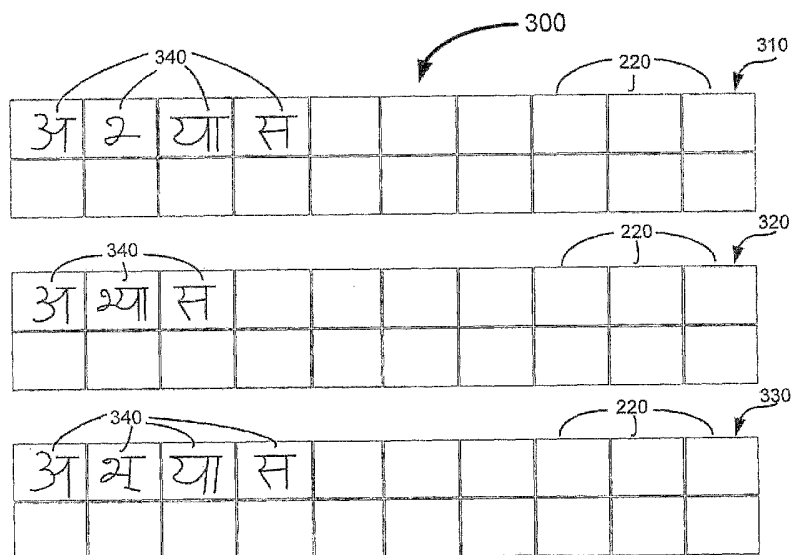
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— with international search report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: SYSTEM, METHOD, AND APPARATUS FOR ACCOMODATING VARIABILITY IN CHUNKING THE SUB-WORD UNITS OF ONLINE HANDWRITING



(57) Abstract: A technique that provides automatic real-time segmentation of an ink stream that does not require learning any chunking methodology, style of writing, and/or a predefined symbol set. In one example embodiment, this is achieved by drawing one or more strokes associated with a desired word of a script in one or more boxes provided on a digitizer screen using a pen. A handwriting recognition engine dynamically segments the one or more drawn strokes using a grammar net to form one or more candidate words upon a pen-up event. The grammar net comprises an inter-symbol grammar network that defines legitimate classes or combinations that can be formed using the characters in an associated script. The formed one or more candidate words are then displayed.

WO 2006/090404 A1

SYSTEM, METHOD, AND APPARATUS FOR ACCOMODATING VARIABILITY IN CHUNKING THE SUB-WORD UNITS OF ONLINE HANDWRITING

5 **Technical Field of the Invention**

The present invention relates generally to a user interface in a computer system, and more particularly relates to a system, method, and apparatus for accommodating variability in chunking of the sub-word units in the handwritten input to facilitate recognizing and/or repurposing handwriting.

10

Background of the Invention

Pen-based computers in which the pen replaces the keyboard and/or mouse as the primary input device have become well known. Such systems provide a digitizing pad and pen for the user to input information. The information is input in the form of strokes or
15 graphic representations of strokes. The computer monitors the movement of the pen and creates a stroke when the pen comes within a predefined proximity of or in contact with the digitizing pad. The computer follows the movement of the pen across the digitizing pad while simultaneously extending the stroke, and ends the stroke when the pen is removed or moved out of proximity from the digitizing pad.

20 Various methods to improve recognizing handwriting are well known. One prior art approach is to provide a series of boxes in the input area (which usually is the display area in a digitizer) for entering character information. These systems use the boxes for entry of text in an attempt to improve accuracy of recognition and to separate one character from the next. In these systems, an array of boxes is displayed and the user writes one character
25 in each box. Although the boxes aid in improving the accuracy of recognition by automatically segmenting digital ink into manageable units like characters, most people find it awkward to write in these boxes because it confines the writer to a specific region. In addition, such systems require learning a new scheme (for example, graffiti). Further, Indic scripts, such as Devanagari, are not amenable to writing as a sequence of characters
30 and hence are not feasible to be written in such boxes. They generally lack a well-accepted

definition for characters. Consequently, grouping of strokes into characters in boxed online handwriting user interfaces may be inconsistent.

Another character recognition system provides a free form interface, in which the user writes text without boxes. In these systems, after a user inputs several ink characters, special algorithms are used to separate the ink strokes to form characters and then
5 recognize each formed ink character. However, these systems are generally difficult to use for languages that include characters which overlap and are not spatially separated. Using these systems for such languages can lead to significant recognition errors.

In addition, the free form handwriting entry and recognition system requires the
10 system to be able to recognize greater stroke order variability and style variations apart from the complexity of pattern shapes. Furthermore, for Indic scripts, such as Devanagari, the state of the art in handwriting recognition does not currently deal with word level recognition as much of the work is still focused on improving the accuracy of the isolated symbols. Hence this necessitates usage of a boxed interface for such scripts.

Moreover, when writing in languages, such as Indic scripts, different users can
15 chunk strokes in various ways, which can result in requiring a large number of boxes for any given word. Therefore, for such scripts the definition of what goes into a box and consequently the set of symbols to be recognizable can be variable. For example, Devanagari script has nearly 20,000 aksharas or characters. Generally, it is possible to
20 constrain the writing style of users, for chunking patterns when using the free form interface, to about 450 pattern classes by restricting the users to split the consonant-consonant-vowel combinations (also known as "conjuncts") into two separate boxes. However, this requires the users to learn split conjuncts and special symbols. In addition, this process can require more writing than what is normally required in natural mode of
25 writing. Further, the users have to remember nearly 450 pattern classes/shapes. Hence, using the free form technique can result in significantly lower recognition accuracy.

Another approach that uses boxes as well as constrained styles for PDAs as word level systems require dictionaries that can increase the footprint of the recognition system. Further, this approach can require significant memory and processing power.

Summary of the Invention

5 According to an aspect of the subject matter, there is provided a method for real-time segmenting of a drawn ink stream, the method including the steps of drawing one or more strokes associated with a desired word of a script in one or more boxes on a digitizer screen using a pen, dynamically segmenting the one or more drawn strokes using a grammar net to form one or more candidate words upon a pen-up event by a handwriting
10 recognition engine, wherein the grammar net comprises an inter-symbol grammar network that defines legitimate classes or combinations that can be formed using the characters in the script, and displaying the formed one or more candidate words.

Brief Description of the Drawings

15 Embodiments of the invention will now be described, by way of example only, with reference to the accompanying drawings in which:

FIG. 1 is a flowchart illustrating an example method for accommodating variability in chunking of sub-word units of online handwriting according to an embodiment of the present subject matter.

20 FIG. 2 shows a screenshot of an input panel including boxes that allows writing one or more strokes associated with a desired word in multiple boxes according to an embodiment of the present subject matter.

FIGS. 3-5 show screenshots of variations in users' chunking strokes for any given word in an Indic script.

25 FIG. 6 shows a schematic diagram of a grammar net formed for a set of grammar classes for Indic scripts according to an embodiment of the present invention.

FIG. 7 shows a grid level array formed according to an embodiment of the present invention.

FIG. 8 is a block diagram of a typical computer system used for implementing embodiments of the present subject matter, such as those shown in FIGS. 1-7.

Detailed Description of the Invention

5 In the following detailed description of the various embodiments of the invention, reference is made to the accompanying drawings that form a part hereof, and in which are shown by way of illustration specific embodiments in which the invention may be practiced. These embodiments are described in sufficient detail to enable those skilled in the art to practice the invention, and it is to be understood that other embodiments may be
10 utilized and that changes may be made without departing from the scope of the present invention. The following detailed description is, therefore, not to be taken in a limiting sense, and the scope of the present invention is defined only by the appended claims.

Terminology

15 The term "sub-word unit" in Devanagari script refers to an "akshara" or "samyukthakshara". The term "sub-word unit" refers to a member of alphabetic, legographic, and/or phonetic/syllabic character set, which includes syllables, alphabets, numerals, punctuation marks, consonants, consonant modifiers, vowels, vowel modifiers, and special characters, and/or any combination thereof. The choice of the sub-word unit
20 can depend on the language itself. For example, languages with a very well defined and small number of syllables may benefit from a syllable-sized unit.

The term "ink stream" refers to a sequence of drawn strokes. The ink stream is generated by a digitizer in a sequence of traces. The term "segmentation" refers to a process of identifying marks in the sequence of drawn strokes such that the collection of
25 strokes separated by these marks share a certain common property, such as (1) the collection of strokes can be interpreted as a work in a certain script, (2) the collection of strokes can represent a set of drawings, and so on.

