

particularly, the present invention relates to an information storage, manipulation and transfer device on which text, computer commands and functions can be entered by writing alphanumeric or any other characters and symbols by means of a penlike stylus on an Input/Output (I/O) screen. In one embodiment the I/O screen includes a transparent touch screen incorporated over a substantially flat output display. The present invention in its preferred embodiment is a self-contained computer system but can also function as a peripheral device for a computer.

Large amounts of information and sophisticated applications software are now available on conventional computers. The utility of this information and of the software could be greatly increased if text and data

requirements of the position sensing input technology. Essentially these requirements include accuracy, resolution, and speed. The technologies include: mechanical, electromagnetic, acoustic, optical, and inertial. One of the goals in this system is to have its use as similar as possible to writing with pen or pencil on paper. One problem with writing with a pen on paper only leaves a trail when actually writing. Many of these technologies require an additional "position" sensor which is awkward to use in many commercial applications. Another problem is writing angle -- a pen leaves a trail independent of writing angle. Many of these technologies have the position detector displaced from the pen tip, which causes erroneous displacements. Beyond these general considerations, each technology has numerous advantages and disadvantages: (1) the pen: size, weight, shape and whether it is battery powered and/or wired, and (2) the writing surface: transparency, smoothness, "feel", and whether or not contact is needed (as opposed to pressure transmitted through overlaying sheets of paper).

collection system is disclosed which employs a solid state Random Access Memory (RAM) for its primary memory. The system which discloses a conventional keyboard for data entry is connected to a hand-held portable data storage and transmission system, as well as an LED display and various Input/Output (I/O) connectors.

U.S. Patent 3,487,731 to Frank discloses a system for converting handwriting into character data through a computer system. The disclosed invention is based on a pattern matching and does not employ any coincidence detection technology.

U.S. Patent No. 4,491,960 to Brown shows a symbol recognition system in which an array of image data in the form of a raster line sampling, is converted into a set of oriented lists which are filtered and compressed to extract topologic features which are then analyzed with a decision mechanism.

an electroscopic information display and entry system, writing stylus. The disclosed embodiment does not have the speed or accuracy to enable handwritten character recognition.

Additional prior art which discloses portable electronic devices that provide large amounts of various stored information include U.S. Patents 4,218,760 to 4,415,486 to Laine; and 3,932,859 to Kriakides et al. Levy and Kriakides et al. patents pertain to electronic dictionaries, while the Laine patent discloses a program television reminder system. None of these devices use of a handwritten input.

In U.S. Patents 4,071,691, 4,129,747, 4,141,429, 4,293,734, 4,302,011, 4,353,552, 4,371,746 and 4,431,111 William Pepper, Jr. various methods or machine-human communication using finger touch are disclosed. The preferred embodiment of each of these inventions lack sufficient speed and accuracy to allow handwritten character recognition with a stylus designed for other purposes. U.S. Patent 4,318,096

ability to recognize and display handwritten symbols
computer to display Font Symbols and, if desired, to
, functions pursuant to Editing Symbols, quickly, easily,
reasonable cost.

This preferred embodiment includes a computer
a flat display panel on which a user may "write" with
has a capability to recognise Handwritten Symbols written
panel with the Stylus and convert them to displayed Font
and/or to execute Editing Functions with Editing Symbols
all with a minimum of technical complexity for the user.
keyboardless, portable computer is loaded with the desired
information and applications software, information and
be used and responded to without requiring skills or knowledge
related to state-of-the-art computers or other data systems.
ease-of-use of the input technology of the apparatus and
utility of the computer for keyboard oriented individuals.
Its portability also allows it to be used in applications
settings in which portable keyboard computers are awkward
or impossible to use. For example, a multiplicity

Below, the present minimum requirements are: accuracy of 0.125mm (0.005 inch), resolution of 0.375mm (0.015 inch), and a writing speed of 150 Points per second. This accuracy allows a 6mm high writing line with over 10 raw input Points along a Stroke of a letter. The resolution provides positioning of the symbol to 0.375mm (0.015 inch) pixels on a present display of 640 pixels to 225mm (9 inches). The writing speed permits about 50 raw input Points for a rapid entry of a single letter (1/3 second).

One embodiment of the present invention comprises a transparent input screen. As the user writes alphanumeric characters or symbols on the input screen, the input is represented as a stream of Points emulating writing with pen on paper. Once the discrete alphanumeric characters or symbols are complete, they are translated into computer text or computer commands that can be displayed on a display screen at a location that is preferably beneath the input screen at a location that is preferably beneath the input screen where they were entered. The system also comprises a pattern recognition algorithm which translates any written character or symbol, such as ideographs and scientific symbols, into computer text.

mathematical symbols are Handwritten Symbols. Other types of Handwritten Symbols are free-hand drawings or any other such written information uniquely configured by a particular writer. Handwritten Symbols may also include Editing Symbols (defined below).

As used herein, "Font Symbols" are computer-generated symbols which are displayed in a predetermined format. By way of example and not limitation, alphanumeric symbols, Font Symbols and displayed in numerous font formats, or Chinese "ideograms" may also be Font Symbols, as well as engineering, scientific, mathematical, architectural, and such characters. Other examples of Font Symbols include a form which can be stored and displayed by a computer as a drawing of a car or a house.

As used herein, an "Editing Symbol" is a symbol (such as a caret, horizontal line, short vertical line, or a line, etc.) which is intended, when recognized, to instruct a computer to execute a particular Editing Function (defined below), such as insert text (caret), delete text

input screen. The system includes a computer program
, compile the positional information into Strokes, to calculate
characteristics, and then compare the Stroke characteristics
those stored in a database in order to recognise the symbols
the stylus. Key features of the system are: (A) touch
position sensing subsystem; (B) underlying display on which
mimic drawing of sensed positions and to show characteristics
(C) means to convert sensed positions first into plotted
then into recognised characters or symbols; and (D) means
"learn" to associate sensed input positions with a character
symbol. The apparatus can be housed in a portable unit with
dimensions up to about 400mm by 400mm by 100mm (16" x 16" x 4")
a weight of up to about 7 kilograms (15 pounds), with a
contained power supply.

screen, stylus and associated electronics;

Fig. 4 is an overall schematic system block diagram of the apparatus of a keyboardless entry computer system according to the present invention;

Fig. 5 is a schematic block diagram depicting the organization of data within the system when modified by handwritten characters and commands;

Fig. 6 is an overall system block diagram illustrating the hierarchy of software used to operate the system;

Fig. 7 is a generalized block diagram of the system for character and pattern recognition algorithm.

Figs. 8A and 8B together are a detailed block diagram of the pattern recognition algorithm.

Fig. 9 is a schematic block diagram of the system for character characterization subroutine.

Fig. 10 is a top plan view of a screen depicting the "initializing" of the database for Handwritten Character Recognition.

Figs. 11A through 11I are a series of top plan views of screens depicting the operation of a text editing system.

in greater detail hereinbelow. Input information is received by microcomputer 14 by stylus 16 "writing" on writing screen 18. Stylus 16 (Fig. 2) is connected to the computer system 10 with wire 17 (Fig. 2). As stylus 16 "writes" on input screen 18, a plurality of locating signals are generated, each of a plurality of corresponding positional coordinates, which are transmitted to microcomputer 14. Microcomputer 14 is programmed in accordance with a computer program described hereinbelow, to recognize the stream of locating signals and to store these signals in a computer memory. The program of microcomputer 14 also provides a corresponding plurality of display signals to a display screen 20. Both input screen 18 and display screen 20 are described in greater detail hereinbelow.

Referring now to Fig. 2, there is shown a perspective view of keyboardless computer system 10 embodying the present invention. Keyboardless computer system 10 is contained in housing 12, which is a rectangular enclosure having a sloped top surface 22 with a multi-line,

such as to enter computer commands. Exemplary commands are "store," "recall," and "delete." Softkeys 26 are "store," "recall," and "delete." Softkeys 28 can be used to switch between different modes (e.g. data entry mode and edit mode). Softkeys 28 are optional and are used to supplement data obtained by handwriting the entries. Stylus 16, used for writing input data and commands in display area 2, is used to activate the selected Softkey 28. An ON/OFF switch 28 is positioned on the side of housing 12 adjacent to Softkey 28. A data output or peripheral connector 31 is located on the upper right side of housing 12.

