

## Description

The present invention relates to an instruction input system according to the precharacterizing part of claim 1 and to a method for inputting instructions to a graphic or word processor according to the precharacterising part of claim 9.

Such an instruction input system and method is known from Us-Patent 44 30 526, which discloses an interactive graphics transmission system for use in video teleconferencing. A graphic to be transmitted is scanned by a video camera and the respective video signals are transmitted by a transmitter to far end graphic communication terminals. The graphic to be transmitted can be manipulated in various ways by use of a light pen. The system recognizes the current position of the light pen by receiving infrared pulses from the light pen. The type of instruction, for example an erase or write instruction, is detected by the system by analyzing the mode under which the light pen is used, i.e. whether it is used within a certain area or not, whether it is used with the one or the other end tilted downwards to the graphic or whether the tip of the light pen is pressed or not.

GB-21 56 114 describes a computing apparatus with a touch screen operator interface, which is arranged to emulate in function and appearance a data-processing instrument such as a calculator. The calculator is displayed on a video screen which is covered by a transparent touch screen device. Calculation can be carried out by pressing the touch screen device at positions in which the respective keys of the calculator are displayed. The result of the calculation is shown on the display field of the displayed calculator.

Japanese Patent Abstract 58-64 536 describes an input system for inputting characters by finger-activating certain locations on a transparent touch operation surface arranged on a display screen.

One of the various user interface problems is how a user or an operator can easily input the necessary information and instructions, especially complex ones, to a computer, i.e., a problem of operability.

Conventional techniques have not yet found a satisfactory solution to this problem.

A wordprocessor will be exemplified to explain the above problem in greater detail. When a wordprocessor is used to perform edit processing (such as delete, insert, replace, and frame operations in conjunction with a text), unless a user specifies the type of instruction and the precise area within the text where the designated instruction is to be executed, the wordprocessor cannot select the necessary processing and hence cannot execute processing, with the result that the user's request cannot be satisfied.

For this reason, in conventional wordprocessors, personal computers, and the like, as disclosed in Japanese Patent Publication No. 60-46465 (corresponding to EP-A-0066047, IBM), in order to transmit all the information required for performing specific edit processing to a CPU such as a wordprocessor, a user transmits (inputs) to the computer two types of information, i.e., information designating a place or an area within the text in which an instruction is to be executed, and information as to the type of instruction which is to be executed in a designated place. Typically, information specifying the area in which an instruction is to be executed is designated by moving a screen cursor on a display, by means of a moving device such as a key cursor on a keyboard or a pointing cursor of a mouse or a track ball. On the other hand, information as to the type of instruction which is to be executed is designated by selecting an edit instruction key and the like (e.g., a delete key and an insert key) on the keyboard or on an icon (tradename). Such a system is disclosed in Japanese Patent Disclosure (Kokai) No. 60-195632.

However, information specifying a place or an area in which an instruction is to be executed and information as to the type of an instruction must be input independently, resulting in a cumbersome operation for a user and a low input speed.

It is therefore, an object of the present invention to provide an instruction input system and a method for inputting instructions, which allow a fast and simple input operation.

In accordance with the invention this object is solved by the features of claims 1 and 9.

Other objects, features and advantages of the present invention will become apparent from the following description, taken in conjunction with the accompanying drawings, in which:

Figures 1A, 1B, and 1C are views respectively showing examples of symbol formats for instruction inputs;

Fig. 2 is a block diagram showing an overall arrangement of an electronic processor used in an embodiment of the present invention;

Fig. 3 is a block diagram showing an arrangement of detected coordinate memory 44 shown in Fig. 2;

Fig. 4 is a block diagram showing an arrangement of instruction analyzer 45 shown in Fig. 2;

Fig. 5 is a main flow chart for explaining the operation of an embodiment of the present invention;

Fig. 6 is a flow chart for explaining insert-instruction identification processing, as another operation of the present invention;

Fig. 7 is a flow chart for explaining delete-instruction identification processing, as yet another operation of the present invention;

Fig. 8 is a flow chart for explaining frame-instruction identification processing, as still another operation of the present invention;

Fig. 9 is a flow chart for explaining insert-instruction issue, as still another operation of the present invention;

Fig. 10 is a flow chart for explaining delete-instruction issue, as yet another operation of the present invention; and

Fig. 11 is a flow chart for explaining frame-instruction issue, as still another operation of the present invention.

An embodiment in which the present invention is applied to a wordprocessor will now be described.

In this embodiment, in order to effect an edit function of a wordprocessor, symbols such as proofreader's symbols of a text are input as edit instruction inputs. Marks are input from a transparent touch operation surface overlapping a text output screen.