The term "digitizer" refers to an input device for obtaining graphics input. It is connected to a computer and includes some form of a pen-like pointing device called a transducer. The digitizer has a flat surface with a defined space, or active area, where the user can place a gesture character map of a desired phonetic script; the map can be an
5 overlay. A pointing device is then used to pinpoint areas within the map, and the tablet sends associate character and/or modifier data to the handwriting recognition engine. The flat surface along with the pen-like pointing device provides a more natural drawing motion for users to input data into a computer.

In the specification and claims herein, the word "pen" is used. When used in the
10 specification and claims, this word shall mean a position-designating device that is sensed by the input tablet or touchpad on the digitizer. For a standard graphics tablet this is a special stylus, while for a standard capacitive touchpad, this is a finger. The special stylus can be used to draw or trace images, which are then converted to digital data that can be processed by the computer.

15 In addition, the language "candidate sub-words" refers to one or more sub-words that substantially closely match the drawn one or more strokes in a current box that are selected from all the sub-words associated with a script. The language "pattern classes" refers to a list of chosen and/or pre-determined sub-word units of a script.

Further in the specification and the claims herein, the term "grammar status" refers
20 to a connectivity rule between candidate words, i.e., it refers to a terminating node of a candidate sub-word in the grammar net. The grammar status contains the information about a grammar class that it can precede and/or succeed. At a grid level, the grammar status is used to generate valid word paths.

The present subject matter provides an inter-symbol grammar network that defines
25 legitimate classes/combinations (a set of rules) to form aksharas or characters to improve reliability in terms of chunking. This technique provides a mechanism for variability in sub-word level chunking when inputting handwritten ink stream in handwriting recognition schemes.

FIG. 1 illustrates an example method 100 for accommodating variability in chunking of sub-word units during online handwriting. At step 110, this example method 100 begins by drawing one or more strokes associated with a desired word of a script in one or more boxes provided on a digitizer screen using a pen.

5 Referring now to FIG. 2, there is shown an example screenshot 200 of one or more drawn strokes 210 of an Indic script in multiple boxes 220 provided on a digitizer screen.

Referring now to FIG. 3, there are shown example screenshots 310 - 330 of variations in user's chunking the one or more drawn strokes 340 of a desired word in an Indic script in one or more boxes 220. It can be seen in FIG. 3 that a user chunks the
10 desired word by drawing the one or more strokes 340 in the one or more boxes 220 in three different ways 310, 320, and 330. Similarly, it can be seen in FIG. 4 that a user chunks another desired word in multiple ways, as shown in screenshots 410, 420, and 430, when drawing the desired word using the one or more strokes 440 in the multiple boxes 220. Similarly, FIG. 5 illustrates another example of a user chunking another desired word, i.e.,
15 in Indic script, in multiple ways, as shown in screenshots 510, 520, and 530, when drawing the desired word using the one or more strokes 540 in the multiple boxes. It can be seen in FIGS. 3-5, that a user can write a conjunct, i.e., a sub-word or a character of a desired word, using one or more strokes in one or more of boxes 220 as opposed to writing in one box. This is due to the differences in the styles of writing with different users. In addition,
20 certain scripts, such as Indic scripts allow users to write sub-word units of a word, i.e., one or more strokes of a word, in different ways in different boxes as described above with reference to FIGS. 3-5.

At step 115, a handwriting recognition engine then dynamically obtains one or more candidate sub-words associated with the one or more drawn strokes in the one or
25 more boxes 220 in the digitizer screen, which is responsive to the user drawing the one or more strokes, upon a pen-up event. The candidate sub-words can include a sub-word unit, a character, or a word. The sub-words associated with a script are generally stored in memory that is communicatively coupled to the digitizer screen.

In some embodiments, the pen-up event can include determining whether the one or more drawn strokes in a current box of a desired word is the last box. In other embodiments, the pen-up event can be based on monitoring a time associated with a pen movement away from the digitizer screen, i.e., from the one or more boxes. This can include monitoring the movement of the pen across the digitizer screen when simultaneously extending the stroke, and ends the stroke when the pen is removed or moved out of proximity from the digitizer screen.

It can be envisioned that all the ink strokes inputted into one or more boxes can be obtained at once upon a recognition event, such as the above-described pen-up event. The obtained ink strokes can be further divided into ink strokes corresponding to each box. Ink strokes corresponding to a continuous sequence of non-empty boxes of a grid input panel can be interpreted as an integral number of characters that form a grammatically valid string, i.e., it can be interpreted as a word. The non-empty box means a box including at least one drawn ink stroke.

At step 120, a grammar status associated with each of the obtained candidate sub-words is dynamically obtained using a grammar net. In some embodiments, the grammar status associated with each of the candidate sub-words is determined by performing principal component analysis (PCA) on each of the one or more candidate sub-words using a list of sub-word units. Generally, an input stream is segmented into smaller units, i.e., as sub-word units or characters, suitable for direct classification either as part of a divide-and-conquer strategy, or using constraints imposed by the application, for example, by restricting the hand-writing to within boxes. PCA is one such scheme that is used to find basis vectors of each class subspace and the orthogonal distance to the subspaces used for classification. Pre-clustering of the training data and modifications of distance measure are explored to overcome some common problems in the traditional subspace method.

The following table illustrates forming a smaller set of symbols, i.e., about 120 pattern classes for Devanagari Script and an inter-symbol grammar that defines legitimate combinations of classes to form aksharas or characters of the Devanagari script. The table

below illustrates separation of the 120 pattern classes, i.e., symbols into a pre-determined number of grammar classes, such as vowels, consonants, half-consonants, matras, vowel modifiers, halant, end-of-character, and so on.

5

Grammar Class # 1	Grammar Class # 2	Grammar Class # 3	Grammar Class # 4	Grammar Class # 5	Grammar Class # 6	Grammar Class # 7	Grammar Class # 8
Vowel	Consonant	Half-consonant	Special character	Vowel Modifier	Matra	Halant	End of character
4-8	19-51	69-93	97-120	1-3, 61,65,67	52-60, 62-64, 94-96	66	-

The table below illustrates the class-labels used in forming the above smaller set of symbols:

0.	-	16.	ओ	32.	ढ	48.	श
1.	ँ	17.	औ	33.	ण	49.	ष
2.	ं	18.	ऑ	34.	त	50.	स
3.	ं	19.	क	35.	थ	51.	ह
4.	अ	20.	ख	36.	द	52.	ा
5.	आ	21.	ग	37.	ध	53.	ि
6.	इ	22.	घ	38.	न	54.	ी
7.	ई	23.	ङ	39.	प	55.	ु
8.	उ	24.	च	40.	फ	56.	ू
9.	ऊ	25.	छ	41.	ब	57.	ृ
10.	ऋ	26.	ज	42.	भ	58.	ै
11.	ऐ	27.	झ	43.	म	59.	े
12.	ए	28.	ञ	44.	य	60.	ै
13.	ऐ	29.	ट	45.	र	61.	ँ
14.	ऐ	30.	ठ	46.	ल	62.	ो
15.	ओ	31.	ड	47.	व	63.	ो

64. ौ	79. थ्	94. क्र	109. न्न
65. ॉ	80. ध्	95. ट्र	110. ह्य
66. ्	81. न्	96. क्	111. ह्य
67. ः	82. प्	97. क्ष	112. ह्
68. ।	83. फ्	98. त्र	113. ज्य
69. क्	84. ब्	99. झ	114. ज्ञ
70. ख्	85. भ्	100. श्र	115. ज्ञ
71. ग्	86. म्	101. क्त	116. ट्ट
72. घ्	87. य्	102. द्	117. ट्ट
73. च्	88. ल्	103. ख्	118. ट्ट
74. ज्	89. व्	104. द्र	119. ड्ड
75. झ्	90. श्	105. घ्	120. ड्ड
76. ज्ञ्	91. ष्	106. द्म	
77. ण्	92. स्	107. ह्	
78. त्	93. ह्	108. स्त्र	

Referring now to FIG. 6, there is illustrates a grammar net 600 formed using the set of symbols shown in the above table. The grammar net 600 shown in FIG. 6 dictates valid combinations of symbols that can be used to form a character. As shown in FIG. 6, the grammar net 600 is formed using a set of grammar nodes, labeled 0, 1, 2, ...20, and a pre-

determined number of grammar classes which are labeled as V, C, HC, M, VM, H, E and SC, which refers to vowels, consonants, half-consonants, matras, vowel modifiers, halants, end-of-characters or start nodes, and special characters, respectively. Any path in the grammar net 600 that starts from a node having a grammar class E and ends in another node having a grammar class E will define a character. Ink inputted into a box, i.e., one or more strokes inputted into a box, is interpreted as a combination of these symbols to form complete or incomplete character(s) and character completeness information is carried forward into the next box. The combination of the strokes drawn in substantially adjacent boxes is constrained by the grammar net 600 as shown in FIG. 6. For example, the grammar net invalidates combinations of adjacent boxes that can lead to invalid combinations of pattern classes that follow each other.