Input screen 18 can be a conventional resistive touch screen in which a voltage is applied to the screen and a stylus detects the voltage at the touched location. The writing surface is a transparent material, typically glass, coated with a thin, uniform, conductive layer (preferably vacuum deposited indium tin oxide). Vertical bus lines and conducting strips (not shown) are used along the

presently preferred embodiment, a soft graphite tip. The voltage is conducted from the pen through a wire 17 in Fig. 2, to an analog to digital converter for the computations described below. The stylus may be a "pen" as described herein, a light pen as is well known in the art, or any other hand-held device which can outline or write symbols on a screen.

An example of a conventional electrostatic touch screen disclosed in the aforementioned 4,318,096 Pepper patent is a resistive type screen has the advantage that the noise caused by the user's hand touching the screen is minimized.

Both horizontal and vertical position sensing is provided by alternately switching the voltage impressed on the conductive layer between the pairs of horizontal and vertical bars by an interface and multiplexer controlled by a computer or microcontroller. In one commercially available input or touch screen, the bus bars are broken into short strips with diodes to prevent the horizontal strips from shorting out vertical strips and vice versa. This

Points will vary with the distance to the pen position X_1 , Y_1 , indicated at 38. These voltages sequentially measured in the X and Y directions by conventional means, such as disclosed in the aforementioned art patents. In Fig. 3 these means are a computer interface/multiplexer 42. A conventional Analog-to-digital converter 43 converts the detected voltages into a digital signal. A microcontroller 44 receives the digital signal and performs standard checks to insure the signal's magnitude is "valid" (e.g. is within the possible range of the system). It then converts the voltages to X and Y distances in digital form as described herein. Microcontroller 44 is conventional and could be replaced by a system computer. Microcontroller 44 provides a digital signal representative of the X and Y distances to measuring Point 38 to an output port 45 which can be a conventional RS 232 port. Alternatively, microcontroller 44 could translate Point X_1 , Y_1 to any reference Point, such as a Point on base line 25 (Fig. 2).

18 and 33 can be a physical solid surface which is transparent or translucent and can be glass or plastic such as Mylar (Registered Trademark). The surface can be coated with a conductive/resistive substance like indium tin oxide. Other physical surfaces can be used for electromagnetic radiation transmission from the touch position to a reference Point or Points and the distance is determined by the time delay or phase shift. Alternative input screens 18 and 33 can use an ethereal or geometrical surface defined by an electromagnetic, optical or sonar.

Position detection can be accomplished with direct contact closure by resistive, capacitive or inductive means or remote sensing by sonic, electric or magnetic fields (UV, IR, or microwave) scanning.

The advantages of the low power position sensing arrangement over other such screens are: 1) it makes stand-by power requirements minimal; 2) it eliminates distortion due to opposing parallel "bus" bars in conventional touch screens; and 3) when a ramped voltage is employed

programmed with the operating and application programs. battery powered Random Access Memory (RAM) 56 is connected to bus 52 for bidirectional data flow onto bus 52. Microprocessor 50 is a conventional single-chip eight-bit or sixteen-bit device which functions to execute the fixed control programs stored in ROM 54, and further receives control programs from the host and provides control signals to the other electronic components connected to bus 52. Microprocessor 50 may be of the type commonly designated Z80 (manufactured by Zilog Microcomputers, Inc., Cupertino, California), of a type 8088 device (manufactured by Intel Corp. of Santa Clara, California) or any similar powerful microprocessor. ROM 54 may be of the type 2716, 2732, 4764, both manufactured by Texas Instruments of Dallas, Texas. The storage capacity of RAM 56 is determined in part by the sizes of the application programs, the operating programs, and the database. As discussed below, RAM 56 may be of the static SRAM or dynamic DRAM type. The primary requirements for RAM 56 are that it have sufficient storage capacity and that it require a minimum of input power.

7110 and 7114 which have storage capacities of 1 megabit and 4 megabits respectively. (Both are manufactured by Intel). Furthermore, it is possible to use a single integrated circuit chip which includes microprocessor 50, at least part of RAM 56 and at least part of RAM 56.

Also connected to bus 52 is an EIA RS-232C interface 62 which provides a means for inputting and outputting data. Data is provided to bus 52, (usually to RAM 56) by connecting an external data source to RS-232C port 62 directly to microprocessor 50 and other elements of the microcomputer. Offloading data from RAM 56 can also be done by microprocessor 50 to an external computer, other data gathering device or data storage device (e.g. floppy and hard disk drive) in an electronic telecommunications system. In like manner, data can be communicated through port 62 to a printer (not shown) by interconnecting bus 52.

Stylus 16 is used to write on input screen 14 to cause the generation of X,Y coordinate information. The optional touch screen interface electronics circuitry 18 generates coordinate information which is communicated via the bus 52 to the microprocessor 50.

power consuming types.

The X,Y coordinates are input to keyboardless computer 14 via input screen interface 64 and communicated via bus 52 to microprocessor 54 which executes programs stored in ROM 54 and RAM 56.

The number of Points (i.e., sets of X,Y coordinates) used in defining each Handwritten Symbol and the number of Points are identified are important for practical use. It is desirable to use at least about 4 Points per millimeter (or 4 Points per inch) and at least about 100 Points per millimeter (or 100 Points per inch) to define Handwritten Symbols. It is to be noted that the more Points per unit length that are identified, the greater the accuracy of the system in identifying Handwritten Symbols. However, the more Points being identified will slow down the speed of identification and require more computer memory. Therefore, a balance will have to be achieved, based on the available memory and processing ability) of the computer, between the requirement for speed of response and accuracy.

the displayed pixels is sufficient for a display with
pixels in a 225mm (nine inch) horizontal scanline.

When switch 30 (Fig. 2) is positioned to the basic display mode is activated and microcomputer (Fig. 4) programmed by the operating system, causes be displayed on display screen 20 (Fig. 1). The menu various software options. A primary software function editing, functions in a manner similar to conventional processing software with the difference being that characters, symbols and commands are interpreted by as if they were entered from a conventional keyboard. The system is capable of learning the editing symbols of a particular writer for functions such as indent, insert, move and reformat and translates those symbols into command functions. Optionally, Softkeys 28 (Fig. 2) by touching those areas on the input screen with stylus function like conventional hard function keys on a keyboard.

The system is particularly adapted as an interactive screen editor or word processor.

text after the Point where a caret was written in.
moved simply by placing brackets or other user-defined
delimiters around a displayed phrase or word and w
caret or other user-defined symbol in the area of
which the user wishes this material to appear. Ne
be set by drawing vertical lines down the side of
text where the new margins should appear.

The basic editor software also allows ne
be created by simply writing Handwritten Symbols o
All documents can be stored, changed and communica
manner in which these functions are accomplished o
conventional word processing system with the diffe
these functions are accomplished with handwritten
Symbols on the (optional) screen or by touching th
with the stylus. The composite text thus produced
can be subsequently offloaded through the RS 232 p
4) to another computer, a similar device, an exten
gathering device or recording device, into a print
a telecommunications system.

translated into computer text is useful or necessary. Typically, the device can function as a new generation processor, or for use in fields such as sales, nursing, industrial control, census taking, claims adjusting, to name just a few of its many uses or as a general learning and testing device in education. Since the pattern recognition software can learn and translate into computer text languages which are not made up of a small or limited set of alphanumeric characters (e.g., Japanese, Korean, and Chinese), it has particular utility for word processing and telecommunications in these languages.

In practice, it is desirable to use a single computer screen to display forms, Font Symbols or other displays to be edited, and to create a nearby "window" of blank space where Handwritten Symbols are to be written, displayed and identified. The Font Symbols corresponding to the Handwritten Symbols are to be displayed. In this way, the user can view the original, edit and the proposed insert or change without any movement (if any) of the head and eyes. This is important.

will have to store a suitable array of Font Symbols
' conversion of the Handwritten Symbols. Different s
Symbols could be created and stored in the permanen
the computer, as in ROM chip 54. For example, in E
language usage, a chip could contain one (or more)
numbers and letters, suitable punctuation symbols a
appropriate mathematical symbols. Other chips coul
stored Font Symbols for the Arabic, Cyrillic or Gre
alphabets, Japanese, Chinese or Korean "Konji", sym
by architects or engineers, or chemical symbols (e.
rings and the like).

