Title: A METHOD FOR CONVERTING KANA CHARACTERS TO KANJI CHARACTERS USING A PEN-TYPE STYLUS AND COMPUTER

Abstract

A method for performing Kana to Kanji conversion using a pen-based computer comprising the steps of: displaying a plurality of edit boxes; using the pen to input data; performing character recognition on the input data; simultaneously indicating a KKC operation and characters for the operation using the pen; determining the KKC operation to perform; retrieving the character string indicated; performing KKC on the string; replacing the existing characters with new characters, typically a Kanji equivalent; and displaying the converted character string including Kanji characters. This method makes KKC with pen-based systems much easier and efficient for the user, and therefore, provides pen-based systems with greater utility for input and output of Asian characters.
### FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

<table>
<thead>
<tr>
<th>AT</th>
<th>Austria</th>
<th>FR</th>
<th>France</th>
<th>MR</th>
<th>Mauritania</th>
</tr>
</thead>
<tbody>
<tr>
<td>AU</td>
<td>Australia</td>
<td>GA</td>
<td>Gabon</td>
<td>MW</td>
<td>Malawi</td>
</tr>
<tr>
<td>BB</td>
<td>Barbados</td>
<td>GB</td>
<td>United Kingdom</td>
<td>NE</td>
<td>Niger</td>
</tr>
<tr>
<td>BE</td>
<td>Belgium</td>
<td>GN</td>
<td>Guinea</td>
<td>NL</td>
<td>Netherlands</td>
</tr>
<tr>
<td>BF</td>
<td>Burkina Faso</td>
<td>GR</td>
<td>Greece</td>
<td>NO</td>
<td>Norway</td>
</tr>
<tr>
<td>BG</td>
<td>Bulgaria</td>
<td>HU</td>
<td>Hungary</td>
<td>NZ</td>
<td>New Zealand</td>
</tr>
<tr>
<td>BJ</td>
<td>Benin</td>
<td>IE</td>
<td>Ireland</td>
<td>PL</td>
<td>Poland</td>
</tr>
<tr>
<td>BR</td>
<td>Brazil</td>
<td>IT</td>
<td>Italy</td>
<td>PT</td>
<td>Portugal</td>
</tr>
<tr>
<td>BY</td>
<td>Belarus</td>
<td>JP</td>
<td>Japan</td>
<td>RO</td>
<td>Romania</td>
</tr>
<tr>
<td>CA</td>
<td>Canada</td>
<td>KP</td>
<td>Democratic People's Republic of Korea</td>
<td>RU</td>
<td>Russian Federation</td>
</tr>
<tr>
<td>CF</td>
<td>Central African Republic</td>
<td>KR</td>
<td>Republic of Korea</td>
<td>SD</td>
<td>Sudan</td>
</tr>
<tr>
<td>CG</td>
<td>Congo</td>
<td>KZ</td>
<td>Kazakhstan</td>
<td>SE</td>
<td>Sweden</td>
</tr>
<tr>
<td>CH</td>
<td>Switzerland</td>
<td>LIE</td>
<td>Liechtenstein</td>
<td>SI</td>
<td>Slovenia</td>
</tr>
<tr>
<td>CI</td>
<td>Côte d'Ivoire</td>
<td>LK</td>
<td>Sri Lanka</td>
<td>SK</td>
<td>Slovak Republic</td>
</tr>
<tr>
<td>CM</td>
<td>Cameroon</td>
<td>LU</td>
<td>Luxembourg</td>
<td>SN</td>
<td>Senegal</td>
</tr>
<tr>
<td>CN</td>
<td>China</td>
<td>LV</td>
<td>Latvia</td>
<td>TD</td>
<td>Chad</td>
</tr>
<tr>
<td>CS</td>
<td>Czechoslovakia</td>
<td>MC</td>
<td>Monaco</td>
<td>TG</td>
<td>Togo</td>
</tr>
<tr>
<td>CZ</td>
<td>Czech Republic</td>
<td>MG</td>
<td>Madagascar</td>
<td>UA</td>
<td>Ukraine</td>
</tr>
<tr>
<td>DE</td>
<td>Germany</td>
<td>ML</td>
<td>Mali</td>
<td>US</td>
<td>United States of America</td>
</tr>
<tr>
<td>DK</td>
<td>Denmark</td>
<td>MN</td>
<td>Mongolia</td>
<td>UZ</td>
<td>Uzbekistan</td>
</tr>
<tr>
<td>ES</td>
<td>Spain</td>
<td></td>
<td></td>
<td>VN</td>
<td>Viet Nam</td>
</tr>
<tr>
<td>FI</td>
<td>Finland</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A METHOD FOR CONVERTING KANA CHARACTERS TO KANJI CHARACTERS

USING A PEN-TYPE STYLUS AND COMPUTER

BACKGROUND OF THE INVENTION

1. Field of the Invention

The present invention relates generally to pen-based computers. In particular, the present invention relates to a method for converting information from its original input form which is most suitable for inputting information to an output form that is most suitable for outputting information to the user. Still more particularly, the present invention relates to a method for using a pen-based computer system to convert Kana and other characters which are optimal for input purposes to Kanji and other phrases that are optimal for data display and output purposes.