Figs. 1A to 1C show input examples of proofreader's symbols, in which Fig. 1A shows an input example of an insert instruction symbol, Fig. 1B shows an input example of a delete instruction symbol, and Fig. 1C shows an input example of a frame instruction symbol. For example, when symbol "↖" as shown in Fig. 1A is input, it is decoded as "insert into text point (3,1)" by an instruction analyzer. In this case, the type of instruction "insert" and an area or a position "text point (3,1)" in which the instruction is to be executed can be simultaneously identified.

The input symbols "↖" shown in Fig. 1A schematically represents the locus of the finger tip moving on the touch-operation surface. In fact, this symbol is a combination of lines as thick as the finger tip. The same holds true of the symbols shown in Figs. 1B and 1C.

As is apparent from this example, according to the embodiment, a user need only input a symbol for an edit instruction, so that a user interface is significantly improved.

In addition, in this embodiment, assume that a processor, i.e., a wordprocessor in this case controls positions with respect to the respective characters of a text displayed on an output screen (e.g., the processor controls character positions by a reference position in the upper left of the output screen, i.e., relative X-coordinate and Y-coordinate from origin (0,0)) at each intersection or each matrix element (e.g., in the case of a text memory, a memory cell at a predetermined row number and a predetermined column number) on a mesh defined by a plurality of straight lines parallel to a row direction, i.e., the X direction and a plurality of straight lines parallel to the Y direction, i.e., a column direction (e.g., a mesh constituted by each

character row and each character column on a display memory, or a page of each character row and each character column on the text memory). Such a position control function is a standard function for conventional wordprocessors.

On the contrary, operation position signals with respect to the touch operation surface, which are input by an input processor having the touch operation surface, are finally detected or identified as intersections on a mesh (to be referred to as an input mesh hereinafter) defined by a plurality of straight lines parallel to the X direction and a plurality of straight lines parallel to the Y direction.

Therefore, an instruction analyzing means analyzes the type of instruction and an area in which the instruction is to be executed in accordance with a pattern formed by a set of intersections on the input mesh. Then, the instruction analyzing means identifies the instruction execution area, defined by intersections or intersection groups on the input mesh, with positions or position groups on the mesh provided on the text memory or the text display memory of the wordprocessor.

As described above, in this embodiment, since the input mesh similar to the mesh of the wordprocessor is used, mesh conversion or scale conversion need only be performed to obtain an area in which an instruction is to be executed by the wordprocessor. Therefore, a quantity of processing and a memory capacity including the area for programs required for the above operation are very small, thereby greatly improving a user interface or response.

Fig. 2 shows an overall arrangement of an electronic processor for explaining an embodiment of the present invention.

Reference numeral 41 denotes an input/output device for inputting proofreader's symbols and for displaying outputs of characters from wordprocessing circuit 46. For example, device 41 is arranged by a combination of a display such as a CRT or an LCD, a touch input device obtained by overlapping a transparent touch panel on a screen of the display, and a coordinate input device.

Reference numerals 42 and 43 denote circuits for detecting operation positions or input coordinates with respect to the touch operation surface. A Y-coordinate is detected by Y-coordinate detector 42, and an X-coordinate is detected by X-coordinate detector 43.

As a touch operation surface as a coordinate input panel and a circuit for detecting an input point, a panel of any conventional type such as a digital detecting type, an analog detecting type, a matrix resistance/conductor type, a voltage divider type, a strain gauge type, an optical type, a capacitive type, and a surface wave type, may be used. More specifically, a panel of any type may be used

if only input positions with respect to the touch operation surface (which need not be a physical panel) can be detected.

Coordinate data detected by detectors 42 and 43 are stored in detected coordinate memory 44.

Stored coordinate data or data of formation of a combination of points are analyzed by instruction analyzer 45 and the type of instruction and an area in which the instruction is to be executed are identified.

An analysis result is supplied to circuit 46 and an actual text edit processing is performed. Note that circuit 46 may be constituted by a conventional wordprocessor. An arrangement of the wordprocessor itself is not a primary concern of the present invention, and a description thereof will be omitted.

#### Detected Coordinate Memory, Instruction Analyzer (Figs. 3 and 4)

Fig. 3 shows an internal arrangement of detected coordinate memory 44, and Fig. 4 shows an internal arrangement of instruction analyzer 45.