In Fig. 10, one of a series of learning s
displayed and the user is prompted to write the num
through 4. The computer will attempt to match the
numbers with the existing database (if any). If it
matched because there is no existing database or be
is a poor match with an existing database, the char
added to the database. This learning process conti
all of the alphanumeric (or other) characters and s

Fig. 10 also provides an example of the "Softkeys". In addition to the input line, a variety of Softkeys appear. Each Softkey corresponds to a function that can be performed by the system. In order to execute a function, the user merely touches the indicated Softkey. The Softkey will then appear in reverse video. When the selected function is performed. There are numerous advantages to Softkeys over traditional function keys. Some of the significant ones are that the user is no longer required to memorize what function key performs what function. The keyboard overlays is eliminated; and different Softkeys can be made available (displayed and made operational) at different points within a program.

Figures 11A to 11I demonstrates some of the applications and modifications in word processing made possible through this invention. In Fig. 11A a standard screen of a word processor is displayed. The user of the keyboardless entry system indicates that additional information needs to be added and

hereinbelow.

Figures 12A-12G illustrate how a blank is used for a hospital patient. The user of the system calls up the proper blank form (Fig. 12A). This is done, for example, by touching an appropriate Softkey. When the information, in this case a pulse observation, is inserted is touched with the pen (Fig. 12B). At the desired location is highlighted, a "window" appears below the space where the observation is to be recorded (Fig. 12C). The nurse then touches the pen on the match block, which appears highlighted (Fig. 12D). The software then converts the handwritten input to the corresponding Font Symbol and displays the result (Fig. 12E). If there is an exact match, the "insert" block is touched (Fig. 12F), and the observation is added to the patient's records (Fig. 12G). This mechanism is clearly applicable to a wide variety of "forms" in which data is inserted into a form or file. For example, it could be used to correct or update financial information in a spreadsheet program. Other

play Handwritten Symbols and the corresponding Font makes it easy to see errors, when the system "misre written Symbol, and then correct errors quickly and

Referring first to Fig. 5, the overall op functioning of the pattern recognition software wil described. When the operating system calls the pat tion program, the program begins in terminal 75 whe of variables and counters are initialized. The sof proceeds to decision diamond 76 where the program d stylus 16 (Fig. 2) is in contact with input screen 2A). The system provides a "pen down" signal, as s processing box 78, as well as the X,Y coordinate vo locating signals, as described above. Microcompute converts the X,Y coordinate locating signals into S characteristics using programs stored in ROM 54 (Fi separate microcomputer can do the conversion, such microcontroller 44. If a pen down signal is recei software proceeds to processing box 80 where the in

indicated in processing box 86. If a match is no
microprocessor 50 (Fig. 4) causes a message to be
indicated in processing box 88, which requests fu
from stylus on input screen 18 (Fig. 4) by either
entry which is close to a match or a non-recognit

As mentioned above, the software compar
characteristics of each Handwritten Symbol to dat
previously stored in a database. In a preferred
the database is arranged into sections of charact
by the number of Strokes needed to make the chara
symbol. Within each section, the entries are ran
at first, but after use, as explained herein, the
frequently used entries "rise" to the top of the
should be noted that each user will have his or h
particular style of writing a Handwritten Symbol
Handwritten Symbol may have a number of different

For example, many people write the lowe
"h" using a single Stroke. They do this by start
the writing tablet at a Point where they wish to

one particular time by one user has one Stroke (ns). Points (np = 20) and x and y coordinate characteristics: minimum, mean and maximum normalized values (1/80th width) as follows: -17 and -6; 0 and 18; and 19 and 28, respectively. The values in the first vertical column are Point-to-Point slopes, normalized to $360^\circ/256$. The values in the second vertical column are the Point-to-Point average positions above the base line, normalized to 1/80 of width. A typical line width is about 1mm (0.04 inch).

Referring now to Fig. 6, a software hierarchy diagram is depicted. At the top, overseeing the entire computer system 10 (Fig. 1), is an operating system indicated by box 90. Applications programs shown in boxes 92 and 94, residing in RAM 56 (Fig. 4) and ROM 54 (Fig. 4), are executed by microprocessor 50 (Fig. 4) under control of the operating systems. When a Handwritten Character is detected, as indicated by an interrupt, handwriting recognition program 96 is called. A first subroutine, indicated in box 98, converts the X,Y coordinates into Strokes. The characteristics

Points, depicted in box 110, of the position of
2) on input screen 18 (Fig. 2) and encodes these
Strokes as depicted in box 112. The program then
the Strokes by some description set, such as con-
length, curvature, slope, and position of the Str-
depicted in box 114. In box 116 the best compar-
found of the characterized Stroke or sequence of
those in the database. If a sufficiently close
the character is identified in box 118 and the d-
is swapped with the entry above it as shown in b-
this way, the most frequently identified character
to the top of the database and the overall system
as measured in time to find a match, will be inc-
match is not found, the user can add to the bott-
database, as indicated in box 122.

With reference now to Figs. 8 and 9, a
the computer program to recognize a particular S-
is set forth. The computer program begins in te-
proceeds to process the X,Y voltages from proces-

another set of Points is read. If the Point count
mented in processing box 164) indicates that there
than zero Points, the program branches to processing
In processing box 172 the Stroke is identified as
height above the base line (HABL) is calculated in
box 173. From processing box 173 the program proc
processing box 171.

However, if the pen down signal is recei
gram branches to processing box 160 where the volt
scaled to determine the coordinate Point using the
formulas:

$$X = a_1 v_1 + b_1$$

$$Y = a_2 v_2 + b_2.$$

The constants a_1 and b_1 are scaling parameters tha
mined from calibrating the input surface of the pa
play.

If the distance between Points is determined to be outside the two criteria, the program drops the Point and branches back to the top of processing box 152 to process the next pair of coordinate Point voltages.

On the other hand, if the Points fall within the two criteria, the program continues to processing box 160. The Point counter is incremented to keep track of the number of Points. This number is used in decision diamond 162 mentioned hereinabove. The program then continues to processing box 166 where the Points are smoothed using any one of a number of formulae. Smoothing is used to remove noise from digitization, from erratic hand motion, and from electronic noise. The simplest smoothing technique is the multiple Point average which results in calculating the smoothed Point (x_j', y_j') as follows:

involves what is called a spline fit wherein the fo

mula is used:

$$x_j' = (x_{j-1} + 4 x_j + x_{j+1})$$

Any of the foregoing methods can be appli
before or after filtering. The filtering is done s
reduce the number of input Points and to space data
difference and/or angle calculations can be made wi
acceptable random error bounds. A simple process o
sequence of Points by excluding the acceptance of s
Points within a set distance of the previously acce
has been found to be an effective filter.

From processing box 166, the program proc
processing box 168 where the Point is stored in an
is incremented for each new Point since the last pe
signal. Thus, an addressable array of Points is cr
each sequence of Points obtained from a pen down to

of the stream of coordinate Points into a Stroke is primarily on the determining when stylus 16 is "up" contact with the surface of input screen 18. Although the stream of Points can be segmented to form Strokes for other considerations. For example they can be segmented based upon changes in a locally calculated curvature. A large local curvature. Local curvature is calculated as the change in distance along the input coordinates divided by the change in slope. This produces radius of curvature. The radius of curvature changes rapidly with respect to distance along the input coordinates, or if the radius is too small then a segmentation Stroke is assumed to end, and a new one begun. Further segmentation techniques can look for relative maximum and minimum in one or both coordinate values, or curve crossings in the coordinates. However, these methods have been determined to be less effective.

Characterizing a Stroke reduces the sequence of coordinates defining the Stroke or segment to a set of characteristics that are unique, generalized and minimal.

, relative to the writing entry line, to previous Stroke
character extent or center;

(2) Stroke shape is characterized by one or more of the following: average slope, change in slope (which is a measure of curvature) and/or a change in curvature, by sequence of slopes over specific length segments or over fractional length segments, a gross description of linear direction or circular direction, and opening direction;

(3) And Stroke length as characterized by the extent of the stroke along the curve and/or the extent extremum along the coordinate system.

In one embodiment of the present invention, the Stroke is characterized by centroid, extent extremum, and starting and ending coordinates have been successfully used. The Stroke is encoded as a sequence of slopes and vertical positions (relative to Stroke centroid). The Stroke length is approximated by the number of filtered Points. Also

3. The absolute values of each difference are scaled, to make each of the five measurements reasonably equivalent to the others such as lengths scaled to distances between lines.