In some embodiments, the probability of pattern matching associated with each obtained candidate sub-word is also dynamically determined along with the grammar status. The probability of pattern matching can be obtained using any well-known pattern matching technique.

Referring back to FIG.1, at step 125, a grid level array is formed using each candidate sub-word and the associated grammar status for the current box. At step 130, a determination is made as to whether the current box is the last box among the one or more boxes that include the one or more strokes drawn by the user. Based on the determination at step 130, the method 100 goes to step 115 and repeats the above-described steps 115-130 (for a stroke drawn in a next box) if the current box is not the last box. Based on the determination at step 130, the method 100 goes to step 135 if the current box is the last box. At step 135, one or more valid word paths are generated using the formed grid level array as a function of the grammar status associated with each candidate sub-word in the grid level array.

Referring now to FIG. 7, there is shown an example of a grid level array 700 formed using the above-described steps 110- 130. As shown in FIG. 7, the grid level array 700 comprises an (M x N) grid level array of candidate sub-words and their associated

grammar status. In some embodiments, the (M x N) grid level array 700 also comprises the associated probability of pattern matching associated with each candidate sub-word. The ink drawn, i.e., the one or more drawn strokes, in the one or more boxes 220 (shown in FIGS. 2-5) corresponds to the strokes drawn in boxes labeled Box 1 ink, Box 2 ink, ...Box N ink in the example grid level array 700. The (M x N) grid level array 700 shown in FIG. 7, comprises (M x N) candidate sub-words, their associated grammar status and the probabilities of pattern matching arranged in M number of rows and N number of columns. In the (M x N) grid level array N is equal to a number of boxes used in writing the desired word and M is equal to a highest number candidate sub-words obtained for one of the one or more strokes drawn in the one or more boxes.

In these embodiments, a valid word path is generated by choosing a current candidate sub-word located at (1 x 1) in the grid level array 700. An associated candidate sub-word located in a substantially next column (M x 2) in the grid level array 700 is then selected as a function of the grammar status associated with the chosen current candidate sub-word located in the column (1 x 1) and the associated candidate sub-words in the (M x 2) column. After selecting the candidate sub-word located in the substantially next column (M x 2), a determination is made as to whether there is another box substantially next to the column (M x 2) in the grid level array 700. The above-described steps are repeated if there is another box substantially next to the column (M x 2). A current valid word path is then generated using the chosen candidate sub-word in each of the selected associated candidate sub-words in the grid level array.

Referring now to FIG. 7, there is shown a line 710 in the grid level array 700 going from Box 1 ink and across to Box N ink indicating the pathway used to generate the valid word path. The pathway 710 is generated by choosing the candidate sub-word in each column by starting from the sub-word (M x 1) located in the first column to the sub-word (M x N) located in the last column as a function of the chosen starting candidate sub-word and subsequent candidate sub-words and their associated grammar status in subsequent columns in the grid level array 700.

In these embodiments, a determination is made as to whether there is another candidate sub-word ($M \times 1$) located in the first column for which a valid word path has to be generated. The above-described steps are repeated if there is another candidate sub-word ($M \times 1$) in the first column. The process of generating a valid word path is stopped if there are no other candidate sub-words left in the first column for generating a valid word path.

Applying the PCA classifier, such as the one shown in table above, to the ink in each of the boxes, Box 1 ink, Box 2 ink, ...Box N ink can result in following combination of valid word paths:

Box 1 -> Recognition lattice -> N best valid word paths for ink in Box 1 and its associated grammar status.

Box 2 -> Recognition lattice -> N best valid word paths for ink in Box 2 and its associated grammar status.

At step 140, a candidate word is formed using candidate sub-words associated with each generated valid word path. At step 150, each of the formed candidate words are displayed. At step 160, a sum of the probabilities of pattern matching of the one or more candidate sub-words associated with each valid word path is computed to obtain a probability of pattern occurrence for each candidate word. At step 170, a final candidate word from the displayed candidate words is selected as a function of the obtained probability of pattern occurrence by the handwriting recognition engine. In some embodiments, the candidate words are displayed on a digitizer screen and the final candidate word is selected using a pen. At step 180, the desired word is inputted into the handwriting recognition engine for handwriting recognition upon selecting one of the one or more displayed candidate words using the pen.

Although the flowchart 100 includes steps 110-180 that are arranged serially in the exemplary embodiments, other embodiments of the subject matter may execute two or more steps in parallel, using multiple processors or a single processor organized as two or more virtual machines or sub-processors. Moreover, still other embodiments may implement the steps as two or more specific interconnected hardware modules with related

control and data signals communicated between and through the modules, or as portions of an application-specific integrated circuit. Thus, the exemplary process flow diagrams are applicable to software, firmware, and/or hardware implementations.

Various embodiments of the present subject matter can be implemented in software,
5 which may be run in the environment shown in FIG. 8 (to be described below) or in any other suitable computing environment. The embodiments of the present subject matter are operable in a number of general-purpose or special-purpose computing environments. Some computing environments include personal computers, general-purpose computers, server computers, hand-held devices (including, but not limited to, telephones and personal digital
10 assistants (PDAs) of all types), laptop devices, multi-processors, microprocessors, set-top boxes, programmable consumer electronics, network computers, minicomputers, mainframe computers, distributed computing environments and the like to execute code stored on a computer-readable medium. The embodiments of the present subject matter may be implemented in part or in whole as machine-executable instructions, such as program
15 modules that are executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, and the like to perform particular tasks or to implement particular abstract data types. In a distributed computing environment, program modules may be located in local or remote storage devices.

FIG. 8 shows an example of a suitable computing system environment for
20 implementing embodiments of the present subject matter. FIG. 8 and the following discussion are intended to provide a brief, general description of a suitable computing environment in which certain embodiments of the inventive concepts contained herein may be implemented.

A general computing device, in the form of a computer 810, may include a
25 processing unit 802, memory 804, removable storage 801, and non-removable storage 814. Computer 810 additionally includes a bus 805 and a network interface (NI) 812.

Computer 810 may include or have access to a computing environment that includes one or more user input devices 816, one or more output devices 818, and one or more

communication connections 820 such as a network interface card or a USB connection. The one or more user input devices 816 can be a digitizer screen and a stylus and the like. The one or more output devices 818 can be a display device of computer, computer monitor, TV screen, plasma display, LCD display, digitizer screen, display on an electronic tablet, and
5 the like. The computer 810 may operate in a networked environment using the communication connection 820 to connect to one or more remote computers. A remote computer may include a personal computer, server, router, network PC, a peer device or other network node, and/or the like. The communication connection may include a Local Area Network (LAN), a Wide Area Network (WAN), and/or other networks.

10 The memory 804 may include volatile memory 806 and non-volatile memory 808. A variety of computer-readable media may be stored in and accessed from the memory elements of computer 810, such as volatile memory 806 and non-volatile memory 808, removable storage 801 and non-removable storage 814. Computer memory elements can include any suitable memory device(s) for storing data and machine-readable instructions,
15 such as read only memory (ROM), random access memory (RAM), erasable programmable read only memory (EPROM), electrically erasable programmable read only memory (EEPROM), hard drive, removable media drive for handling compact disks (CDs), digital video disks (DVDs), diskettes, magnetic tape cartridges, memory cards, Memory Sticks™, and the like; chemical storage; biological storage; and other types of data storage.

20 “Processor” or “processing unit,” as used herein, means any type of computational circuit, such as, but not limited to, a microprocessor, a microcontroller, a complex instruction set computing (CISC) microprocessor, a reduced instruction set computing (RISC) microprocessor, a very long instruction word (VLIW) microprocessor, explicitly parallel instruction computing (EPIC) microprocessor, a graphics processor, a digital signal
25 processor, or any other type of processor or processing circuit. The term also includes embedded controllers, such as generic or programmable logic devices or arrays, application specific integrated circuits, single-chip computers, smart cards, and the like.