X,Y-coordinate comparator 441 prevents inputs of the same coordinates. When coordinate data different from previous coordinate data is detected, comparator 441 supplies a coordinate detection signal to coordinate memory controller 443 to inform detection of an effective coordinate, supplies coordinate data through a data bus, and accesses coordinate memory RAM 442 to store coordinate data different from previously input data. On the other hand, controller 443 supplies the coordinate detection signal from comparator 441 to CPU 451 in instruction analyzer 45 (Fig. 4) consisting of CPU 451 and program memory 452 so as to designate a write address for RAM 442. Therefore, when data of a write address to be designated in RAM 442 is sent back from CPU 451, controller 443 outputs the address designation signal through the address buffer and designates the corresponding write address of RAM 442. Since a request for designating the write address is thus completed, the coordinate detection signal is disabled and is no longer supplied from controller 443 to CPU 451. This operation is repeated to sequentially input and store updated coordinate data different from previous data into RAM 442.

When the coordinate detection signal is not generated from comparator 441 for a predetermined period of time, RAM 442 is started to be accessed by outputs from analyzer 45 through controller 443. Coordinate data analyzed by analyzer 45 is output and stored in comparator 441 in a FIFO (first-in first-out) manner through controller 443.

When an analyzation operation is completed and an address data output from analyzer 45 to controller 443 is terminated, controller 443 supplies an output control signal to comparator 441, and coordinate data is transferred to RAM 442 until the stack becomes empty, i.e., the coordinate detection signal is disabled.

#### (Operation)

An operation of the embodiment of the electronic processor having the above arrangement used in the present invention and especially processing of analyzer 45 will be described below.

#### Overall Processing (Fig. 5)

Fig. 5 shows a main flow executed by analyzer 45.

##### Steps G1, G2:

Insert instruction identification steps for checking whether a set of input coordinate data is of insert instruction type.

##### Step G3 :

An output step executed in the case of an insert instruction, including a conversion step with respect to a text point.

##### Steps G4, G5:

Delete instruction identification steps executed unlike in the case of an insert instruction, for checking whether an input coordinate data is of a delete instruction type.

##### Step G6 :

A step executed in the case of a delete instruction, including a conversion step with respect to the text point.

##### Steps G7, G8:

Frame instruction identification step executed unlike in the case of a delete instruction, for checking whether the input coordinate data is of a frame instruction type.

##### Step G9 :

A step executed in the case of a frame instruction, including a conversion step with respect to the text point.

##### Step G10 :

A step executed when the set of input coordinate data is of no instruction type, for outputting an analysis disable signal.

#### Insert Instruction Identification Step (Fig. 6)

Fig. 6 shows a flow chart of an insert instruction identification step.

In this embodiment, as shown in Fig. 1A, an insert instruction is identified when symbols such as "\^" and "\^" are input.

Three points are required to constitute a symbol (step A1). An X-coordinate of a first text point is added to one to obtain the next X-coordinate (step A2), and the X-coordinate of a second point is offset to the right by one to obtain an X-coordinate of a third point (step A3). A Y-coordinate corresponding to the X-coordinate of the first point and a Y-coordinate corresponding to the X-coordinate of the third point are at the same height (step A4), and a difference between the Y-coordinate of the second point and the Y-coordinate of the first point is one (step A5). If all the conditions are satisfied, an insert symbol is identified (step A6).

Otherwise, the symbol is not identified as an insert symbol (step A7).

Note that if the conditions for identifying the insert mark are too strict, the conditions may be made less strict. Changes in program required for this can be easily performed by those skilled in the art.

#### Delete Instruction Identification Step (Fig. 7)

Fig. 7 shows a flow chart of a delete instruction identification step.

As shown in Fig. 1B, a delete instruction is identified when a symbol such as "→" is input assuming that one line is deleted in a lateral writing format in this embodiment. Therefore, the conditions are such that for all the coordinate inputs, X-coordinates are shifted in the right one by one and all Y-coordinates are set at the same height.

A total number of points is obtained, and an initial value of variable  $i$  of the number representing the order of points is set (step B1). Assuming that pairs of coordinates are sequentially formed from an end point to a start point and the number representing a point near the end point is variable  $i$ , the following processing is performed (steps B2 to B7).

When an  $i$ th X-coordinate is located in the right of an  $(i - 1)$ th X-coordinate at the left side thereof by one unit (one unit of the input mesh) (step B2), and when an  $i$ th Y-coordinate and an  $(i - 1)$ th Y-coordinate are located at the same horizontal level (step B3), an  $i$ th point is determined to be located on a line to be deleted, and the coordinate number returns to previous coordinate number  $x(i - 1)$  (step B4). The above steps are repeated until  $i = 1$  (step B5). A delete instruction is identified when all the points are located on the line to be deleted (step B6), and a delete instruction is not identified when even one point does not satisfy the above conditions (step B7).