2. Description of the Related Art

Pen-based computers are now becoming more commercially available. One important aspect of pen-based computer systems is the ability to receive input from a pen-type stylus. The user simply writes or prints in his/her normal handwriting with the stylus on a digitizing pad or computer screen, and the computer stores an image of the strokes. With some systems, the strokes may be converted by applying handwriting recognition to image of the strokes. The character recognition process is possible with Latin based languages because there is a relatively small character set. However, there can be thousands of different characters with Asian languages. It is not sufficient to only perform character recognition for such a large character set because: the writer often inputs a portion of the character incorrectly, the system often fails to recognize very rare characters, or some characters are cumbersome and difficult to write. Therefore, especially for keyboard entry, computer systems working with Asian languages often provide an indirect entry method where a smaller set of characters (i.e., Kana) can be recognized and translated into characters in the larger set (i.e., Kanji). Once input and recognized, groups of Kana characters can then be translated into Kanji characters by
manipulation of the computer system. Typically, each Kanji or Kanji phrase represents several Kana characters.

Because of their newness, most pen-based systems do not provide any mechanism for converting Kana characters to Kanji characters. There have been attempts in the prior art to provide an efficient and usable system for Kana to Kanji Conversion (KKC) with standard keyboard-based systems. However, the approaches of the prior art are very cumbersome, and difficult to use. One approach is selection and operation. In this approach, a sub-string of text that has been input is selected. Then an operation is chosen, typically from a menu, that will operate on the selected text to perform the conversion. This approach requires too many keystrokes and selections which makes it cumbersome. Another approach provides a custom or dedicated area of the display for performing KKC. The dedicated window is also used for the input and output of Kanji characters. Unfortunately, using a dedicated window detracts from the usable space of the display available for other applications. This concern becomes even more heightened for portable systems where the display area is severely limited. The prior art also includes keyboard emulation and other gadgets, however, these devices also suffer from the shortcomings of being difficult to learn as well as requiring additional keystrokes.

KKC with existing pen-based systems is also problematic because character recognition is slower and more inaccurate than with keyboard-based systems. The prior art has attempted to address the shortcomings of character recognition by providing commands for having the computer generate guesses at the correct conversion. With such systems the user is typically forced to specify that KKC will be performed or select a mode of operation. Only then will the systems generate guesses using exact homophone matches or semantic equivalents of selected strings. However, the type of guesses generated often must also be identified which requires further input from the user. Additionally, the existing systems for correcting recognition are not sufficient because it is very difficult for pen users to detect errors in
character recognition, and the KKC competes with other applications for the attention of the user. Another problem in the art is the concurrent use of the keyboard and the pen or stylus. Typically, the user interfaces and their operating characteristics are very different for keyboards as compared with pens. Thus, it is difficult to switch between using the keyboard and the pen, or vice versa. When using the keyboard, strong modes are used to define the operations being performed. For example, some applications provide a different window, and there is no integrated use of the pen and keyboard. With front end processors, pen input is used to generate characters in the input stream, and the characters are distinct from any KKC processes. In other systems, the pen operates with a similar functionality to that of a "mouse" type controller which allows selection of buttons or other information on the display. Thus, none of the systems in the prior art effectively and cleanly integrate the use of the keyboard and the pen.

A further problem for systems that allow pen-based input is the resolution of spatial modality conflicts. Spatial modality refers to the interpretation of an event or input depending on the instance and/or location at which the event occurred. For example, in current pen-based systems a particular input can have two different meanings and effects based on the mode or where it is performed. The solution in the prior art has been to attach a single particular meaning to the input and prevent the use of the other meaning. This is problematic because it has the negative impact of a limiting the functionality associated with an input.

Therefore, there is need for a method of converting Kana characters to Kanji characters that is much simpler and automatic for the user. In particular, there is a need for a method that accommodates handwritten input and conversion using a pen-type stylus. There is also a need for a system that gives the user the option of using the keyboard or the pen-type stylus at any instance of operation of the conversion process.
SUMMARY OF THE INVENTION

The present invention overcomes the deficiencies of the prior art with a method for performing Kana to Kanji conversion using a pen-based computer. The method of the present invention takes full advantage of the pen as an input device, and provides for KKC using gestures or simple pen strokes. In accordance with the present invention, the preferred method for converting Kana characters to a Kanji character comprises the steps of: displaying a plurality of edit boxes; using the pen to input data; performing character recognition on the input data; simultaneously indicating a KKC operation and characters in an original string for the operation using the pen; determining the KKC operation to perform; retrieving the original string indicated; performing KKC on the original string; replacing the original characters with a new equivalent returned by KKC; and displaying a new string including the equivalents returned by KKC. This method makes KKC with pen-based systems much easier and efficient for the user, and therefore, provides pen-based systems with greater utility for input and output of Asian characters.