Embodiments of the present subject matter may be implemented in conjunction with

program modules, including functions, procedures, data structures, application programs, etc., for performing tasks, or defining abstract data types or low-level hardware contexts.

Machine-readable instructions stored on any of the above-mentioned storage media are executable by the processing unit 802 of the computer 810. For example, a computer
5 program 825 may include machine-readable instructions capable of accommodating variability in chunking in a drawn an ink stream according to the teachings and herein described embodiments of the present subject matter. In one embodiment, the computer program 825 may be included on a CD-ROM and loaded from the CD-ROM to a hard drive in non-volatile memory 808. The machine-readable instructions cause the computer 810 to
10 encode according to the various embodiments of the present subject matter. As shown, the computer program 825 includes a handwriting recognition engine 830.

In operation, one or more strokes of a desired word are drawn in a current box is drawn using a user input device 816, which can include drawing the one or more strokes on a digitizer screen using a pen. The handwriting recognition engine 830 coupled to the
15 digitizer screen that is responsive to the one or more drawn strokes using the pen dynamically segments the one or more drawn strokes using a grammar net to form one or more candidate words upon a pen-up event. The grammar net is formed using a set of manageable valid combination of symbols that can form a character of a script. The output module 818 coupled to the handwriting recognition engine displays the formed one or more
20 candidate words upon the pen-up event. One of the displayed one or more candidate words is then inputted into the handwriting recognition engine by selecting the one of the displayed one or more words using the user input device 816 such as the pen.

In these embodiments, the handwriting recognition engine 830 dynamically obtains the one or more candidate sub-words associated with the one or more drawn strokes in a
25 current box upon the pen-up event from a set of sub-word units associated with the script stored in the memory 804. The handwriting recognition engine 830 then dynamically computes a grammar status associated with each determined candidate sub-word using the

grammar net. The handwriting recognition engine 830 forms a grid level array using each candidate sub-word and the associated grammar status for the current box.

Further in these embodiments, the handwriting recognition engine 830 then determines whether the one or more drawn strokes in the current box are the last one or
5 more strokes. The handwriting recognition engine 830 then dynamically obtains one or more candidate sub-words associated with the one or more drawn strokes in a next box upon the pen-up event from a set of sub-word units associated with the script if the one or more drawn strokes in the current box is not the last one or more strokes. The handwriting recognition engine 830 then dynamically computes a grammar status associated with each
10 determined candidate sub-word using the grammar net. The handwriting recognition engine 830 then updates the formed grid level array using each candidate sub-word and the associated grammar status for the next box.

Furthermore in these embodiments, the handwriting recognition engine 830 generates one or more valid word paths using the grid level array as a function of the
15 grammar status associated with each candidate sub-word in the grid level array if the one or more drawn strokes in the current box is the last one or more strokes. The handwriting recognition engines 830 then forms a candidate word using candidate sub-words associated with each valid word path. The handwriting recognition engine 830 then outputs each candidate word associated with each valid word path.

20 Also in these embodiments, the handwriting recognition engine 830 dynamically determines the one or more candidate sub-words and their associated grammar status and probabilities of pattern matching by performing PCA on the list of sub-word units. In some embodiments, the handwriting recognition engine 830 then computes a sum of the probabilities of pattern matching of the one or more candidate sub-words associated with
25 each valid word path to obtain probability of pattern occurrence. The handwriting recognition engine 830 then selects a final candidate word from the displayed candidate words as a function of the obtained probability of pattern occurrence.

The operation of the computer system 800 for triggering the handwriting recognition

is explained in more detail with reference to FIGS. 1-22.

The handwriting recognition engine of the present subject matter is modular and flexible in terms of usage in the form of a "Distributed Configurable Architecture". As a result, parts of the handwriting recognition engine may be placed at different points of a network, depending on the model chosen. For example, the handwriting recognition engine can be deployed in a server and the input and output data elements streamed over from a client to the server and back, respectively. The handwriting recognition engine can also be placed on each client, with the database management centralized. Such flexibility allows faster deployment to provide a cost effective solution to changing business needs.

Further, the above-described technique provides a smaller set of symbols as described-above for Indic script, such as Devanagari and an inter-symbol grammar that defines legitimate combination of classes to form aksharas or characters, which facilitates in accommodating with variations in chunking patterns and writing styles of different users. Hence, this process improves recognition accuracy during online handwriting. In addition, this scheme provides a mechanism for scripts like Devanagari to be written in boxes. Further, the technique allows variability in sub-word level chunking when using a handwriting recognition system. The above-described technique also allows combining one or more strokes when inputting into boxes provided on a digitizer screen. This process does not require the user to learn a certain chunking methodology, style of writing, or a predefined symbol thereby improving the handwriting recognition accuracy.

Although the embodiments shown in FIGS. 2-8 are for Devanagari script, these embodiments are merely illustrative. The above-described technique is not limited to the embodiments shown in FIGS. 2-8. Thus, the above-described technique can be used for any script.

The above technique can be implemented using an apparatus controlled by a processor where the processor is provided with instructions in the form of a computer program constituting an aspect of the above technique. Such a computer program may be stored in a storage medium as computer readable instructions so that the storage medium

constitutes a further aspect of the present subject matter.

The above-described technique provides various embodiments for accommodating variability in chunking during online handwriting. It is to be understood that the above-description is intended to be illustrative, and not restrictive. Many other embodiments will be apparent to those of skill in the art upon reviewing the above-description. The scope of the subject matter should, therefore, be determined with reference to the following claims, along with the full scope of equivalents to which such claims are entitled.

As shown herein, the present subject matter can be implemented in a number of different embodiments, including various methods, a circuit, an I/O device, a system, and an article comprising a machine-accessible medium having associated instructions.

Other embodiments will be readily apparent to those of ordinary skill in the art. The elements, algorithms, and sequence of operations can all be varied to suit particular requirements. The operations described-above with respect to the method illustrated in FIG. 1 can be performed in a different order from those shown and described herein.

FIGS. 1-8 are merely representational and are not drawn to scale. Certain proportions thereof may be exaggerated, while others may be minimized. FIGS. 1-8 illustrate various embodiments of the subject matter that can be understood and appropriately carried out by those of ordinary skill in the art.

It is emphasized that the Abstract is provided to comply with 37 C.F.R. § 1.72(b) requiring an Abstract that will allow the reader to quickly ascertain the nature and gist of the technical disclosure. It is submitted with the understanding that it will not be used to interpret or limit the scope or meaning of the claims.

In the foregoing detailed description of the embodiments of the invention, various features are grouped together in a single embodiment for the purpose of streamlining the disclosure. This method of disclosure is not to be interpreted as reflecting an intention that the claimed embodiments of the invention require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive invention lies in less than all features of a single disclosed embodiment. Thus the following claims are hereby

incorporated into the detailed description of the embodiments of the invention, with each claim standing on its own as a separate preferred embodiment.

CLAIMS:

1. A method comprising:
drawing one or more strokes associated with a desired word of a script in one or
5 more boxes on a digitizer screen using a pen;
dynamically segmenting the one or more drawn strokes using a grammar net to
form one or more candidate words upon a pen-up event by a handwriting recognition
engine, wherein the grammar net comprises an inter-symbol grammar network that
defines legitimate classes or combinations that can be formed using the characters in the
10 script; and
displaying the formed one or more candidate words.
2. The method of claim 1, further comprising:
inputting the desired word into the handwriting recognition engine by selecting
15 one of the one or more displayed candidate words using the pen.
3. The method of claim 1, wherein dynamically segmenting the one or more drawn
strokes using the grammar net comprises:
dynamically obtaining one or more candidate sub-words associated with the one
20 or more drawn strokes in a current box upon the pen-up event from a set of sub-word
units associated with the script;
dynamically computing a grammar status associated with each candidate sub-
word using the grammar net;
forming a grid level array using each candidate sub-word and the associated
25 grammar status for the current box;
determining whether the current box is the last box in the one or more boxes;
if not, then repeating the drawing, dynamically obtaining, dynamically
computing, forming, and determining steps for a next box;
if so, then generating one or more valid word paths using the formed grid level
30 array as a function of the grammar status associated with each candidate sub-word in the
grid level array;

forming a candidate word using candidate sub-words associated with each valid word path; and

displaying each candidate word associated with each valid word path.

5 4. The method of claim 3, wherein, in dynamically obtaining, the candidate sub-word comprises a sub-word unit, a character, or a word.