Similar to the above example, a delete instruction may be identified by the less strict conditions. For example, if the start and end points satisfy the conditions of steps B2 and B3, the type of instruc-

tion is identified as a "delete instruction", or a reverse direction (right-to-left direction) is allowed.

#### Frame Instruction Identification Step (Fig. 8)

Fig. 8 shows a flow chart of a frame instruction identification step.

An example of a frame input is shown in Fig. 1C. In this example, a symbol input is obtained by the following array of (X,Y) points. That is, (1,1), (2,1), (3,1), (4,1), (5,1), (5,2), (5,3), (5,4), (5,5), (4,5), (3,5), (2,5), (1,5), (1,4), (1,3), (1,2), and (1,1). The set of points is characterized in that the start and end points are located at the same position and a line for a frame is drawn on a line obtained by connecting the points. In addition, in this example, assume that four squares of the input mesh (represented by numbers not encircled in Figs. 1A, 1B, and 1C) correspond to one square of the output mesh (represented by encircled numbers, e.g., (①, ①), (①, ②), (②, ①), and (②, ②)), and that an intersection of the output mesh (the center of a character in the square) coincides with only an intersection of a row point (Y-coordinate) and a column point (X-coordinate) both having even numbers of intersections on the input mesh. Therefore, it is checked that the start and end points coincide with each other in steps C1 and C2, and 16 points required for a frame as shown in Fig. 1C are set as  $i = 16$  in step C3. In step C4, 0 is inserted in another variable  $j$ . Then, in steps C5 and C6, a value of  $j$  is obtained by determining whether least significant bits of the X-coordinate and the Y-coordinate of the  $i$ th point are 0 or 1, and it is determined whether the obtained value of  $j$  is 0 in step C7. Assuming that the point at the center of the frame is input by a touch operation, X- and Y-coordinates of the point consist only of even numbers. If all the input points consist of only even numbers, it is determined not to satisfy  $i \neq 0$ , i.e., not to identify a frame instruction, and the input operation is stopped through step C10. On the other hand, if it is determined that  $j \neq 0$  in step C7, a calculation in which 1 is subtracted from the  $i$ th point is performed in step C8, thereby obtaining new variable  $i$ . If a value of  $i$  is  $i \neq 0$ , the flow returns to step C4, and the input points are checked again. If  $i$  is not  $i = 0$ , the frame instruction is determined in step C11.

#### Issue Instruction (Figs. 9, 10, and 11)

When the insert instruction is identified, the flow of Fig. 9 begins.

An insert instruction signal is supplied to circuit 46 to inform that the type of instruction is the "insert instruction" (step S1). An X-coordinate of the start point representing an insert position and a

Y-coordinate of a second point are converted into X- and Y-coordinates of a text point and then supplied to circuit 46 (steps S2 and S3).

As described above, in this embodiment, input coordinates are defined using an input mesh similar to an output mesh relating to the text point. Therefore, in an area in which an instruction is to be executed, conversion from input mesh to output mesh systems can be easily performed (step S2 of Fig. 9; see Fig. 1).

Circuit 46, which receives the insert instruction signal and information of a text point to which the insert operation is performed, rearranges the text memory by a normal system and rewrites memory contents in the display memory (refresh memory) so as to execute the insert instruction.

Fig. 10 shows a flow chart for issuing a delete instruction corresponding to step G6 of Fig. 5, and Fig. 11 shows a flow chart for issuing the frame instruction corresponding to step G9 of Fig. 5. Operations are apparent from descriptions in Figs. 10 and 11, respectively, and a detailed description thereof will be omitted.

#### Modifications

The present invention is not limited to the above embodiment, but may be variously modified or changed.

For example, in the above embodiment, a transparent input device having a touch operation surface on a display screen is used as an input means. During an interactive operation, operation inputs easily correspond to response results in this device, resulting in convenience.

However, a coordinate input device having a touch operation surface independently of a display screen, such as a tablet or a digitizer, may be used.

In addition, the present invention is applied to a wordprocessor in the above embodiment. However, the present invention may also be applied to an interactive graphic processor.

Furthermore, many kinds of symbol inputs with respect to the touch operation surface may be made in addition to those of the above embodiment, but not all of them can be described in this specification.

For example, in addition to a symbol of one stroke, a symbol of two or more strokes may be used.

More specifically, "○" (this symbol means that a space or spacing between characters or lines is removed), "←" (this symbol means that a character array or a symbol array is moved in a direction indicated by an arrow, i.e., to the left), and the like may be made.

Instruction analyzer 45 can be easily so modified as to correspond to symbol inputs of a plurality of strokes.