4. The five thus-determined values are averaged.

5. A predetermined threshold is used as a test of recognition -- too high a value and Font S is infrequently recognized and too low a value causes S to be misidentified. Thresholds of approximately 0.5 are used initially and then switched to approximately 0.7 for improved recognition. If the threshold is exceeded, the comparison is discarded and an error message is displayed.

6. The database is searched to find a minimum difference. If the minimum difference is less than the acceptable threshold for recognition, the corresponding Symbol is displayed on the screen or the command is executed as the case may be.

is eliminated, the program proceeds to a processing
where the address of the next data entry is received
which the program loops back to the top of processing.
In decision diamond 174, the first eliminator is
seeking if the number of Strokes are different. If
of Strokes are the same, the program proceeds to
diamond 176 where the average Height Above Base Line
calculated and compared with the HABL of the data
entry is eliminated if the difference in the average
greater than one-half the height of the entry line.
negative determination in decision diamond 176, the
proceeds to decision diamond 178 where the number
Stroke are compared and the database entry is eliminated
difference in number of Points is greater than ten.
determination varies from that made in decision diamond 176
because it is concerned only with the number of Points
Stroke. However, in decision diamond 174, certain
such as the capital letters "E" and "A", have more
Stroke per letter.

optimize the comparison using off-diagonal elements.

From processing box 182, the program proceeds to decision diamond 186 where a match is determined. In a match is determined by the application of an arbitrary (maximum allowable variance), which is the sum of the absolute values of the differences between the entered Str characterization and that of the stored database entry. From processing box 183, the lower of the present gauge and the previous lower gauge is saved as the best match. The program then goes to decision diamond 184 where a determination is made whether the present entry is the last database entry. If not, the program branches to processing box 180 where the next entry is selected. If it is the last entry, the program proceeds to decision diamond 185 where a determination is made if a match is made on the basis of the gauge being below a predetermined threshold. This threshold is set based on experience with the system.

If no match is obtained, the program branches to decision diamond 188 where a determination is made.

database in processing box 194 and the program branches to the top of processing box 152, or the program branches directly to the top of processing box 152.

On the other hand, if a match is determined at decision diamond 186, the program branches to processing box 196 where the program shuffles the database by interchanging the serial location of the matched entry with the entry at the top of processing box 152. The program then proceeds to processing box 198 where the program zeros the Point counter and the increment counter. The program next proceeds to processing box 198 where the program displays the identified Stroke or Strokes and characterized Stroke or Strokes are displayed by the computer as the identified Font Symbol. This display is at the position in which the entry was made on input (Fig. 2).

From processing box 198, the program proceeds to processing box 200 where the program can act on any character which it has interpreted. An alternative character for the Stroke uses the Points themselves rather than the slope, curvature and position.

processing box 224. From processing box 224, the
ceeds to processing box 226 where the calculated
values for each Point are stored in an addressabl
subroutine then returns to the program through te

When the comparison is made between each
the stored values, the comparison is made by the
Point slope and Point height above base line. As
above, a match is determined by an arbitrary gauge
sum of absolute values of the differences between
Stroke and the stored or dictionary Stroke. The
by adding new Strokes to the dictionary database.
database fills up, those Font Symbols that are in
used are replaced by new entries.

In a working embodiment of the present
algorithm successfully identified upper and lower
and numbers when written discretely from one another.
Handwritten Symbols that are written such that the
continuous, direct extrapolation would require se
database sequentially for one, two, three, etc. S

in Appendix II is written in Microsoft Basic, a computer language available for virtually all microcomputers operating systems. The program is a complete text demonstration, which shows the improvements that can be made upon traditional word processing systems.

Program lines 2600 to 4000 contain the character recognition subroutine which includes the software necessary to get X and Y coordinates. This section of the program corresponds to Fig. 8.

Program lines 2600 to 2699 make up a subroutine designed to obtain the X and Y coordinates of a given character. This code corresponds to boxes 152, 154, 160, 162, 164 and 166 of Fig. 8.

Program lines 3000 to 3339 constitute a Pattern Stroke analysis and characterization routine embodied in boxes 156, 158, 170, 172 and 173 of Fig. 8.

222 of Fig. 9.

Programs lines 3253, 3270-3273 are used the Point height and slope and correspond to box

Program line 3253 is used to store the baseline and embodies box 226 of Fig. 9.

The foregoing program can be stored in a microcomputer of microprocessor with a requirement approximately 25K of machine memory, so that it can be that the use of the program does not use up a lot of memory and is relatively fast in executing the program operation. If the program is written in a language Basic, requiring less memory, such as assembly language, the size of the program can be made smaller.

Boxes 195 to 200 of Fig. 8 appear in the code throughout the code.

A dictionary of the variables of the program section is included as Appendix III.

With reference now to Fig. 13, a flowchart of the editing software ("Editor") demonstrated by Figs

executes the Editing Function. If the character is not to be an Editing Symbol, then the alphanumeric corresponding to the handwritten entry is displayed in processing box 237. In an alternate configuration of Font Symbols will only be accepted when the Editor is in "Insert Mode." This structure insures that each Font Symbol is verified before being added to a document.

The Editor uses a variety of symbols designed for editing on the system similar to, but much more efficient than traditional editing with pencil and paper. These editing functions include, but are not limited to:

DELETE symbol - "_____" A horizontal line drawn through a character or characters. The Editor will delete the underlying characters and reformat the text.

ADJUST MARGINS symbol - "|" A vertical line drawn then the height of one line on the display. The Editor will adjust the margin to the indicated position and reformat the text.

the text.

REPLACE MARKED TEXT - An insert symbol
the marked text displays an input line and replaces
text with the inputted text.

The Editing Symbols described above can be
the particular Editing Symbols preferred by each user
customizing the Editor and preventing new users from
learn unfamiliar Editing Symbols.

Further modifications and enhancements to the
system will be apparent to those skilled in the art.
example, the common characteristics of each Font
be extracted and organized into a synthetic symbol
synthetic symbol's characteristics could then be used to
maximize their variance from all other synthetic symbols.
would create a very compact, optimal database.
hand, as an example, a database created by the designer
preferred embodiment of the invention usually requires
three different characterizations for each symbol.

in CAD/CAM applications involving symbols, geometric shapes, and pen-like style.

We have described a particular preferred embodiment of a keyboardless computer on which usual computer functions are performed by writing in a normal manner with a pen-like input screen placed directly over a flat display. The computer illustrated is ideally configured for use by non-keyboard oriented individuals, by keyboard individuals for whom the use of the computer is enhanced, and in various settings and applications in which keyboard entry is awkward or impossible. Computer-based information and application software can be stored into a portable device for later viewing, manipulation, deletion of data, and adding new text and data in a normal handwriting. Thereafter the user may transmit this computer text to another computer, a similar device, an external electronic storage device, a hard copy printer, or through a telecommunications system. The computer is capable of recognising Handwritten Symbols with a high degree of accuracy and of "learning" individual styles of handwriting. The data and commands can be input with a pen or stylus.

'H' (1) ns = 2 x = (-27..0..37) y = (-20..18..61)
1) np = 14 x = (-27..-12..-2) y = (-20..22..)

-9,	37
4,	38
-39,	32
-48,	25
-58,	16
-71,	8
-72,	2
-64,	-4
-89,	-11
-72,	-17
-80,	-24
-89,	-30
-80,	-36

2) np = 20 x = (-21..10..37) y = (-15..16..)