5. The method of claim 3, wherein dynamically obtaining the one or more candidate sub-words associated with the one or more drawn strokes comprises:

10 dynamically determining the one or more candidate sub-words and their grammar status and probabilities of pattern matching associated with the one or more drawn strokes by performing principal component analysis (PCA) on each of the one or more candidate sub-words using a list of standardized sub-word units, wherein the list standardized sub-word units are based on formed pattern classes.

15

6. The method of claim 5, further comprising:

computing a sum of the probabilities of pattern matching of the one or more candidate sub-words associated with each valid word path to obtain a probability of pattern occurrence; and

20 selecting a final candidate word from the displayed candidate words as a function of the obtained probability of pattern occurrence.

7. The method of claim 3, wherein determining whether the current box is the last box in the one or more boxes comprises:

25 monitoring a time associated with a pen movement away from the digitizer screen; and

determining whether the one or more drawn strokes in the current box are the last one or more strokes of the desired word as a function of the time associated with the pen movement away from the digitizer screen.

30

8. A method for real-time segmenting of a drawn ink stream comprising:

drawing one or more strokes associated with a desired word of a script in one or more boxes on a digitizer screen using a pen;

dynamically obtaining one or more candidate sub-words associated with the one or more drawn strokes in a current box upon the pen-up event from a set of sub-word units associated with the script;

dynamically computing a grammar status associated with each determined candidate sub-word using the grammar net;

forming a grid level array using each candidate sub-word and the associated grammar status for the current box, wherein, the grid level array comprises (M x N) candidate sub-words and their associated grammar status and probabilities of pattern matching arranged in M number of rows and N number of columns, wherein N is equal to a number of boxes used in writing the desired word and M is equal to a highest number candidate sub-words obtained for the one or more strokes drawn in the one or more boxes;

determining whether the one or more drawn strokes in the current box are the last one or more strokes;

if not, then repeating the drawing, dynamically obtaining, dynamically computing, forming, and determining steps for a next box;

if so, then generating one or more valid word paths using the formed grid level array as a function of the grammar status associated with each candidate sub-word in the grid level array;

forming a candidate word using candidate sub-words associated with each valid word path; and

displaying each candidate word associated with each valid word path.

25

9. The method of claim 8, wherein generating the one or more valid word paths using the formed grid level array comprises:

choosing a current candidate sub-word located at (1 x 1) in the grid level array;

selecting an associated candidate sub-word (M x 2) located in a column

substantially next to the sub-word located at (1 x 1) as a function of the grammar status

30

associated with the chosen current candidate sub-word located at (1 x 1) and the associated candidate sub-words located at (M x 2);

determining if there is another box substantially next to the column including the sub-word at (M x 2);

- 5 if so, then repeating the choosing, selecting, and determining steps above;
 if not, then generating a current valid word path using the chosen candidate sub-word and each of the selected associated candidate sub-words in the grid level array;
 determining if there is another candidate sub-word in the column including the sub-word located at (M x 1) for which a valid word path has to be generated;
10 if so, then repeating the choosing, selecting, and determining steps above to generate a next valid path; and
 if not, stopping the generation of the one or more valid word paths.

10. The method of claim 3, wherein the grammar classes are formed by separating the
15 characters in the script based on classes selected from the group consisting of vowels, consonants, half-consonants, matras, vowel modifiers, halant, and end-of-character.

11. An article comprising:
 a storage medium having instructions that, when executed by a computing
20 platform, result in execution of a method comprising:
 drawing one or more strokes associated with a desired word of a script in one or more boxes on a digitizer screen using a pen;
 dynamically segmenting the one or more drawn strokes using a grammar
25 net to form one or more candidate words upon a pen-up event by a handwriting recognition engine, wherein the grammar net comprises an inter-symbol grammar network that defines legitimate classes or combinations that can be formed using the characters in an associated script; and
 displaying the formed one or more candidate words.

12. The article of claim 11, further comprising:
inputting the desired word into the handwriting recognition engine by selecting one of the one or more displayed candidate words using the pen.
- 5 13 The article of claim 11, wherein dynamically segmenting the one or more drawn strokes using the grammar net comprises:
dynamically obtaining one or more candidate sub-words associated with the one or more drawn strokes in a current box upon the pen-up event from a set of sub-word units associated with the script;
10 dynamically computing a grammar status associated with each candidate sub-word using the grammar net;
forming a grid level array using each candidate sub-word and the associated grammar status for the current box;
determining whether the current box is the last box in the one or more boxes;
15 if not, then repeating the drawing, dynamically obtaining, dynamically computing, forming, and determining steps for a next box;
if so, then generating one or more valid word paths using the formed grid level array as a function of the grammar status associated with each candidate sub-word in the grid level array;
20 forming a candidate word using candidate sub-words associated with each valid word path; and
displaying each candidate word associated with each valid word path.
14. The article of claim 13, wherein, in dynamically obtaining, the candidate sub-
25 word comprises a sub-word unit, a character, or a word.
15. The article of claim 13, wherein dynamically obtaining the one or more candidate sub-words associated with the one or more drawn strokes comprises:
dynamically determining the one or more candidate sub-words and their grammar
30 status and probabilities of pattern matching associated with the one or more drawn strokes by performing principal component analysis (PCA) on each of the one or more

candidate sub-words using a list of standardized sub-word units, wherein the list standardized sub-word units are based on formed pattern classes.

16. The article of claim 15, further comprising:

- 5 computing a sum of the probabilities of the one or more candidate sub-words associated with each valid word path to obtain a probability of pattern occurrence; and
 selecting a final candidate word from the displayed candidate words as a function of the obtained probability of pattern occurrence.

10 17. The article of claim 13, wherein determining whether the current box is the last box in the one or more boxes comprises:

- monitoring a time associated with a pen-up event by the pen; and
 determining whether the one or more drawn strokes in the current box is the last one or more strokes of the desired word as a function of the time associated with the pen-
15 up event.

18. An apparatus for segmentation of inputted ink stream comprising:

- a digitizer screen and a pen for drawing sequentially one or more strokes of a desired word in a current box on the digitizer using the pen;
20 a handwriting recognition engine coupled to the digitizer screen that is responsive to the one or more drawn strokes using the pen, wherein the handwriting recognition engine dynamically segments the one or more drawn strokes using a grammar net to form one or more candidate words upon a pen-up event, wherein the grammar net is formed using a set of manageable valid combination of symbols that can form a character of a
25 script; and
 an output module coupled to the handwriting recognition engine that displays the formed one or more candidate words upon the pen-up event, and wherein one of the displayed one or more candidate words is inputted into the handwriting recognition engine by selecting one of the displayed one or more words using the pen.

30

19. The apparatus of claim 18, wherein the handwriting recognition engine dynamically obtains one or more candidate sub-words associated with the one or more drawn strokes in a current box upon the pen-up event from a set of sub-word units associated with the script,

- 5 wherein the handwriting recognition engine dynamically computes a grammar status associated with each determined candidate sub-word using the grammar net, and
 wherein the handwriting recognition engine forms a grid level array using each candidate sub-word and the associated grammar status for the current box.

- 10 20. The apparatus of claim 19, wherein the handwriting recognition engine determines whether the one or more drawn strokes in the current box are the last one or more strokes,

 wherein the handwriting recognition engine dynamically obtains one or more candidate sub-words associated with the one or more drawn strokes in a next box upon
15 the pen-up event from a set of sub-word units associated with the script if the one or more drawn strokes in the current box is not the last one or more strokes,

- wherein the handwriting recognition engine dynamically computes a grammar status associated with each determined candidate sub-word using the grammar net, and
 wherein the handwriting recognition engine updates the formed grid level array
20 using each candidate sub-word and the associated grammar status for the next box.

21. The apparatus of claim 20, wherein the handwriting recognition engine generates one or more valid word paths using the grid level array as a function of the grammar status associated with each candidate sub-word in the grid level array if the one or more
25 drawn strokes in the current box is the last one or more strokes,

 wherein the handwriting recognition engine forms a candidate word using candidate sub-words associated with each valid word path, and

 wherein the handwriting recognition engine outputs each candidate word associated with each valid word path.

22. The apparatus of claim 20, wherein the handwriting recognition engine dynamically determines the one or more candidate sub-words and their associated grammar status and probabilities of pattern matching by performing PCA on the list of sub-word units.

5

23. The apparatus of claim 22, wherein the handwriting recognition engine sums the probabilities of one or more candidate sub-words associated with each valid word path to obtain probability of pattern occurrence, and wherein the handwriting recognition engine selects a final candidate word from the displayed candidate words as a function of the
10 obtained probability of pattern occurrence.