In addition, in the above embodiment, an input mesh relating to operation positions of the touch operation surface is similar to an execution mesh (text mesh in the above embodiment) relating to execution of instructions. This means that an area in which an instruction is to be executed can be immediately obtained without complex processing in accordance with input coordinates or coordinate groups. Therefore, response can be greatly improved, and the quantity of processing and a memory capacity can be greatly saved.

Scale conversion from the input mesh to the execution mesh need not be limited to one scheme. For example, when the present invention is applied to a wordprocessor capable of enlarging a multi-sized character font on a display screen, by performing communication between a wordprocessing circuit and an instruction analyzer to switch a scale conversion coefficient, scale conversion can be performed onto the execution mesh at the current time, and the current position, i.e., instruction execution area information can be defined on the execution mesh. This applies to a processor with a multi-window function having independent execution meshes in units of windows. Thus, an idea of a mesh can provide a great effect in association with symbol inputs. Since conversion is performed between meshes, geometric conversion (scale conversion, parallel conversion (offset), rotational conversion, and shearing) need only be performed to convert coordinates from the input mesh onto the execution mesh. In the simple case, only the scale conversion is required.

However, if a processor for executing instructions has no such idea of the execution mesh, processing of identification of an object to be processed (e.g., a graphic pattern to be deleted) and determination of a processing area (determination of a position of the graphic pattern to be deleted) is utilized in accordance with a position of a screen cursor, as in a certain interactive graphic processor. That is, a coordinate input position (a position on the input mesh) is replaced with a screen cursor position (a position of an image) in this processing, thereby executing the instruction.

Assume that a display is provided with an image memory (e.g., a refresh memory) for storing display data on a screen, a position memory for controlling and storing positions of graphic symbols (originally graphic patterns obtained from codes) included in display data on the screen, and a code memory for storing a code at a position represented by the position memory (i.e., a code of a graphic symbol displayed at this position). In addition, assume that an instruction analyzer analyzes a

set of input points and obtains "replace" as the type of instruction. In this case, assume that the instruction analyzer admits the set of input points or a subset thereof, e.g., two points as a replacement range of the graphic symbols. Note that in this stage, the replacement range is defined on only the input mesh. Therefore, first, the replacement range is redefined on the screen, on the image, and on the position memory. Then the position memory is accessed to check whether a graphic symbol is present in this range. If the graphic symbol is present, a code of the symbol is found by referring to the code memory. Thus, all the positions of the graphic symbols to be replaced (two in this case) are found on the screen. In addition, all the codes of the graphic symbols to be replaced (two in this case) are found on the screen. Therefore, using these position information and code information, the image memory (refresh memory) is rewritten. That is, erase codes are used instead of the two found codes to erase two symbols from the image memory (refresh memory) in accordance with two pieces of found position information. Then, the two found codes are supplied to a memory such as a code/graphic pattern conversion memory or a conversion program and are converted into graphic symbols. A first converted symbol is written in the image memory using the position where a second symbol was located (this position has been found in previous processing). The second converted symbol is written in the image memory using a position where the first symbol was located (this position has been found in previous processing). Thus, two graphic symbols on the screen are replaced with each other.

### Claims

1. An instruction input system (41, 42, 43, 44, 45, 46) for an electronic graphic or word processor controlling an image on a display means comprising input means (41, 42, 43) and instruction-analyzing means (44, 45) for determining a type of an instruction input **characterized in that** the input means (41, 42, 43) has a touch-sensitive input panel whereby an instruction is input in the form of a pattern produced by a touch operation on said panel, said instruction being determined by said instruction analyzing means by analyzing the form of the pattern input via said touch-sensitive panel, said determined instruction being executed by said processor within an area of said display means which is determined by the location of said pattern on said panel.
2. An instruction input system according to claim 1 **characterized in that** said touch-sensitive input panel is arranged on a display screen of said display means.
3. An instruction input system according to claim 1 or 2 **characterized in that** said pattern is generated by a finger activating said touch-sensitive input panel.
4. An instruction input system according to at least one of the preceding claims, **characterized in that** said pattern is a symbol of proof-readers' symbols or symbols similar thereto.
5. An instruction input system according to claim 4 **characterized in that** said symbol is a stroke.
6. A system according to at least one of the preceding claims, **characterized in that** said input means (41, 42, 43) comprises position detecting means (42, 43) for receiving an operation position signal on said touch-sensitive input panel, said position-detecting means (42, 43) being provided with means for producing the input operation position in the form of intersection coordinates on a first mesh, defined by a plurality of straight lines parallel to an X direction, and a plurality of straight lines parallel to a Y direction.
7. A system according to at least one of the preceding claims **characterized in that** said processor (46) controls positions of graphic symbols or text characters which are displayed, on said display means (41), in intersections or squares on a second mesh defined by a plurality of straight lines parallel to the X or row direction, and a plurality of straight lines parallel to the Y or column direction.
8. A system according to claim 6 or 7 **characterized in that** said instruction-analyzing means (44, 45) comprises means for converting the area in which the instruction is to be executed from an area on the first mesh, defined by said position-detecting means (42, 43) to an area on the second mesh, defined by said processor (46).
9. A method for inputting instructions to an electronic graphic or word processor for manipulating an image on a display screen, controlled by said processor, **characterized in that** a desired instruction is inputted via a touch-sensitive input panel associated with said processor by selecting an area on said panel cor-