27,	43
-80,	37
-71,	29
-98,	23
-80,	16
-78,	8
-85,	1
-89,	-6
-80,	-12
-80,	-18
-64,	-25
-64,	-31
48,	-25
56,	-18
48,	-12
64,	-6
89,	1
-124,	0
-124,	-2

```

170 IF EDLN <> 1 THEN GOSUB 6100
180 IF EDLN <> 1 THEN GOSUB 12500
200 REM EDIT
205 LINY=0
210 IF SWTPN = -1 THEN GOSUB 7100 ELSE GOSUB 2600 :REM TOUCH T
II @ 9600
211 IF SWTPN =1 THEN GOSUB 7300
220 REM RECOGNIZE SYMBOLS
221 IF NP > 2000 THEN LOCATE 23,1: PRINT "TOO MANY POINTS" : ST
222 IF NP < 4 THEN GOTO 200
230 GOSUB 3000 :GOSUB 3400 :REM -, <> ^, |, ^, \, /, P, E
240 ON NCHAR GOTO 1200,1300,1700,6200,6700,6500,6600,12500,6900
MOVE,FORMAT,INSERT,WRITE OVER,REMOVE,PARAGRAPH,INSERT CHARACTER
250 GOTO 200
500 END
600 REM DISPLAY TEXT FROM SINGLE CHARACTER ARRAY
605 LOCATE 23,20: PRINT " ",
610 I=II : J=JJ : REM INITIAL DISPLAY POINT FOR NC1
620 FOR K=NC1 TO NC
630 IF A$(K) <> CR$ THEN GOTO 660
640 LOCATE J,I : PRINT " " : I=I+1 : IF I < 81 GOTO 640
650 J=J+1 : I=IMARGN : LL(J)=K : GOTO 740
660 IF I< IMARGX THEN GOTO 670 ELSE IF A$(K)=" " GOTO 690
670 IF J> JBTM THEN LOCATE 21,20: PRINT "READY", :RETURN
680 LOCATE J,I : IF I=IN THEN GOTO 710 ELSE PRINT A$(K) : I=I+1
690 LOCATE J,I : PRINT " " : I=I+1: IF I< 81 GOTO 690
700 J=J+1 : I=IMARGN : LL(J)=K+1 : GOTO 740
710 IF A$(K) <> " " THEN GOTO 730
720 IF LL(J)=K GOTO 740
730 I=I+1 : PRINT A$(K)
740 IF I> IMARGX+10 THEN GOTO 690 ELSE NEXT K
750 IF J> JTX THEN GOTO 790
760 JS=J
765 LOCATE J,I : PRINT " ": I=I+1: IF I< 81 GOTO 765
770 J=J+1 : I=IMARGN: IF J > JTX THEN GOTO 780 ELSE GOTO 765

```

```

1350 IF A$(ML)="" THEN M1=M1+1
1360 IF A$(ML)=CR$ THEN M1=M1+1
1390 XX=SXX/2: XN=SXN(2): YX=SYX(2): YN=SYN(2)
1400 XP=(XN+XX)/2: YP=(YN+YX)/2 :REM AVE BRACKET POSITION
1410 I=INT(XP/8)+1: J=INT(YP/8)+1:REM CONVERT X,Y TO I,J
1420 ML=LL(J): M2=ML+I-IN :REM CALC A ARRAY INDEX OF LETTER AT I,J
1430 IF A$(ML)="" THEN M2=M2+1
1440 IF A$(ML)=CR$ THEN M2=M2+1
1450 M3=0:REM INITIATE COUNTER
1460 FOR M=M1 TO M2 :REM LOOP OVER PORTION CUT
1470 M3=M3+1: REM INCREMENT COUNTER
1480 AS$(M3)=A$(M) :NEXT M :REM COPY CUT TEXT
1490 XX=SXX(3): XN=SXN(3): YX=SYX(3): YN=SYN(3)
1500 REM INSERT WITH M3 LETTERS STORED IN AS$ & xx,xn,yx,yn
1520 XP=(XN+XX)/2: YP=(YN+YX)/2 :REM AVE BRACKET POSITION
1530 I=INT(XP/8)+1: J=INT(YP/8)+1:REM CONVERT X,Y TO I,J
1540 ML=LL(J): M3=ML+I-IMARGN :REM CALC A ARRAY INDEX OF LETTER AT
1550 IF A$(ML)="" THEN M3=M3+1
1560 IF A$(ML)=CR$ THEN M3=M3+1
1570 MD=M2-M1+1 : IF MD > 0 THEN GOTO 1580 ELSE GOTO 200
1580 FOR M=M1 TO NC : REM LOOP OVER A$ ARRAY POSITIONS TO BE SHIFTED
1590 A$(M)=A$(M+MD) : REM SHIFT A$ ARRAY LETTERS
1600 IF M+MD = NC THEN GOTO 1610 ELSE NEXT M : REM STOP AT END OF A$
1610 IF M3 > M2 THEN M3=M3-MD :REM OFF SET INSERTION POINT TO ACCOUNT
ON
1620 FOR M=NC-MD TO M3 STEP -1 : REM SPRED ARRAY TO MAKE ROOM FOR INSERT
1630 A$(M+MD)=A$(M) :REM MOVE LETTERS
1640 NEXT M
1650 FOR M=1 TO MD :REM LOOP OVER CHARACTERS
1660 A$(M3+M-1)=AS$(M) :REM INSERT LETTERS
1670 NEXT M
1680 CLS: II=IMARGN: JJ=JTOP: NC1=1 : GOSUB 600
1690 GOTO 200
1700 REM REFORMAT MARGINS

```

```

2698 IF DEBUG (<) 0 THEN LOCATE 20,12 :PRINT NP
2699 RETURN
3000 REM CALCULATE CHARACTERIZATION OF STROKES
3005 REM GOSUB 4200 :REM 2 POINT AVERAGE POINTS
3010 A1=0: A2=0: DA1=0: DA2=0: SEND=0: ZX=0: ZY=0: ZA=0: ZDA=0: ZL
3015 NSTRK=0: XX=ABS(X(1)): XN=XX: YX=Y(1): YN=YX: M1=1: M2=1: M3=
3020 FOR M=2 TO NP
3025 IF X(M) < 0 THEN SEND=2 :X(M)=-X(M)
3030 IF M > M1 GOTO 3035 ELSE 3391
3035 IF X(M1) < 0 THEN M1=M: GOTO 3391
3040 DX=X(M)-X(M1): DY=Y(M)-Y(M1): ADX=ABS(DX): ADY=ABS(DY): DS=AD
3043 IF AY > 0 THEN DY=-DY
3044 IF DS < 3 THEN GOTO 3391
3052 IF X(M) < XN THEN XN=X(M)
3053 IF X(M) > XX THEN XX=X(M)
3054 IF Y(M) < YN THEN YN=Y(M)
3056 IF Y(M) > YX THEN YX=Y(M)
3058 ZX=ZX+X(M)*DS: ZY=ZY+Y(M)*DS: ZL=ZL+DS: NSP=NSP+1
3060 IF ADX < ADY THEN GOTO 3080 :REM ENCODE ANGLE 0-255 +-n256
3070 IF DX > 0 THEN A1=64-(32*DY)\DX ELSE A1=192-(32*DY)\DX
3075 GOTO 3085
3080 IF DY > 0 THEN A1= (32*DX)\DY ELSE A1=128+(32*DX)\DY
3085 IF NSP>M2 THEN GOTO 3090
3086 IF A1<0 THEN A1=A1+256
3087 ZA=ZA+A1*DS
3089 GOTO 3190
3090 IF A1-A2 < -127 THEN A1=A1+256 : GOTO 3090
3095 IF A1-A2 > 128 THEN A1=A1-256 : GOTO 3095
3100 ZA=ZA+A1*DS
3110 DA1=A1-A2
3115 ZDA=ZDA+DA1
3120 IF NSP > M3 THEN ADA=(DA2+ADA)/2 ELSE ADA=DA1 : ASTRT(NSTRK+1
3130 IF NSP = M3+1 THEN ASTRT(NSTRK+1)=(A1+A2)\2
3140 IF ABS(DA1-ADA) > 80 THEN SEND=1
3150 IF ABS(DA1) > 64 THEN SEND=1
3190 IF SEND=0 THEN GOTO 3300
3200 REM END OF STROKE