The present invention also includes other methods that are advantageous for performing KKC including a method for resolving gesture/character conflicts that provides the user with either the gesture or the character option regardless of the context. The present invention also includes a method for displaying substitutes or guesses for both Kana and Kanji characters when the character recognition is inaccurate. The presentation of substitutes is performed by presenting an alternative list or by automatic replacement of a character with another guess using gestures. Finally, the present invention also provides a method that allows the user to easily switch between use of the keyboard and the pen at any instant or level of the KKC process.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 illustrates a block diagram of a pen-based system upon which the method of the present invention operates;
Figure 2A is graphical representation of the display device showing a preferred embodiment for an edit pad for inputting characters using the keyboard or pen-type stylus;

Figures 2B and 2C are graphical representations of the display device showing edit boxes, a gesture and a box for presenting substitute options or guesses in the alternative list to the user;

Figures 2D and 2E are graphical representations of the display device showing edit boxes and gestures for automatically selecting and replacing a character with a substitute or guess from the alternative list;

Figure 3 is a table of preferred gestures for performing Kana to Kanji Conversion in accordance with the present invention;

Figures 4A and 4B are a flowchart of a preferred method of the present invention for converting Kana characters to Kanji characters;

Figures 5A and 5B are a flowchart of a preferred method of maintaining and presenting the alternative list;

Figure 6 is a flowchart of a preferred method of resolving spatial modality conflicts according to the present invention;

Figure 7 is a state diagram for a preferred method of resolving conflicts between pen and keyboard entry in accordance with the present invention;

Figures 8A-8H are graphical representations of the display device showing preferred embodiments of the edit boxes, input, highlighting and character conversion using the keyboard according to the present invention; and

Figures 9A-9F are graphical representations of the display device showing preferred embodiments of the edit boxes, input, gestures, highlighting and character conversion using the pen according to the present invention.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention will now be described with reference primarily to the conversion of Kana characters to Kanji characters. However, it should be understood that the present invention is directed to methods that convert data
from a form optimal for input to a form optimal for output, and vice versa. For example, the conversion method of the present invention can convert Kana characters to Kanji characters, a mixture of Kanji and Kana characters to Kanji characters, codes to Kanji characters, or even Kanji characters to Kana characters. The present invention advantageously operates to perform any of the above conversions to translate the input into a form for display desired by the user.

Referring now to Figure 1, a preferred embodiment of a pen-based computer system for converting characters from Kana to Kanji in accordance with the present invention is shown. The preferred embodiment of the pen-based system comprises a central processing unit (CPU) 10, input devices 12, a display device 14, an addressable memory means 16 and mass storage 18. The CPU 10 is coupled to and controls the display device 14 to produce a variety of images in response to inputs supplied to the CPU 10 by user manipulation of the input device 12. The CPU 10 is also coupled to other sources of information such as mass storage 18 and addressable memory 16 in a conventional architecture. In an exemplary embodiment, the CPU 10 may be a microprocessor from the 'X86 family produced by Intel or the 68000 family produced by Motorola.

The input device 12 is a conventional type as known in the art. The input device 12 preferably includes both a keyboard and a pen-type stylus (pen) with a digitizing pad. The pen is manipulated by a user 30 as would a normal writing instrument, but over the digitizing pad or capacitive grid proximate the display device 14. Together the pen and pad translate the movement of the pen into digital signals usable by the CPU 10. Thus, normal handwriting and gestures can be input using the pen to direct the operation of the system. The system may also be operable in the more conventional manner using the keyboard.

The display device 14 is also a conventional type known in the art. The display device 14 is preferably a liquid crystal display used with the CPU 10 in a conventional manner to produce images on the display device 14 such as from
a group of dots or pixels in graphics mode. The display device 14 is preferably mounted directly above the digitizer or pad such that moving the pen on top or proximate the screen of the display device 14 produces an image or changes on the display 14. The display device 14 may also be a raster-type display that produces images of characters generated from codes such as ASCII in text mode. The display device 14 also operates in a conventional manner with the input device 12 to produce a various ink images on the display device 14 or gestures for inputting commands to the system.

The addressable memory 16 is a conventional type and preferably includes Random Access Memory (RAM) and Read Only Memory (ROM). The addressable memory 16 further comprises processing routines, programs and data for interactive display control, input/output and KKC as the user 30 runs application programs and uses the system. For example, the memory 16 preferably includes an operating system 20 such as PenPoint 1.0 by GO Corporation. The memory 16 also includes routines for transferring data from the CPU 10 to the display device 14 and for presentation of the data on the display device 14. The memory 16 further includes other routines and application programs 28 as conventional in the art.

Still more particularly, the memory means 16 of the present invention further comprises a Kana to Kanji Conversion (KKC) dictionary 22, pen and keyboard input routines 24, and KKC routines 26. The KKC dictionary 22 is a preferably a table or list of Kana characters and their Kanji equivalents (input/output pairs). The KKC dictionary 22 also comprises routines for displaying a portion of the list, for adding pairs to the list, for deleting pairs from the list, and for searching the list. The KKC routines 26 and the pen and keyboard input routines 24 are used for converting Kana characters, and processing input and output from the display device 14 and the input device 12, respectively, as will be described in more detail below.