set of input points and obtains "replace" as the type of instruction. In this case, assume that the instruction analyzer admits the set of input points or a subset thereof, e.g., two points as a replacement range of the graphic symbols. Note that in this stage, the replacement range is defined on only the input mesh. Therefore, first, the replacement range is redefined on the screen, on the image, and on the position memory. Then the position memory is accessed to check whether a graphic symbol is present in this range. If the graphic symbol is present, a code of the symbol is found by referring to the code memory. Thus, all the positions of the graphic symbols to be replaced (two in this case) are found on the screen. In addition, all the codes of the graphic symbols to be replaced (two in this case) are found on the screen. Therefore, using these position information and code information, the image memory (refresh memory) is rewritten. That is, erase codes are used instead of the two found codes to erase two symbols from the image memory (refresh memory) in accordance with two pieces of found position information. Then, the two found codes are supplied to a memory such as a code/graphic pattern conversion memory or a conversion program and are converted into graphic symbols. A first converted symbol is written in the image memory using the position where a second symbol was located (this position has been found in previous processing). The second converted symbol is written in the image memory using a position where the first symbol was located (this position has been found in previous processing). Thus, two graphic symbols on the screen are replaced with each other.

### Claims

1. An instruction input system (41, 42, 43, 44, 45, 46) for an electronic graphic or word processor controlling an image on a display means comprising input means (41, 42, 43) and instruction-analyzing means (44, 45) for determining a type of an instruction input **characterized in that** the input means (41, 42, 43) has a touch-sensitive input panel whereby an instruction is input in the form of a pattern produced by a touch operation on said panel, said instruction being determined by said instruction analyzing means by analyzing the form of the pattern input via said touch-sensitive panel, said determined instruction being executed by said processor within an area of said display means which is determined by the location of said pattern on said panel.
2. An instruction input system according to claim 1 **characterized in that** said touch-sensitive input panel is arranged on a display screen of said display means.
3. An instruction input system according to claim 1 or 2 **characterized in that** said pattern is generated by a finger activating said touch-sensitive input panel.
4. An instruction input system according to at least one of the preceding claims, **characterized in that** said pattern is a symbol of proof-readers' symbols or symbols similar thereto.
5. An instruction input system according to claim 4 **characterized in that** said symbol is a stroke.
6. A system according to at least one of the preceding claims, **characterized in that** said input means (41, 42, 43) comprises position detecting means (42, 43) for receiving an operation position signal on said touch-sensitive input panel, said position-detecting means (42, 43) being provided with means for producing the input operation position in the form of intersection coordinates on a first mesh, defined by a plurality of straight lines parallel to an X direction, and a plurality of straight lines parallel to a Y direction.
7. A system according to at least one of the preceding claims **characterized in that** said processor (46) controls positions of graphic symbols or text characters which are displayed, on said display means (41), in intersections or squares on a second mesh defined by a plurality of straight lines parallel to the X or row direction, and a plurality of straight lines parallel to the Y or column direction.
8. A system according to claim 6 or 7 **characterized in that** said instruction-analysing means (44, 45) comprises means for converting the area in which the instruction is to be executed from an area on the first mesh, defined by said position-detecting means (42, 43) to an area on the second mesh, defined by said processor (46).
9. A method for inputting instructions to an electronic graphic or word processor for manipulating an image on a display screen, controlled by said processor, **characterized in that** a desired instruction is inputted via a touch-sensitive input panel associated with said processor by selecting an area on said panel cor-



responding to a desired area on said display screen in which the desired instruction has to be executed and by generating a pattern symbolising the desired instruction by touch operating said touch-sensitive input panel at said selected area, said pattern is analysed by said processor and interpreted as the desired instruction, said desired instruction is executed by said processor in the said desired area of the display screen.

10. A method according to claim 9 characterized in that said pattern is generated by a finger activating said touch-sensitive input panel.
11. A method according to claim 9 or 10 characterized in that said generated pattern is a symbol of proofreaders' symbols.