```

```

3280 IF DBUG > 3 THEN LPRINT USING "#####";M,NSP,X(M),Y(M),ZL,ZA,
,(A1+A2+A3+A4)\4 ELSE IF DBUG < -3 THEN PRINT USING "#####";M,NSP
ZA,A1,DA1,ADA,ZDA,(A1+A2+A3+A4)\4
3281 IF DBUG > 2 THEN LPRINT "SL,SA,SC,AS,AE",SL(NSTRK) SA(NSTRK)
T(NSTRK) AEND(NSTRK) ELSE IF DBUG < -2 THEN PRINT "SL,SA,SC,AS,AE"
NSTRK) SC(NSTRK) ASTRT(NSTRK) AEND(NSTRK)
3285 A1=0: A2=0: DA1=0: DA2=0: DS2=0: DS1=0: SEND=0: ZX=0: ZY=0: ZA=0
3290 XX=ABS(X(M)): XN=XX :YX=Y(M): YN=YX: NLONG=0 :NSP=0
3295 GOTO 3391
3300 REM FINISH LOOP
3310 M1=M
3320 DA3=DA2: DA2=DA1: DS2=DS1: DS1=DS: A4=A3: A3=A2: A2=A1
3390 IF DBUG > 3 THEN LPRINT USING "#####";M,NSP,X(M),Y(M),ZL,ZA,
,(A1+A2+A3+A4)\4 ELSE IF DBUG < -3 THEN PRINT USING "#####";M,NSP
ZA,A1,DA1,ADA,ZDA,(A1+A2+A3+A4)\4
3391 NEXT M
3392 IF DBUG > 1 THEN LPRINT "      I      NO      X      Y      L      A      C
X      YN      ST      ND"
3393 IF DBUG < -1 THEN LOCATE 9,1 :PRINT "      I      NO      X      Y
XX      XN      YX      YN      ST      ND"
3394 IF DBUG < -1 THEN LOCATE 10,1 :FOR I=1 TO NSTRK:PRINT USING "
),SX(I),SY(I),SL(I),SA(I),SC(I),SXX(I),SXN(I),SYX(I),SYN(I),ASTRT(
XT I
3395 IF DBUG > 1 THEN FOR I=1 TO NSTRK: LPRINT USING "#####";I,SN
I),SL(I),SA(I),SC(I),SXX(I),SXN(I),SYX(I),SYN(I),ASTRT(I),AEND(I)
3396 IF NSTRK > 1 THEN GOSUB 4000 ELSE GOTO 3399
3397 IF DBUG < -1 THEN FOR I=1 TO NSTRK:PRINT USING "#####";I,SNO(
,SL(I),SA(I),SC(I),SXX(I),SXN(I),SYX(I),SYN(I),ASTRT(I),AEND(I) :N
3398 IF DBUG > 1 THEN FOR I=1 TO NSTRK: LPRINT USING "#####";I,SN
I),SL(I),SA(I),SC(I),SXX(I),SXN(I),SYX(I),SYN(I),ASTRT(I),AEND(I)
3399 RETURN
3400 REM EDITING SYMBOL RECOGNITION
3410 FOR L=1 TO NSTRK
3412 A1=(SA(L)+32)\64: IF A1 > 3 THEN A1=0
3416 IF A1 = 0 THEN IF SL(L) > 100 THEN NCHAR=3: RETURN: REM VERTI
3418 IF A1 = 1 THEN IF SL(L) > 60 THEN NCHAR=1: RETURN: REM HORIZ
3420 IF A1 = 2 THEN IF SL(L) > 100 THEN NCHAR=3: RETURN: REM VERTI
3422 IF A1 = 3 THEN IF SL(L) > 60 THEN NCHAR=1: RETURN: REM HORIZ

```

```

3629 ICONT=L1 :GOTO 3635
3630 IF SXN(L-L1) > SXX(L)+LHT4 THEN GOTO 3635 ELSE GOTO 3620
3635 NEXT L
3640 IF ICONT=0 THEN NLET=NLET+1 :SLET(L)=NLET: GOTO 3649
3645 NLET=SLET(L-ICONT) : SLET(L)=NLET :SLET(L-1)=NLET
3646 IF DBUG > 3 THEN LPRINT "L,NLET,SLET",L,NLET,SLET(L-1),SLET(L)
3649 NEXT L
3650 REM LOOP OVER LETTERS
3652 IF ED=1 THEN ZLINY=0: FOR LMN=1 TO NSTRK : ZLINY=ZLINY+SYX(L)
EXT LMN : LINY=ZLINY/NSTRK :LINY=(1+LINY\16)*8+4 :IF DBUG > 0 THEN
",LINY ELSE IF DBUG < 0 THEN LOCATE 22,1 :PRINT "LINY=",LINY
3655 MCHR=1: NST=0: XX=-1: XN=1000: XX1=-100 :YX=-1 :YN=1000 :YX1=-100
3656 FOR LSTRK=1 TO NSTRK+1
3657 LST=LSTRK :IF LSTRK = NSTRK+1 THEN GOTO 3659
3658 IF SLET(LSTRK) = MCHR THEN NST=NST+1: GOTO 3690
3659 IF NST > 10 THEN LOCATE 23,1 : PRINT "TOO MANY STROKES", :STOP
3660 FOR M=1 TO NST :LMN=LST-NST+M-1
3661 DBC(1,M)=SL(LMN): DBC(2,M)=SA(LMN): DBC(5,M)=SC(LMN)
3662 DBC(6,M)=(100*(SX(LMN)-SX(LST-NST)))\LHT :DBC(3,M)=ASTRT(LMN)
D(LMN)
3663 DBC(7,M)=(100*(SY(LMN)-LINY))\LHT : DBC(9,M)=( 50*(32*Y(SNO(LMN)-
INY))\LHT: IF LMN =1 THEN DBC(8,M)=( 50*(32*Y(2)\AY+BY-LINY))\LHT
( 50*(32*Y(SNO(LMN-1)+2)\AY+BY-LINY))\LHT
3664 IF YX < SYX(LMN) THEN YX=SYX(LMN)
3665 IF YN > SYN(LMN) THEN YN=SYN(LMN)
3666 IF XX < SXX(LMN) THEN XX=SXX(LMN)
3667 IF XN > SXN(LMN) THEN XN=SXN(LMN)
3668 IF DBUG > 0 THEN LPRINT USING "#####";MCHR,DBC(1,M),DBC(2,M),
,M),DBC(5,M),DBC(6,M),DBC(7,M),DBC(8,M),DBC(9,M)
3669 IF DBUG < 0 THEN LOCATE 10+M,27: PRINT USING "#####";MCHR,DBC(1,M),
,DBC(3,M),DBC(4,M),DBC(5,M),DBC(6,M),DBC(7,M),DBC(8,M),DBC(9,M)
3670 NEXT M :GOSUB 3700
3671 NPOS=(2+(XX+XN)\16-IMARGN)\2
3672 REM PRINT "MCHR,NPOS,CHAR,ED" ,MCHR,NPOS,CS$,ED :INPUT IJK
3673 IF ED = 1 THEN NPOS=0: AS$(MCHR)=CS$
3674 IF NPOS > 0 THEN IF NPOS < 40 THEN AS$(NPOS)=CS$
3675 REM
3676 IF ED <> 1 THEN LOCATE (LINY-LHT+1)\8,(XX+XN)\16+1 :PRINT CS$

```

```

3803 CS$=CHR$( DC(MS1)) .
3804 REM
3805 IF DBUG <> 0 THEN LOCATE 19,1 :PRINT R1,MS1,CS$
3806 REM
3810 IF R1\NST > ACPT GOTO 3900 :REM CHARACTER NOT FOUND
3830 RETURN
3900 REM LEARN NEW CHARACTER
3902 IF LERN = 0 THEN CS$=CHR$(2) : RETURN
3904 IF KYBD=0 THEN LOCATE 23,1 :INPUT "CHARACTER NOT UNDERSTOOD,
IN: ",CS$
3906 IF KYBD = 0 THEN LOCATE 23,1 :PRINT "

3910 IF KYBD=1 THEN GOSUB 11400
3912 IF KYBD =1 THEN GOSUB 9600
3914 IF KYBD =1 THEN GOSUB 11600 : CS$=CHR$(MSEL)
3915 IF CS$="~" THEN GOTO 3960
3917 IF ED=1 THEN RETURN
3920 IF NDBT = 300 THEN PRINT "Too Many Symbols": STOP
3926 NDBT=NDBT+1 : DN(NDBT)=NST : DC(NDBT)=ASC(CS$)
3930 IF M3+NST > 800 THEN PRINT "Too Many Strokes" :STOP
3936 FOR M1=1 TO NST : FOR M2=1 TO 9
3940 DIM2,M1+M3)=DBC(M2,M1) : NEXT M2 : NEXT M1
3950 RETURN
3960 CS$=" "
3980 RETURN
4000 REM CONCATENATE STROKES
4010 FOR L=2 TO NSTRK
4014 IF X(SNO(L-1)) < 0 THEN GOTO 4090
4016 IF SL(L-1) < 4 THEN GOTO 4100
4020 ALNO=(ABS(ABS(X(SNO(L)))-ABS(X(SNO(L-1))))+ABS(Y(SNO(L))-Y(SNO(L-1))))
4021 REM LOCATE 1,1: PRINT "L,ALNO",L,ALNO
4022 IF ALNO > 100 THEN GOTO 4090 : REM TOO FAR AWAY
4024 ALNO= 8*ALNO/(SNO(L)-SNO(L-1))
4026 DD =ABS(ABS(X(SNO(L-1)))-ABS(X(SNO(L-1)+2)))+ABS(Y(SNO(L-1))-Y(SNO(L-1)+2))
4027 REM PRINT "L,DD,ALNO",L,DD,ALNO
4028 IF DD > ALNO THEN GOTO 4090 : REM TOO FAR AWAY
4030 IR1=80*SC(L)\SL(L) : IR2=80*SC(L-1)\SL(L-1)
4031 REM PRINT "L,IR1,IR2",L,IR1,IR2