Referring now to Figures 2, 8 and 9, an overview of the preferred methods of the present invention will be described. The input process begins by presenting an edit pad 40 on the display device 14, as shown in Figure 2A. The edit
pad 40 may have a variety of lengths and formats including boxed, ruled, and ruled/boxed. The edit pad 40 of Figure 2A has a boxed format with a plurality of boxes 42 for insertion of a character in each box 42. The edit pad 40 is used to input data, perform handwriting recognition, correct the results of handwriting recognition, as well as perform Kana to Kanji conversion. The user 30 preferably moves the pen proximate the display device 14 and over the edit pad 40 to input handwriting images, gestures and other commands. Once characters have been handwritten and converted to Kana characters, the methods of the present invention are used to covert the Kana characters to Kanji characters.

The preferred process for keyboard based character conversion and correction is shown in Figures 8A-8H. The conversion begins in the initial state where a mix of Kana and Kanji characters 44, 48 may be displayed as shown in Figure 8A. The keyboard is used to insert additional characters 46 into edit boxes 42 at the position of a cursor 47 as shown in Figure 8B. The characters 46 that were most recently entered are displayed with a weak highlight 50 along the bottom of their respective edit boxes 42. In the preferred embodiment, the weak highlight 50 is shown as an enlarged or bolded strip along the bottom of the boxes 42 having the weak highlight 50. Those skilled in the art will realize that a variety of other types of highlighting such as reverse video, bold or underlining may be used to set apart recently converted characters.

The conversion can then be initiated. The conversion will operate on the characters between the beginning position and the cursor 47. These are the same characters 46 having the weak highlight 50. The initial guess at a probable conversion equivalent is shown in Figure 8C. The results of KKC (phrases 54 and 56) are then substituted in the string in place of the Kana characters 46 they represent as shown in Figure 8D. The present invention advantageously displays the first phrase 54 with a strong highlight 60, and the other phrase 56 with a weak highlight 50. The strong highlight 60 is preferably in the form of a darkening or bolding of the entire box 42 in the edit pad 40. However,
those skilled in the art will realize that a variety of other highlighting techniques may be used.

The first phrase 54 is shown with the strong highlight 60 to enable the user to immediately proceed to the correcting phase. Using keyboard commands, the user 30 can display an alternative list 64 in a box 62 positioned below the phrase 54, as shown in Figure 8E. The keyboard may then be used to select a choice 65 from the list 64. As illustrated in Figure 8F, the choice 65 is advantageously display with a strong highlight 60 to set the choice apart from the other alternatives. It should be noted, the present invention only displays the portions of the phrase alternatives that are different. Thus, since all the alternatives in the lists 64 of Figures 8E and 8F end in the same character, it is not displayed as part of the alternatives. Therefore, each of the alternatives is a substitute for only the first two characters of phrase 54. Once the choice 65 has been selected, it is inserted in place of the initial guess as shown in Figure 8G, and the next phrase 56 is strong highlighted. This phrase 56 may be similarly corrected using an alternative list 64 corresponding to its characters. Once the desired result has been reached all highlighting is removed, and additional input can be accepted as shown in Figure 8H. Alternately, the user 30 may continue input at any time and the current state will be accepted and input will continue.

Referring now to Figures 9A-9F, the preferred process for pen based character conversion and correction will be described. As shown in Figure 9A, the characters 44, 46, 48 are displayed on a portion of the edit pad 40 after handwriting recognition has been performed. The Kana to Kanji conversion process begins with the user 30 identifying the operation to be performed, and characters upon which to perform the operation. The method of the present invention advantageously uses a gesture (simply shaped pen stroke) to indicate both the operand and the operation. Thus, little effort is required by the user 30 to initiate the conversion process. As shown in Figure 9B, the exemplary gesture 52 for performing KKC is moving the pen right over some distance and
then up (Right Up). The method of the present invention recognizes the gesture 52 and performs KKC on the characters 46, 48 beginning with a hot point and ending with the next space or period encountered. The hot point for this gesture is the location or character where the pen first touched the screen. The KKC returns only a single phrase 54. This phrase 54 is substituted in the string for the characters they represent. As shown in Figure 9C, the other characters 67, 48 in the string are returned unchanged. When the phrase 54 is inserted in the string, it is advantageously displayed with the strong highlight 60. The strong highlight 60 is preferably in the form of a darkening or bolding of the entire box 42 in the edit pad 40 similar to the keyboard based conversion.

If the KKC process has improperly translated the Kana characters 46, the user 30 simply performs the tap gesture 70, illustrated in Figure 9D, on the first phrase 54 by tapping with the pen at the location on the display 14 where the first phrase 54 is shown. In response, the system of the present invention displays the box 62 with the list 64 of alternatives based on the phrase 54 that was tapped upon, as shown in Figure 9E. The user 30 simply taps with the pen again on any phrase 65 in the alternate list 64 and that phrase 65 is substituted in the string shown in the boxes 42 of edit pad 40 at the position of the first phrase 54. The substituted phrase 65 is then displayed with strong highlighting 60, as shown in Figure 9F.