#### Patentansprüche

1. Befehlseingabesystem (41, 42, 43, 44, 45, 46) für einen elektronischen Grafik- oder Wortprozessor, der ein Bild auf einer Anzeigeeinrichtung steuert mit einer Eingabeeinrichtung (41, 42, 43) und einer Befehlsanalysiereinrichtung (44, 45) zum Bestimmen der Art eines eingegebenen Befehls, **dadurch gekennzeichnet**, daß die Eingangeinrichtung (41, 42, 43) eine berührungsempfindliche Eingabefläche aufweist, wobei ein Befehl in Form eines Musters, welches durch eine Berührungsaktivität auf der Fläche erzeugt worden ist, eingegeben wird, der Befehl durch die Befehlsanalysiereinrichtung durch Analysieren der Form des über die berührungsempfindliche Fläche eingegebenen Musters bestimmt wird, der bestimmte Befehl von dem Prozessor innerhalb eines Bereichs der Anzeigeeinrichtung ausgeführt wird, welcher über die Lage des Musters auf der Fläche bestimmt wird.
2. Befehlseingabesystem nach Anspruch 1, **dadurch gekennzeichnet**, daß die berührungsempfindliche Eingabefläche auf einem Anzeigeschirm der Anzeigeeinrichtung angeordnet ist.
3. Befehlseingabesystem nach Anspruch 1 oder 2, **dadurch gekennzeichnet**, daß das Muster mittels einer Fingertätigkeit auf der berührungsempfindlichen Eingabefläche erzeugt wird.
4. Befehlseingabesystem nach mindestens einem der vorhergehenden Ansprüche, **dadurch gekennzeichnet**, daß das Muster ein Symbol aus Korrektursymbolen oder dazu ähnlichen Symbolen darstellt.
5. Befehlseingabesystem nach Anspruch 4, **dadurch gekennzeichnet**, daß das Symbol ein Strich ist.
6. System nach mindestens einem der vorhergehenden Ansprüche, **dadurch gekennzeichnet**, daß die Eingabeeinrichtung (41, 42, 43) eine Positionsdetektiereinrichtung (42, 43) aufweist zum Empfangen eines Betätigungspositionssignals auf der berührungsempfindlichen Eingabefläche, wobei die Positionsdetektiereinrichtung (42, 43) mit einer Einrichtung zum Erzeugen der Eingangsoperationsposition in Form von Zwischenkoordinaten eines ersten Netzes, welches durch eine Vielzahl von geraden Linien parallel zur X-Richtung und eine Vielzahl von geraden Linien parallel zur Y-Richtung definiert wird, ausgestattet ist.
7. System nach mindestens einem der vorhergehenden Ansprüche, **dadurch gekennzeichnet**, daß der Prozessor (46) die Positionen von grafischen Symbolen oder Textzeichen, welche auf der Anzeigeeinrichtung (41) angezeigt werden, in Zwischenabschnitten oder Quadraten auf einem zweiten Netz, welches durch eine Vielzahl von geraden Linien parallel zur Xoder Zeilen-Richtung, und einer Vielzahl von geraden Linien parallel zur Y- oder Spaltenrichtung definiert wird, steuert.
8. System nach Anspruch 6 oder 7, **dadurch gekennzeichnet**, daß die Befehlsanalysiereinrichtung (44, 45) eine Einrichtung zum Umwandeln des Bereichs, in welchem der Befehl ausgeführt werden soll, von einem Bereich auf dem ersten Netz, welcher durch die Positionier-Detektionseinrichtung (42, 43) definiert ist, in einen Bereich auf dem zweiten Netz, welches durch den Prozessor (46) definiert wird, aufweist.
9. Verfahren zum Eingeben von Befehlen an einen elektronischen Grafik- oder Wortprozessor zum Beeinflussen eines Bildes auf einem Anzeigeschirm, welches von dem Prozessor gesteuert wird, **dadurch gekennzeichnet**, daß ein gewünschter Befehl über eine berührungsempfindliche Eingabefläche, die mit dem Prozessor assoziiert ist, **dadurch** eingegeben wird, indem ein Bereich auf der Fläche ausgewählt wird, welcher mit einer gewünschten Fläche auf dem Anzeigeschirm, in dem der gewünschte Befehl ausgeführt werden soll, korrespondiert und indem ein Muster, welches den



gewünschten Befehl symbolisiert mittels einer Berührungstätigkeit auf der berührungsempfindlichen Eingabefläche an dem ausgewählten Bereich, erzeugt wird,  
das Muster durch den Prozessor analysiert wird und als der gewünschte Befehl interpretiert wird,  
der gewünschte Befehl durch den Prozessor in dem gewünschten Bereich des Anzeigeschirms ausgeführt wird.