```

```

4133 SY(L1)=(SL(L1)*SY(L1)+SL(L)*SY(L))/SLL
4137 SA(L1)=(SL(L1)*SA(L1)+SL(L)*SA(L))/SLL
4140 SL(L1)=SLL +DD
4150 SC(L1)=SC(L1)+SC(L)+((SC(L1)+SC(L))*DD)\SLL
4160 SNO(L1)=SNO(L) :AEND(L1)=AEND(L)
4170 NSTRK=NSTRK-1 : IF NSTRK < L THEN RETURN
4180 FOR LL=L TO NSTRK
4185 L1=LL+1 :SXX(LL)=SXX(L1) :SXN(LL)=SXN(L1) :SYX(LL)=SYX(L1) :
4187 SX(LL)=SX(L1) :SY(LL)=SY(L1) :SA(LL)=SA(L1) :SC(LL)=SC(L1) :
4189 SL(LL)=SL(L1) :ASTRT(LL)=ASTRT(L1) :AEND(LL)=AEND(L1)
4190 NEXT LL
4195 GOTO 4010
4200 REM 2 POINT AVERAGE POINTS
4210 M2=0
4220 FOR M=2 TO NP STEP 2 :M1=M-1
4230 IF X(M1) > 0 GOTO 4240
4235 M2=M2+1 :X(M2)=X(M1) :Y(M2)=Y(M1) :GOTO 4280
4240 IF X(M) > 0 GOTO 4250
4245 M2=M2+1 :X(M2)=X(M) :Y(M2)=Y(M) :GOTO 4280
4250 IF ABS(X(M)-X(M1)) > 30 THEN GOTO 4280
4260 IF ABS(Y(M)-Y(M1)) > 30 THEN GOTO 4280
4270 M2=M2+1 :X(M2)=(X(M)+X(M1))\2 :Y(M2)=(Y(M)+Y(M1))\2
4280 NEXT M
4283 X(M2)=X(NP)
4285 NP=M2
4290 RETURN
4340 REM ORDER STROKES BY SXN
4342 XCH=0 :IF NSTRK < 2 THEN RETURN
4345 FOR L=2 TO NSTRK
4350 IF SXN(L) > SXN(L-1) THEN GOTO 4380
4355 XCH=1 :L1=L-1
4360 ST=SXX(L1) :SXX(L1)=SXX(L) :SXX(L)=ST
4361 ST=SXN(L1) :SXN(L1)=SXN(L) :SXN(L)=ST
4362 ST=SYN(L1) :SYN(L1)=SYN(L) :SYN(L)=ST
4363 ST=SYX(L1) :SYX(L1)=SYX(L) :SYX(L)=ST
4364 ST=SX(L1) :SX(L1)=SX(L) :SX(L)=ST
4365 ST=SY(L1) :SY(L1)=SY(L) :SY(L)=ST
4366 ST=SA(L1) :SA(L1)=SA(L) :SA(L)=ST

```

```

6202 XP=(XN+XX)/2: YP=(YN+YX)/2 :REM AVE CHARACTER POSITION
6203 I=INT(XP/8)+1: J=INT(YP/8)+1 :ISAV=I: JSAV=J: REM CONVERT X,Y
6204 ML=LL(J): M1=ML+I-IMARGN :REM CALC A ARRAY INDEX OF LETTER AT
6205 IF A$(ML)=" " THEN M1=M1+1
6206 IF A$(ML)=CR$ THEN M1=M1+1
6207 MSAV=M1
6210 JJ=JSAV : FOR I=1 TO 40 :A$(I)=" " :NEXT I
6215 LINY=8*(JJ+3)-1
6220 LINE (0,LINY-LHT-8)-(639,LINY+LHT\2),0,BF
6230 LINE (8*IMARGN-1,LINY)-(639,LINY-LHT),1,B
6240 LINE (607,LINY)-(639,LINY-LHT),1,BF
6250 IF SWTPN = -1 THEN GOSUB 7100 ELSE GOSUB 8600 :REM TOUCH TECH
CII @ 9600
6251 IF SWTPN = 1 THEN GOSUB 7300
6255 IF NP < 8 THEN GOTO 6270
6256 GOSUB 3000
6257 GOSUB 3600
6258 IF DBUG <> 0 THEN LOCATE 22,IMARGN: FOR I=1 TO 40: PRINT A$(I)
6259 GOTO 6250
6270 JB=0: J=1
6271 REM LOCATE 16,1: FOR I=1 TO 40: PRINT A$(I);: NEXT I
6272 IE=2 :FOR I=40 TO 2 STEP -1 : IF A$(I) <> " " THEN IE=I+1 : G
6273 NEXT I
6274 FOR I=2 TO IE
6275 IF A$(I) <> " " THEN GOTO 6280
6276 JB=JB+1
6278 IF JB < 3 THEN GOTO 6288
6280 J=J+1: JB=0
6282 A$(J)=A$(I)
6288 NEXT I
6289 REM LOCATE 17,1: FOR I=1 TO 40: PRINT A$(I);: NEXT I
6290 A$(J+1)=" "
6295 IF J < 2 THEN IF A$(1) =" " THEN LOCATE 18,1 : PRINT "NO LETT
N" :GOTO 6210
6300 MD=J+1 : M3=MSAV : I=ISAV : J=JSAV
6310 REM LOCATE 20,1: FOR I=1 TO 40: PRINT A$(I);: NEXT I
6315 REM INPUT IJK
6320 GOTO 1620

```

```

6950 IF A$(ML)="" THEN M1=M1+1
6960 IF A$(ML)=CR$ THEN M1=M1+1
6970 ASS(2)=A$(1)
6980 M3=M1 :MD=1
6990 GOTO 1570
7000 REM SET UP RERUN
7010 OPEN "I",#1,"POINTS.DAT"
7020 INPUT #1,AX,BX,AY,BY
7030 RETURN
7100 REM READ POINTS
7110 INPUT #1,NP
7120 FOR N=1 TO NP
7130 INPUT #1,X(N),Y(N)
7132 NEXT N
7133 IF IFAVE = 1 THEN GOSUB 4200
7134 FOR N=1 TO NP
7135 IF X(N) > 0 THEN I=32*X(N)/AX+BX ELSE I=-32*X(N)/AX+BX
7136 J=32*Y(N)/AY+BY
7137 PSET(I,J)
7140 NEXT N
7148 IF DBUG > 5 THEN FOR N=1 TO NP: LPRINT N,X(N),Y(N):NEXT N
7149 IF DBUG <> 0 THEN LOCATE 20,2 :PRINT "NP=",NP
7150 RETURN
7200 REM SET UP STORE POINTS
7210 OPEN "O",#1,"POINTS.DAT"
7220 PRINT #1,AX,BX,AY,BY
7230 RETURN
7300 REM STORE POINTS
7310 PRINT #1,NP
7320 FOR N=1 TO NP
7330 PRINT #1, USING "#####"; X(N),Y(N)
7340 NEXT N
7350 RETURN
8600 REM TT ACQUIRE POINTS IN ASCII @ 9600 WITH INSERT
8603 M=20
8605 IF LOC(2) > 0 THEN INPUT$=INPUT$(1,2) : GOTO 8605
8606 PRINT #2,"G" :N=0
8610 IF LOC(2) < 40 THEN GOTO 8610