The present invention also supports the generation of alternative lists 64 for handwritten characters as shown in Figures 2B and 2C. Figure 2B shows the boxes 42 of the edit pad 40 displaying a string of Kana characters 68. If the user 30 taps on a character 72 in the string 68 as represented by the dot and arrow 70, the display 14 changes to display the character 72 with strong highlighting 60, and presents the alternative list 64 of likely Kana alternatives based on handwriting and other similarities, as shown in Figure 2C.

The present invention also provides for direct substitution of likely equivalents without displaying the alternative list 64. The present invention provide two
gestures, flick up 74 and flick down 76, for direct substitution. A flick up 74 on a character 78, as shown in Figure 2D, will substitute the next character 80 in the list 64 for the character 78 over which the flick up gesture 74 was drawn, as shown in Figure 2E. Similarly, the flick down gesture 76 on a character 82, as shown in Figure 2E will substitute the previous character in the list 64 for the character over which the flick down gesture 76 was drawn. Both these operations are performed without a display of the alternative list 64 or highlighting of the substitution process. It should be understood that the flick up 74 and flick down 76 gestures operate similarly for direct substitution of KKC Kanji characters and phrases alternatives.

Referring now to Figure 3, a table of a preferred set of gestures for performing various KKC operations is shown. The table indicates the gesture, the action performed by the gesture, and any keyboard equivalents.

Referring now to Figure 4, a preferred embodiment of the method of the present invention for converting Kana characters to Kanji characters is shown. The method begins in step 400, and displays the edit pad 40 like that shown in Figure 2A on the display device 14 in step 401. The edit pad 40 is then used to input data using the pen in step 402. The handwritten images input using the pen and displayed on the edit pad 40 are then converted to Latin or Kana characters under control of the user in step 403. A typical scenario involves having the user 30 hand write characters in the boxes 42 of the edit pad 40 using the pen. The user 30 then taps with the pen on the OK button 41 to initiate character recognition on the handwriting entered in the boxes 42. The handwriting recognition process returns the characters identified or its best guesses. For the handwritten image for which no match or guess can be formulated, an unknown symbol such as a question mark is displayed. The user 30 can then reenter the handwritten character over the question mark until it is recognized.

Once steps 402 and 403 have been completed, a string of Kana characters are displayed by the pen-based computing system for conversion to Kanji characters as shown in Figure
9A. Next, in step 404, the user 30 draws a gesture 52 over characters 46 displayed on the edit pad 40 to initiate the conversion process. The present invention uses a group of gestures, as noted in Figure 3, for performing the various conversion operations. The use of gestures is particularly advantageous because with a single action, the user 30 is able to specify both the operation and the operands. This makes the method of the present invention significantly easier and more efficient for the user 30 as compared to other conversion methods. The present invention also accommodates use of the keyboard to initiate the conversion process, identify the operands, and specify the operation, as has been detailed with reference to Figure 8.

Once the gesture 52 or keyboard command has been input, the method of the present invention determines the character where the conversion begins, in step 405. For keyboards, the conversion begins with the first character entered after the conversion mode is selected. For gestures, the conversion begins with a hot point or the position where the pen first touched the screen. For example, the right up gesture 52, shown in Figure 9B, begins at the third character 46 from the left. The method of the present invention uses the first character 46 marked with the gesture 52 as the beginning point for the conversion process. Then in step 407, the present invention retrieves an original string of characters from the beginning character to the end character inclusive. The end character is again different depending on whether the keyboard or pen is used. The end character for the keyboard use, is the character at the position of the cursor when the covert command is input. For pen input, the end character is the first period or space encountered to the left of the beginning character. With gesture input, all the characters to the left of the beginning character 46 are retrieved until a period or a space (i.e., empty box) is encountered. Those skilled in the art will realize that the character string to be converted may be retrieved by searching to the left, right, or in a downward direction depending on the format of the edit pad 40 and the convention of character flow being observed, or that characters referred throughout as
"Kana" may in fact include Kanji, and that characters to which they convert called "Kanji" may include Kana.

Once the character string of Kana characters 46, 48 has been identified in step 407, the Kana to Kanji conversion operation is performed on this character string in step 408. The KKC action is preferably performed using a conventional translator known to those skilled in the art. For example, KKC engines such as those provided by Vacs VJE may be used. Once a new equivalent phrase 54 to a portion of the original character string 46 has been identified, the new equivalent phrase 54 is substituted in place of the equivalent characters 46 in the original string in memory, in step 409. Edit pads 40 displaying the converted characters 54 are shown in Figure 8C and 9C. During the KKC step of 407, the present invention also generates the list 64 of conversion alternatives using the KKC dictionary in step 410. The alternative list 64 contains a list of likely alternates for each phrase that has been converted from Kana characters. The alternatives are determined using standard usage principles, visual similarities, homophones and other factors as will be described in more detail below. After, the alternative list 64 has been generated or supplemented by the KKC process in step 410. The new character string having the Kana characters replaced by Kanji characters is shown on the display device 14. The KKC process ends in step 412.