10. Verfahren nach Anspruch 9, **dadurch gekennzeichnet**, daß das Muster durch eine Fingertätigkeit auf der berührungsempfindlichen Eingabefläche erzeugt wird.
11. Verfahren nach Anspruch 9 oder 10, **dadurch gekennzeichnet**, daß das erzeugte Muster ein Korrektursymbol ist.

#### Revendications

1. Système d'entrée d'instructions (41, 42, 43, 44, 45, 46) pour un traitement de graphique ou un traitement de texte électronique qui commande une image sur un moyen d'affichage comprenant un moyen d'entrée (41, 42, 43) et un moyen d'analyse d'instruction (44, 45) pour déterminer un type d'une entrée d'instruction, caractérisé en ce que le moyen d'entrée (41, 42, 43) comporte un panneau d'entrée tactile d'où de sorte qu'une instruction soit entrée sous la forme d'un motif produit par une opération de toucher sur ledit panneau,  
ladite instruction étant déterminée par ledit moyen d'analyse d'instruction en analysant la forme du motif entré via ledit panneau tactile,  
ladite instruction déterminée étant exécutée par ledit dispositif de traitement dans une zone dudit moyen d'affichage qui est déterminée par l'emplacement dudit motif sur ledit panneau.
2. Système d'entrée d'instructions selon la revendication 1, caractérisé en ce que ledit panneau d'entrée tactile est agencé sur un écran d'affichage dudit moyen d'affichage.
3. Système d'entrée d'instructions selon la revendication 1 ou 2, caractérisé en ce que ledit motif est généré par un doigt qui active ledit panneau d'entrée tactile.
4. Système d'entrée d'instructions selon au moins l'une des revendications précédentes, caractérisé en ce que ledit motif est un symbole de codes typographiques ou des symboles qui leur sont similaires.
5. Système d'entrée d'instructions selon la revendication 4, caractérisé en ce que ledit symbole est un caractère de clavier.
6. Système d'entrée d'instructions selon au moins l'une des revendications précédentes, caractérisé en ce que ledit moyen d'entrée (41, 42, 43) comprend un moyen de détection de position (42, 43) pour recevoir un signal de position d'opération sur ledit panneau d'entrée tactile, ledit moyen de détection de position (42, 43) étant muni d'un moyen pour produire la position d'opération d'entrée sous la forme de coordonnées d'intersection sur un premier maillage défini par une pluralité de lignes droites parallèles à une direction X et par une pluralité de lignes droites parallèles à une direction Y.
7. Système selon au moins l'une des revendications précédentes, caractérisé en ce que ledit dispositif de traitement (46) commande des positions de symboles graphiques ou de caractères de texte qui sont affichées, sur ledit moyen d'affichage (41), dans des intersections ou des carrés d'un second maillage défini par une pluralité de lignes droites parallèles à la direction X ou direction de rangée et par une pluralité de lignes droites parallèles à la direction Y ou direction de colonne.
8. Système selon la revendication 6 ou 7, caractérisé en ce que ledit moyen d'analyse d'instruction (44, 45) comprend un moyen pour convertir la zone dans laquelle l'instruction doit être exécutée d'une zone du premier maillage définie par ledit moyen de détection de position (42, 43) en une zone du second maillage définie par ledit dispositif de traitement (46).
9. Procédé d'entrée d'instructions sur un traitement de graphique ou un traitement de texte électronique pour manipuler une image sur un écran d'affichage commandé par ledit dispositif de traitement, caractérisé en ce qu'une instruction souhaitée est entrée via un panneau d'entrée tactile associé audit dispositif de traitement en sélectionnant une zone sur ledit panneau correspondant à une zone souhaitée sur ledit écran d'affichage dans laquelle l'instruction souhaitée doit être exécutée et en générant un motif qui symbolise l'instruction souhaitée en actionnant de façon tactile ledit panneau d'entrée tactile au niveau de ladite zone sélectionnée,  
ledit motif est analysé par ledit dispositif de traitement et est interprété en tant qu'instruction souhaitée, et



ladite instruction souhaitée est exécutée  
par ledit dispositif de traitement dans ladite  
zone souhaitée de l'écran d'affichage.

10. Procédé selon la revendication 9, caractérisé 5  
en ce que ledit motif est généré par un doigt  
qui active ledit panneau d'entrée tactile.
11. Procédé selon la revendication 9 ou 10, carac- 10  
térisé en ce que ledit motif généré est un  
symbole de codes typographiques.

15

20

25

30

35

40

45

50

55

10