```

```

9660 II=32*XP\AX+BX+II: JJ=32*YP\AY+BY+JJ: N=N+1
9670 IF N < 10 THEN GOTO 9674
9672 I=II\10: J=JJ\10: N=0 :II=0: JJ=0
9673 LINE (I+1,J+1)-(I-1,J-1),1,B
9674 IF LOC(2) > 9 GOTO 9640
9675 MM=0
9680 MM=MM+1: IF LOC(2) > 9 THEN GOTO 9640 ELSE IF MM < M THEN GOT
9690 IF LOC(2) > 9 THEN GOTO 9610 ELSE NU=NU+1 :IF NU < 1500 THEN
9695 PRINT #2,"S"
9699 RETURN
10700 REM SET UP TOUCH SCREEN FOR ASCII @ 9600
10710 OPEN "I",#2,"CAL.DAT"
10720 INPUT #2,AX,BX,AY,BY
10730 CLOSE #2
10740 OPEN "COM1:9600,N,8,1,CS0,DS0" AS #2
10750 PRINT #2,"S"
10760 RETURN
11000 REM UTILITIES
11100 REM input selection menus
11101 REM mnum= no. of items
11102 REM mtitl$=titles
11103 REM mi0,mie,mj0,mje=start & end points for boxes
11105 IF KYBD=1 GOTO 11400
11110 LINE(MI0,MJ0)-(MIE,MJE),0,BF :LINE(MI0,MJ0)-(MIE,MJE),1,B
11115 XX=MI0 : MRATO=(MIE-MI0)\MNUM
11120 FOR K=1 TO MNUM : LINE (XX,MJ0)-(XX,MJE)
11130 MJA=(MJ0+MJE+8)\16+1 : LOCATE MJA,(XX+20)\8 : PRINT MTITL$(K)
11140 XX=XX+MRATO :NEXT K
11160 RETURN
11300 REM DETERMINE MENU ITEM SELECTED
11310 IF KYBD=1 GOTO 11600
11320 XX=0 : FOR K=1 TO MNUM : XX=XX+MRATO : IF I < XX THEN GOTO 1
11330 NEXT K
11340 MSEL=K
11350 RETURN
11400 REM DISPLAY KEYBOARD
11405 JMENU=163

```

```

11780 IF MSEL=34 THEN MSEL=44 : RETURN
11790 IF MSEL=35 THEN MSEL=59 : RETURN
11800 IF MSEL=36 THEN MSEL=58 : RETURN
11810 IF MSEL=37 THEN MSEL=34 : RETURN
11820 IF MSEL=38 THEN MSEL=36 : RETURN
11830 MSEL=38 : RETURN
11850 RETURN
12000 REM input output
12100 REM read in text
12110 OPEN "I", #1,"TEXT" :NC=0
12120 IF EOF(1) THEN CLOSE #1 : GOTO 12160
12130 INPUT #1, A1$ :KK=0
12140 FOR K=1 TO LEN(A1$) : KK=KK+1 :A$(NC+KK)=MID$(A1$,K,1)
12143 IF LEN(A$(NC+KK))=0 THEN KK=KK-1
12145 NEXT K
12150 NC=NC+KK:IF NC < 1500 THEN GOTO 12120
12155 PRINT "TOO MUCH TEXT" : STOP
12160 REM TEXT EDITOR INITIALIZATION
12170 II=IMARGN : JJ=JTOP : NC1=1 : REM DISPLAY ALL TEXT
12175 CLOSE #1
12180 RETURN
12500 REM SAVE SYMBOL DATA BASE
12505 IF SWTPN <> 0 THEN CLOSE #1
12510 OPEN "O",#1,"SYM.DAT"
12520 PRINT #1,NDBT
12530 M2=0 :FOR M=1 TO NDBT
12540 PRINT #1, DN(M) DC(M)
12550 FOR M1=M2+1 TO M2+DN(M)
12560 PRINT #1, D(1,M1) D(2,M1) D(3,M1) D(4,M1) D(5,M1) D(6,M1)
12570 NEXT M1 :M2=M2+DN(M) : NEXT M
12580 CLOSE #1
12590 END
12700 REM RESTART READ IN DATA BASE
12710 OPEN "I",#1,"SYM.DAT"
12720 INPUT #1,NDBT
12730 M2=0 :FOR M=1 TO NDBT
12740 INPUT #1,DN(M),DC(M)
12750 FOR M1=M2+1 TO M2+ DN(M)

```


M - Pointer to Point being considered
 NP - number of Points
 X - array of X coordinate
 Y - array of Y coordinate
 DX - change in X
 DY - change in Y
 ADX - absolute value of DX
 ADY - absolute value of DY
 DS - pseudo arc length $ADX + ADY$
 AY - calibration multiplier for vertical (Y) direction
 AY = 0 mean coordinate system true downward
 ADA - running average change in angle
 DA2 - previous change in angle
 ASTRT - array for starting angle of Stroke
 SZL - single precision Stroke length
 SNO - array of number of the 1st Point for each
 LHT5 - height between lines divided by 5 in display
 used as a measure
 SL - array of Stroke lengths scaled to height between lines
 AEND - array of angles at end of Strokes
 SC - array of angle changes for each stroke
 AZA - average angle through Stroke
 SX - array of averages centroid X coordinate for each Stroke
 SY - array of averages centroid Y coordinate for each Stroke
 SA - array of averages angle (slope) coordinate for each Stroke
 SXX - array of maximum X coordinate for each Stroke
 SXN - array of minimum X coordinate for each Stroke
 SYN - array of minimum Y coordinate for each Stroke
 SYX - array of maximum Y coordinate for each Stroke
 DEBUG - debugging flag for printing
 INPUT - string of input bytes to be rejected
 XP, YP - input coordinates
 XPA - average 2 Points in X direction
 YPA - average 2 Points in Y direction
 XPO - initial X average on pan down
 YPO - initial Y average on pan down
 I - display pixel corresponding to X coordinate

symbols comprising predetermined font symbols and at least one command symbol;

means coupled to the processing means and to the display screen for displaying on the display screen a symbol associated with the handwritten symbol if the handwritten symbol is identified as a font symbol; and

means coupled to the processing means and to the display screen for executing an associated command if the handwritten symbol is identified as a command symbol.

2. Apparatus according to claim 1, further comprising means coupled to the digitising means and to the display screen for displaying on the display screen an image of the handwritten symbol as it is written;

3. Apparatus according to claim 1 or 2, wherein the processing means including a database of characteristics of the predetermined symbols, and means for comparing characteristics of the handwritten symbol with those in the database to assess the identity of each handwritten symbol.

6. Apparatus according to any of claims 1 to 5, the predetermined command is an editing command.

7. Apparatus according to claim 6, wherein said predetermined command symbol is a caret symbol.

8. Apparatus according to claim 6, wherein said predetermined command symbol is the writing of one horizontal or vertical lines through displayed font said screen.

9. Apparatus according to claim 6, 7 or 8, further comprising means for changing a first group of computer generated font symbols in text displayed on said screen to a second group of computer generated font symbols, wherein a handwritten symbol corresponds to the editing command.

10. Apparatus according to claim 6, 7, 8 or 9, wherein a plurality of predetermined symbols includes a plurality of editing commands.

other information, for entering handwritten symbols.

13. Apparatus according to claim 12, wherein said window can be of various sizes and shapes.

14. Apparatus according to claim 12 or 13, wherein said window can be used to insert graphic images as well as handwritten symbols.

15. Apparatus according to any preceding claim, wherein said apparatus comprises a portable unit having dimensions to about 400mm by 400mm by 100mm, a weight of up to about 1 kilograms and is adapted to include a self-contained power supply.

16. Apparatus according to any preceding claim, wherein predetermined font symbols include computer generated symbols.

17. Apparatus according to any preceding claim, including

processing means comprises means for determining, for each stroke of a symbol (i) the length of the stroke, (ii) the average slope of the stroke, (iii) the stroke centre height above a predetermined base line, (iv) the rate of change in the slope of the stroke, and (v) the location of the centroid of the stroke compared to the location of the centroid of the handwritten symbol.

21. A computer system including the apparatus of claims 1 to 20.

22. A method of receiving a handwritten character comprising:

using a hand held means to write or draw a handwritten symbol on or over at least a portion of a display screen;

digitising the signals from the display screen to produce a digital representative of the handwritten symbol;

processing the digitised signal to determine the identity of the handwritten symbol as one of a plurality of symbols.

substantially as herein described with reference to
drawings.

25. A method of receiving a handwritten character,
substantially as herein described with reference to
drawings.