As has been previously mentioned, the KKC method of the present invention advantageously creates and maintains an alternative list 64 for each character or phrase displayed in the edit pad 40. The alternative list 64 is preferably generalized to include alternatives produced from assumptions about what errors are likely to occur in the handwritten input stream. Certain types of mis-recognition are more likely to occur than others. For example, some hiragana and katakana characters have variants that are the same shape, only smaller in size. Also, many of the Katakana characters can resemble the Kanji characters from which they were derived. In the preferred embodiment, the alternative list 64 is generated from appropriate combinations of: normal homophones, semantically related choices, shape-related characters or
choices, homophones based on shape variants of the input string, the original input string, choices that are normally considered a reverse conversion process, expansions of user defined abbreviations or aliases, thesaurus entries, Kanji characters related by radical components, or rendering variations of any choice. Thus, when a small letter is mis-recognized as a large letter, the choices for the string containing the small letter will also be present in the alternative list 64. When alternative lists are being generated on a character shape known to be a problem, the lists of the present invention will contain known look-alike characters in a fixed order which does not vary and which is put at the front of the list. Therefore, using the factors and methods set forth above, the present invention assembles very comprehensive alternative lists.

Referring now to Figures 5A and 5B, a flowchart of the preferred method for displaying and using the alternative list 64 is shown. The process begins in step 601 with the input of a gesture by the user 30. Then in step 602, the system determines the beginning point for the gesture. Once the beginning point has been determined, the system tests whether the gesture is over a previously converted phrase or a new character in step 503. If the beginning point of the gesture is over a previously converted phrase, the method proceeds to step 507 to derive the alternative list 64 from the KKC dictionary or the list 64 is retrieved from a table if KKC was recently performed. Then the method proceeds to step 508. However, if the gesture was written over newly input character in step 503, the gesture is probably over a character, and the preferred method retrieves character alternatives from a look up table in step 504. The method continues to step 505 to retrieve further character alternatives based on handwriting. The two lists are then merged in step 506, and the method proceeds to step 508. In step 508, the process next determines whether the gesture was a flick or a tap. If the gesture was a tap the process continues in step 509, and if the gesture was flick the process continues in step 510.

As shown in Figure 5B, the processing of the tap
gesture on a character in the edit pad 40 proceeds as follows. First, the alternative list 64 is displayed on the display device 14, as shown in Figure 2C. Next, in step 512, the user inputs another gesture to substitute a character or remove the alternative list 64 from the display 14. In step 513, the method determines whether the gesture input in step 512 was a tap on a character in the alternative box 62 or any other gesture outside the character box 62. If the gesture was outside the character box 62, the box 62 and the alternative list 64 are removed from the screen in step 515, and the character string remains unchanged. Then the process ends in step 516. However, if the gesture input in step 512 was a tap on a character in the alternative box 62, the method proceeds to step 514. In step 514, the character at the beginning point determined in step 502 is replaced both in memory and on the display 14 with the character tapped upon in the character box 62. Then in step 515, the alternative list 64 is removed from the display and the process ends in step 516.

If it was determined that a flick was entered in step 508, the process continues to process the flick in step 517. In step 517, the method further processes the gesture to determine whether it was a flick up or a flick down. If the gesture was a flick up, the method determines the character at the beginning point and replaces the character with the next character in alternative list 64 in step 518. On the other hand, if the gesture is a flick down, the method jumps to step 519, where the character at the beginning point is replaced with the previous character in alternative list 64, in step 518. After completion of either step 518 or 519, the method displays the updated character string on the display device 14 in step 520. The updated character string comprises the character string with the character at the beginning point replaced with next or previous character in the alternative list 64. The process then ends in step 516.

Referring now to Figure 6, another feature of the present invention will be described. The method of the present invention advantageously includes a process for automatically resolving spatial modality conflicts. Spatial modality refers to the interpretation of an input based on the
instance or location of the input. For example, two identical
or similar inputs can have very different meanings depending
on the time and context in which they are input. The process
begins in step 601 with entry of input by the user 30 using
the pen. Then in step 602, the method of the present
invention determines whether the input is similar to a
character in the spatial modality list. The spatial modality
list is a list of pen stroke shapes that can have more than
one meaning depending on the context in which they are input.
Examples of such inputs in the pen-based system that the
preferred embodiment operates upon include and "X" or a "."
These inputs have two different possible meanings. The "X"
can mean either the character x or the gesture for the delete
operation. The "." also has two different meanings. It can
be the input of a period as a punctuation mark or the tap
gesture. Those skilled in the art will realize there may be a
variety of inputs that have spatial modality conflicts
depending on the set of pen gestures relied upon to perform
the functions of the pen-based operating system.