24. A computer system for triggering handwriting recognition comprising:

a network interface;

a processing unit;

15 a memory coupled to the processing unit;

a user input device coupled to the network interface that receives the input data via the network interface, wherein one or more strokes of a desired sub-word unit is drawn using the user input device;

a handwriting recognition engine coupled to the user input device that is
20 responsive to the one or more drawn strokes, wherein the handwriting recognition engine dynamically segments the one or more drawn strokes using a grammar net to form one or more candidate words upon a pen-up event, wherein the grammar net is formed using a set of manageable valid combinations of symbols that can form a character of a script;
and

25 an output device coupled to the handwriting recognition engine that displays the formed one or more candidate words upon the pen-up event, and wherein one of the displayed one or more candidate words is inputted into the handwriting recognition engine by selecting one of the displayed one or more words using the pen.

30 25. The system of claim 24, wherein the handwriting recognition engine dynamically obtains one or more candidate sub-words associated with the one or more drawn strokes

in a current box upon the pen-up event from a set of sub-word units associated with the script,

wherein the handwriting recognition engine dynamically computes a grammar status associated with each determined candidate sub-word using the grammar net, and

5 wherein the handwriting recognition engine forms a grid level array using each candidate sub-word and the associated grammar status for the current box.

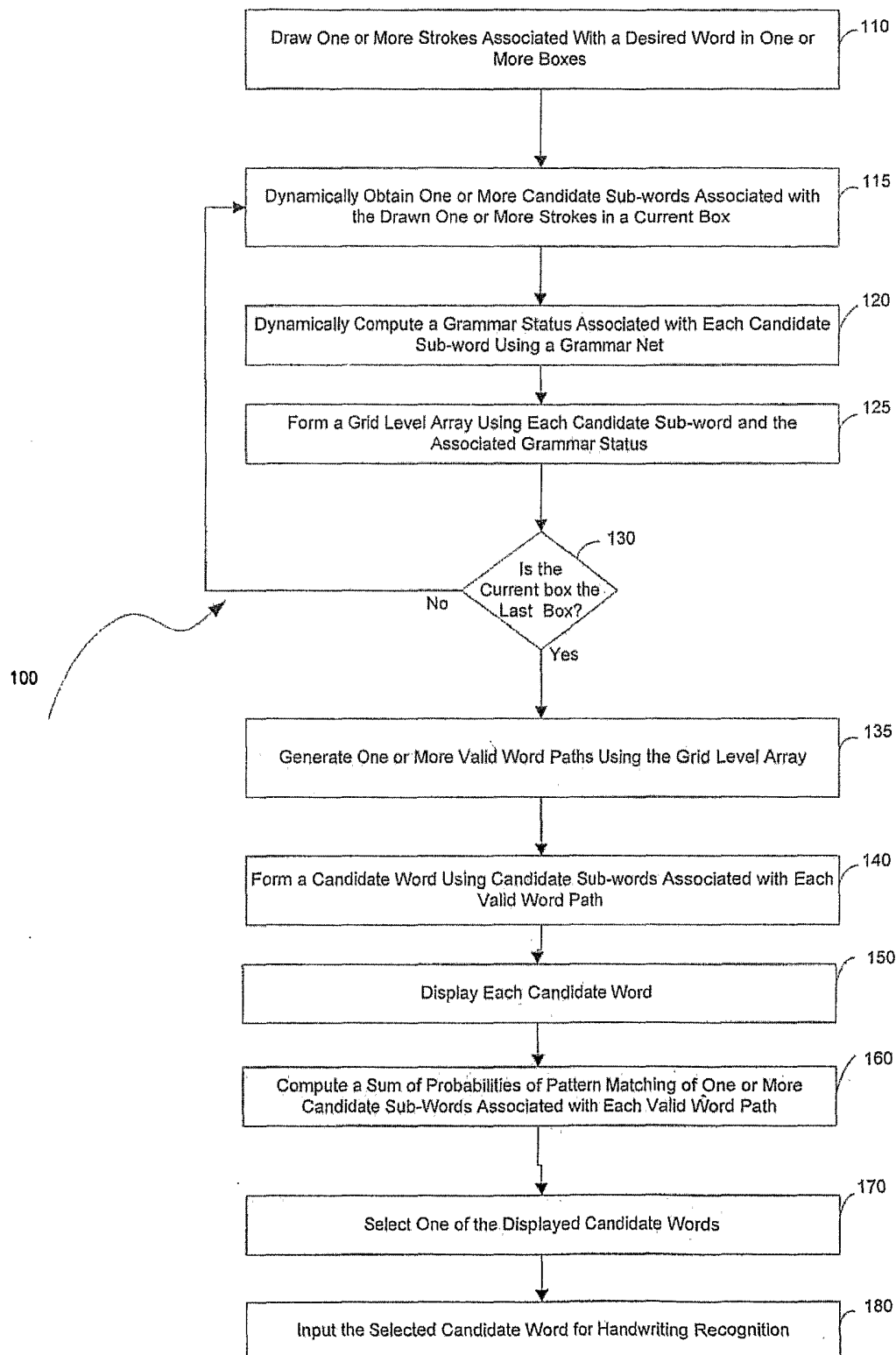


FIG. 1

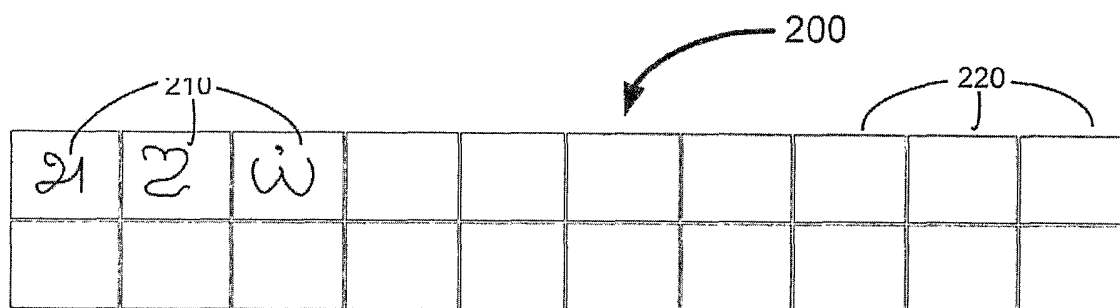


FIG. 2

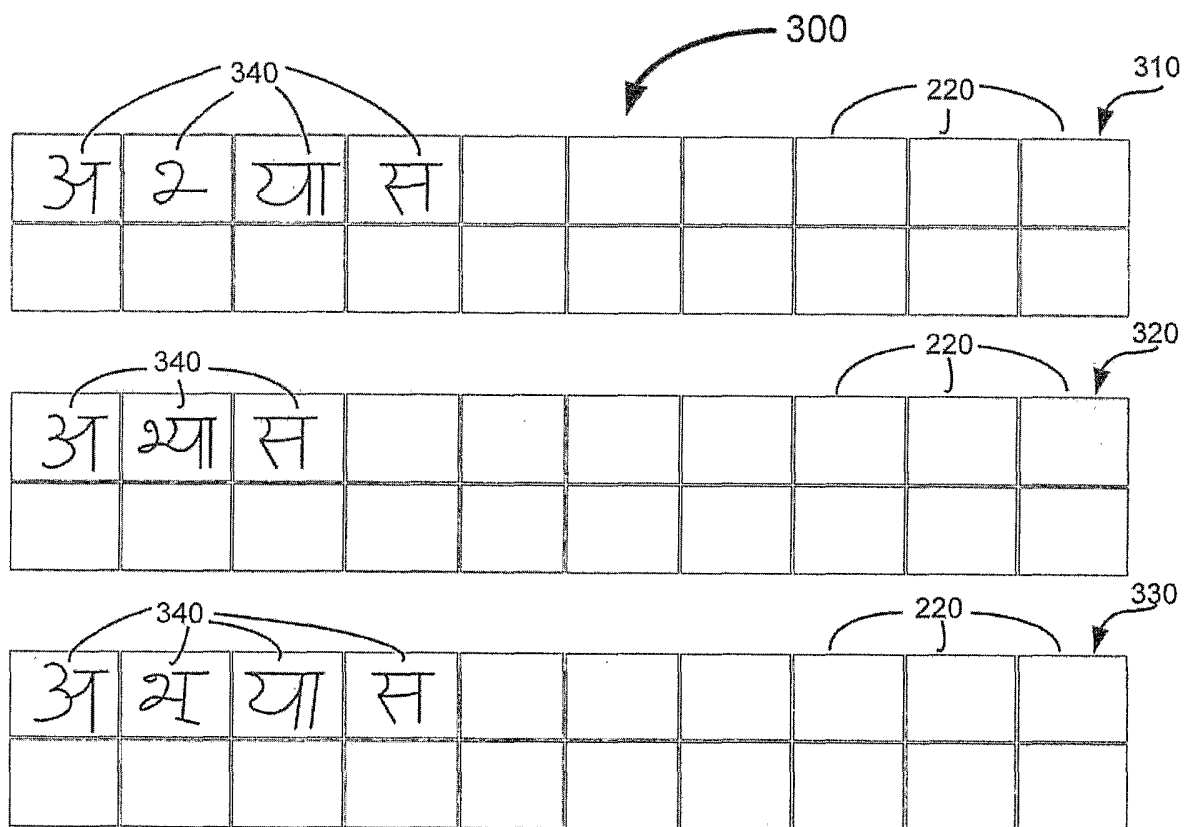
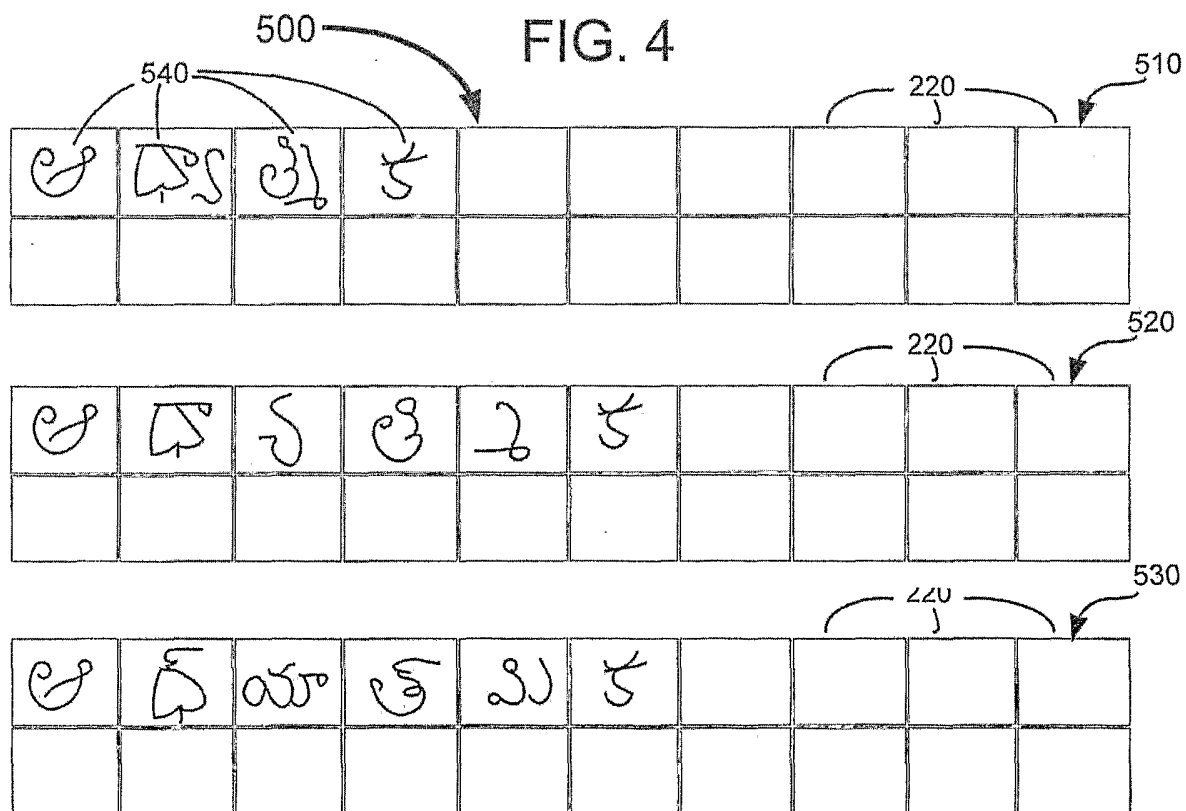
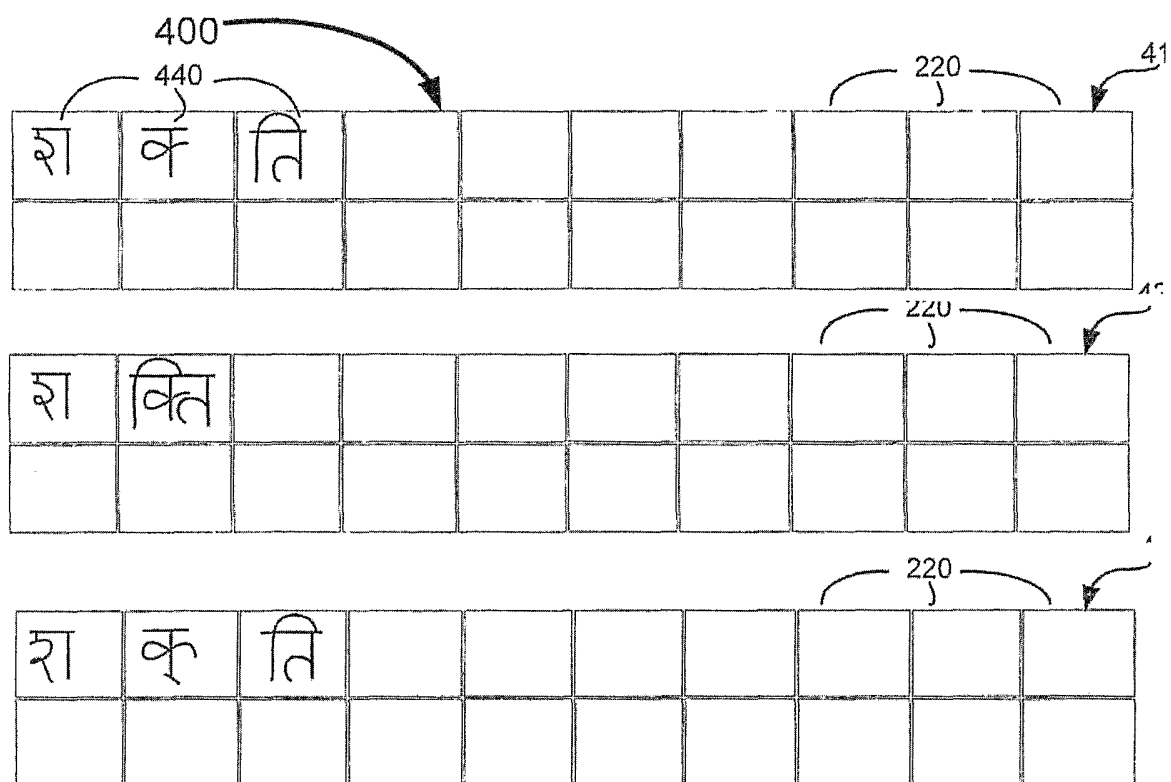