If the input does not match any characters in the
spatial modality list, the method of the present invention
simply processes the input in a normal fashion since no
spatial modality problem exists as evidence by no match with
the spatial modality list. The method then tests whether the
input matches or resembles a gesture in step 603. If the
input does match a gesture, it is processed as a normal
gesture in step 604 and the process ends in step 606.
However, if the there is no match between the input and the
gestures of the system, then character recognition is
performed on the input in step 605 and the process ends in
step 606.

However, if there is a match between the input of
step 601 and the spatial modality character list, the method
proceeds to step 607. In step 607, the gesture and character
that are similar to the input are determined. This is
conventionally done using handwriting recognition techniques.
Next, in step 608, the method determines whether the input was
written over a character or a blank space. If the input was
written over a character, the method proceeds to step 611.
In step 611, the method assumes that the input should be interpreted as a character, and a character is substituted for the blank space. The process then ends in step 606. However, if the input was written over a character, the method assumes the input was a gesture, and performs the operation associated with the gesture in step 609. Next, in step 610, the method of the present invention interprets the input as a character, and adds the character to the on the alternative list 64. Thus, the present invention provides a simple method by which the input or character can be recalled if is was incorrectly interpreted as a gesture. The user 30 only has to tap with the pen on a character or phrase, and a list of alternatives will be presented. The character just input has been added to the alternative list 64 in step 610, and therefore, can be easily inserted in the appropriate space if the operation indicated by the gesture was not intended by the user 30. Once the character has been added to the alternative list 64 in step 610, the process ends in step 606.

The present invention further increases the efficiency and ease of use of pen-based systems by integrating the keyboard and pen into the Kana to Kanji conversion process. The system and method of the present invention receives a variety of commands and character inputs. The input can be provided with either the keyboard or the pen. In other words, for each command or character, there is both a keyboard input (keys) or a pen input (gesture) that can initiate the command or input the character. More particularly, the present invention is advantageously integrated such that at any instant during the character entry or conversion process, the user 30 may switch between using the pen and the keyboard, or vice versa, as the input device.

Referring now to Figure 7, a state diagram for a preferred method of operation of the pen and keyboard with a system of Figure 1 will be described. During the KKC process, the method of the present invention is divided into four possible states of operation. These states include: a quiescent state 701, a collecting state 702, a translating state 703 and a correcting state 704. The method begins in the quiescent state 701 in which system awaits input. The
system leaves the quiescent state 701, if either a character is input using the keyboard or a conversion gesture is input using the pen. Thus, a transition out of the quiescent state 701 occurs with the use of either the pen of keyboard. If a conversion gesture is input, the system transitions to the translating state 703. If a character is input using the keyboard, then the system transitions to the collecting state 702.

Once in the collecting state 702, the system records the position of the character just input that caused the transition to the collecting state 702. The system also monitors for additional input from the keyboard. In this state, the system collects or supplements the character string on which KKC is to be performed by building an active range of characters. Additional character input from the keyboard is processed, and the system remains in the collecting state 702. As with all the states in the present invention, either a keyboard input or a pen input can cause the system to transition out of the collecting state 702. For example, in the collecting state 702, either a convert command from the keyboard or the convert gesture from the pen will cause the system to proceed to the translating state 703.

The translating state 703 is a transitional state in which no input is accepted. As noted above, the system may transition to the translating state 703, either directly from the quiescent state 701 upon the input of the convert gesture, or from the collecting state 702 upon the input of either a convert gesture from the pen or a convert command from the keyboard. During the translating state 703, the system sends a message to perform KKC on the string just entered in the previous collecting state 702 as denoted by the active character range. If the transition was from the quiescent state 701, the active range is determined by the location and range upon which the gesture was made (i.e., the text over which the gesture was input). The characters in the active range are packaged with highlighting boundaries and other information, and processed by other KKC routines. After the KKC process is complete, the system automatically proceeds to the correcting state 704.
In the correcting state 704, the results of the KKC process are presented to the user 30. During the transition from the translating state 703 to the correcting state 704, a variety of display parameters are passed for the appropriate highlighting and display of the KKC results. The user 30 may input a variety of commands via pen or keyboard to correct characters that were not correctly translated. For example, the system will remain in the correcting state 704 when commands such as substitute next alternative character, substitute previous alternative character, substitute previous Kanji phrase, activate alternate menu, extend phrase, or reduce phrase are input using either the keyboard or the pen. The system transitions out of the correcting state 704 in two ways. If the user 30 enters characters using the keyboard, then the system will transition to the collecting state 702 assuming that the user 30 is inputting additional characters for conversion. If the user 30 enters an accept conversion command using the keyboard or the pen then the system will transition back to the quiescent state 701 to await further input. Thus, with the transitions and states provided above, the present invention is particularly dynamic since it allows smooth and seamless switching between pen operation and keyboard operation at anytime during the conversion process.

While the present invention has been described with reference to the preferred embodiments, those skilled in the art will recognize that various modifications may be provided. These and other variations upon and modifications to the described embodiments are provided for by the present invention, the scope of which is limited only by the following claims.
WHAT IS CLAIMED IS:

1. A method for converting Kana characters to Kanji characters using a computer system having a display device and a pen input device, said method comprising the steps of:
   displaying an input string of Kana characters on the display device;
   inputting a gesture with the pen input device to simultaneously indicate a conversion operation and conversion operand, said gesture being input over a first sub-string of at least two Kana characters;
   determining the conversion operation by translating the gesture;
   retrieving a conversion sub-string beginning with the first sub-string;
   performing Kana to Kanji character conversion to produce a Kanji sub-string;
   creating a new string by replacing the conversion sub-string with the Kanji sub-string; and
   displaying the new string on the display device.

2. The method of claim 1, wherein said step of retrieving a conversion sub-string comprises the sub-step of:
   determining a start character over which the gesture was first input; and
   selecting the conversion sub-string to include the start character, and characters between the start character and one from the group of a space character and a period character.

3. A method for resolving spatial modality conflicts arising with a pen input device in a pen-based computer system, said method comprising the steps of:
   monitoring for pen input from the pen input device;
   comparing the pen input to a list of inputs with spatial modality conflicts;
   processing the pen input using normal gesture and character recognition techniques, if the pen input does not match an input in the list of spatial modality conflicts;
performing the following sub-steps if the pen input does match an input in the list of spatial modality conflicts:

performing handwriting recognition on the pen input to determine the character input and the gesture input;

determining whether the pen input was over a blank space or a character;

if the pen input was over a blank space, interpreting the pen input as a character and inserting the character in the blank space; and

if the pen input was over a character, interpreting the pen input as a gesture, and performing the operation associated with the gesture on the character over which it was input.

4. The method of claim 3, further comprising the step of:

if the pen input was over a character, interpreting the pen input as a character, and adding the character to an alternative list.
Figure 1
<table>
<thead>
<tr>
<th>GESTURE</th>
<th>NAME</th>
<th>KEYBOARD</th>
<th>ACTION</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Gesture" /></td>
<td>TAP</td>
<td>SPACE (AFTER CONVERSION HAS TAKEN PLACE)</td>
<td>OPEN LIST OF ALTERNATIVES</td>
</tr>
<tr>
<td><img src="image2" alt="Gesture" /></td>
<td>FLICK</td>
<td>UP, DOWN, LEFT, RIGHT, HOME, END</td>
<td>NEXT CHOICE PREV CHOICE (SUBSTITUTE IN PLACE)</td>
</tr>
<tr>
<td><img src="image3" alt="Gesture" /></td>
<td>RIGHT–UP</td>
<td>SPACE</td>
<td>CONVERTS CHARACTERS</td>
</tr>
<tr>
<td><img src="image4" alt="Gesture" /></td>
<td>RIGHT–UP–FLICK</td>
<td></td>
<td>ALTERNATE KKC</td>
</tr>
<tr>
<td><img src="image5" alt="Gesture" /></td>
<td>RIGHT–DOWN</td>
<td></td>
<td>REVERSE CONVERT</td>
</tr>
<tr>
<td><img src="image6" alt="Gesture" /></td>
<td>LEFT ARROW RIGHT ARROW</td>
<td>SHIFT+LEFT ARROW SHIFT+LEFT ARROW</td>
<td>SHORTEN PHRASE EXTEND PHRASE</td>
</tr>
</tbody>
</table>

**FIGURE 3**

**FIGURE 2D**

**FIGURE 2E**
400 Start

401 Display edit pad

402 Input data using pen

403 Convert input data to characters using handwriting recognition

404 Use gesture or keyboard to indicate conversion operation and operands

405 Determine the character that conversion begins with (target character)

406 A

Figure 4A
Retrieve original character string from beginning character to the end character.

Perform KKC on the character string.

Store new characters found in KKC for characters of the original string.

Generate character alternative list for KKC using KKC dictionary.

Display new character string including equivalents from KKC.

End.

Figure 4B
500  
Start

501  
User inputs a gesture

502  
Determine the beginning point of the gesture (i.e., the target character)

503  
Is the beginning point over a previously converted phrase?

504  
Retrieve character alternatives from a static look up table

505  
Retrieve additional character alternatives based on handwriting recognition

507  
Derive character alternative list of Kanji from table/KKC dictionary

508  
Is the gesture a flick or a tap?

509  
Tap

510  
Flick

506  
Merge the two list into a character alternative list

Figure 5A
Figure 5B
Figure 6

600 Start

601 Enter input with the pen

602 Is input similar to a character in spacial modality list?

603 Is the input a gesture?

604 Process as a normal gesture

605 Perform Character Recognition

606 End

607 Determine the similar gesture and character

608 Is the input written in a blank space or does it have a character beneath it?

609 Interpret as a gesture and perform operation of the gesture on the character below

610 Interpret input as a character and add the character on the character alternative list

611 Interpret as a character and insert the character in the blank space.
Figure 7

Diagram:

- **701 Quiescent State**
  - Keyboard Character
  - Convert Gesture

- **702 Collecting State**
  - Convert Key or Gesture

- **703 Translating State**

- **704 Correcting State**
  - Accept Conversion Key or Gesture
  - Keyboard Correcting Key or Gesture