FIG. 3



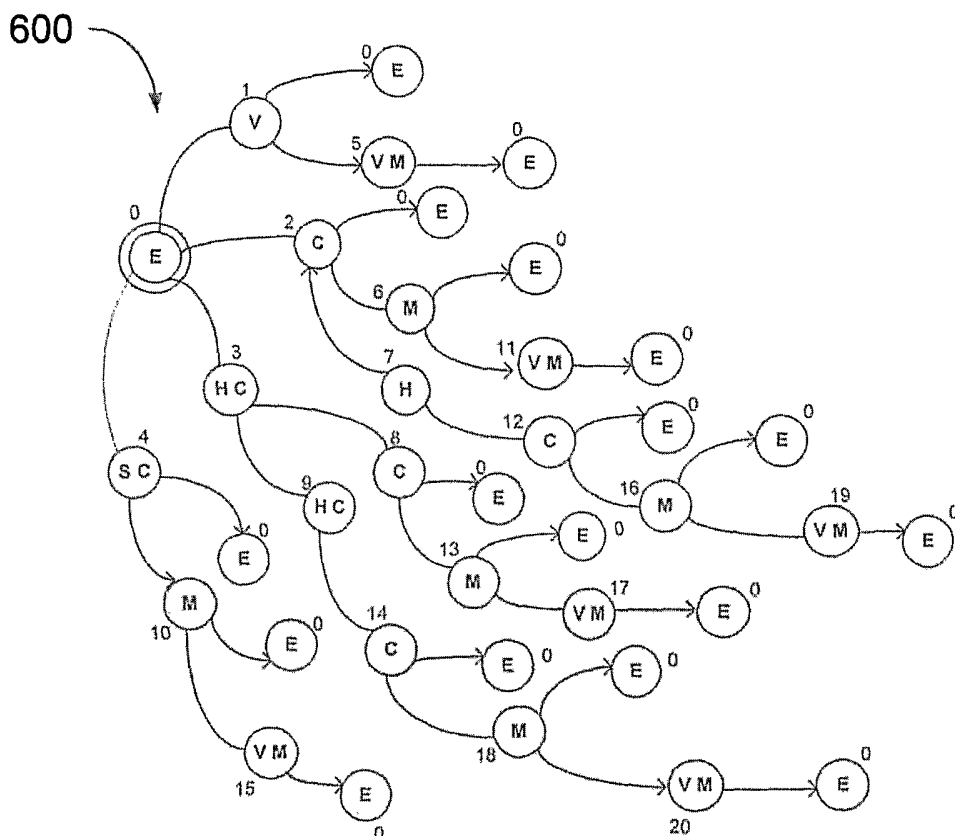


FIG. 6

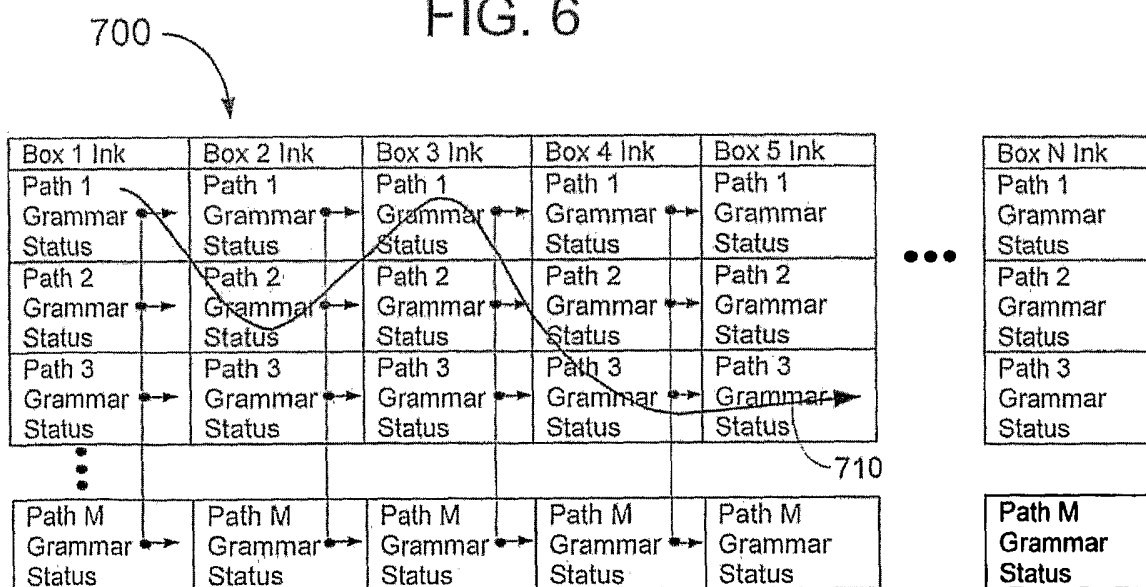


FIG. 7

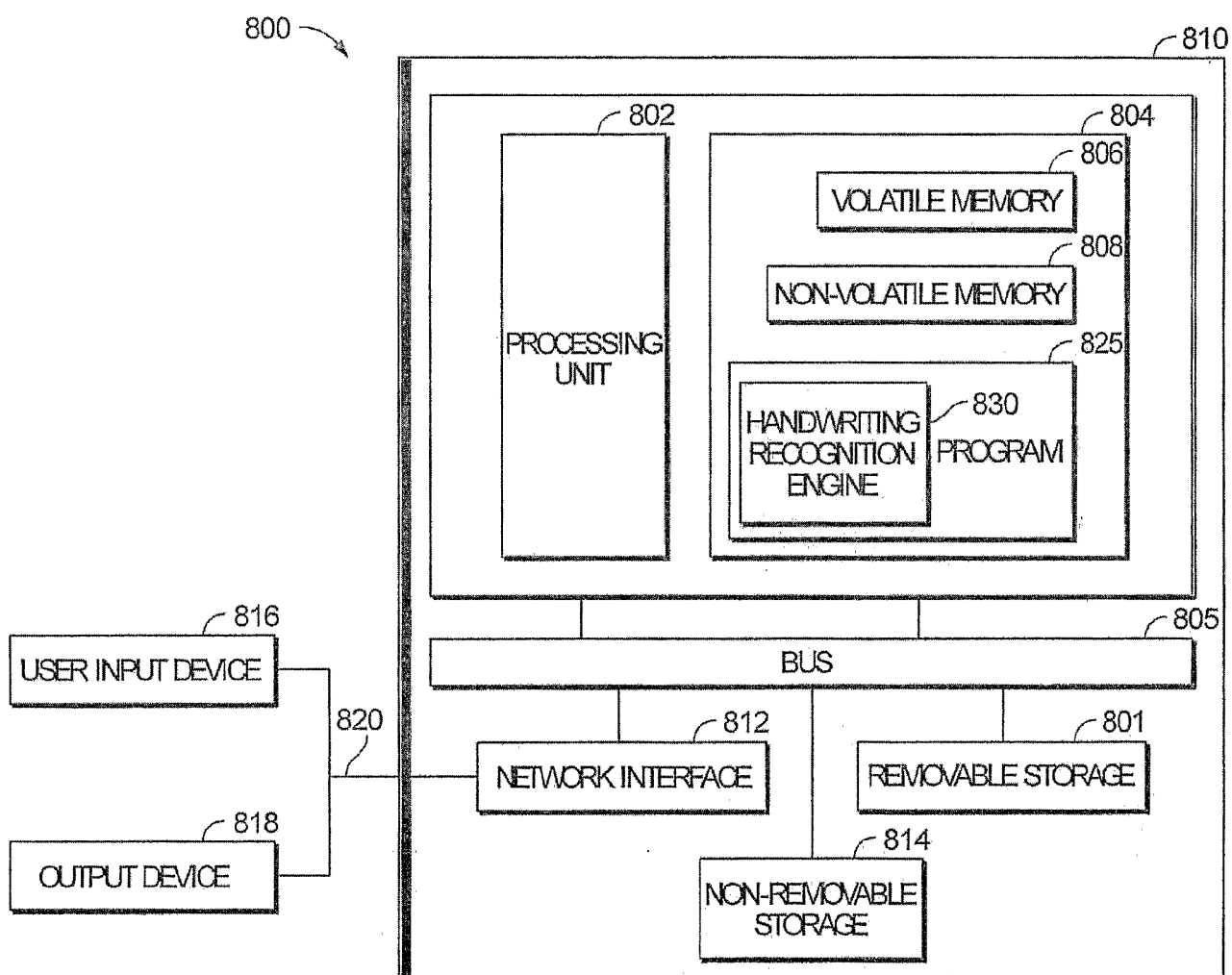


FIG. 8

INTERNATIONAL SEARCH REPORT

International Application No

PCT/IN2005/000059

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06K9/00

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)

IPC 7 G06K

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 6 560 360 B1 (NESKOVIC PREDRAG ET AL) 6 May 2003 (2003-05-06) abstract	1-25
X	US 6 418 239 B1 (HULLENDER GREGORY N ET AL) 9 July 2002 (2002-07-09) abstract	1-25
A	WO 03/025829 A (ANOTO AB; CRAVEN-BARTLE, THOMAS; PETTERSON, LARS; HOLLSTROEM, MAGNUS) 27 March 2003 (2003-03-27) abstract	7



Further documents are listed in the continuation of box C.



Patent family members are listed in 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

I 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 September 2005

Date of mailing of the international search report

05/10/2005

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

Neubüser, B

INTERNATIONAL SEARCH REPORT

International Application No
PCT/IN2005/000059

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
US 6560360	B1	06-05-2003	NONE	
US 6418239	B1	09-07-2002	AU 7722998 A WO 9855957 A1 US 6111985 A	21-12-1998 10-12-1998 29-08-2000
WO 03025829	A	27-03-2003	SE 520022 C2 SE 0103179 A	06-05-2003 22-03-2003