

[54] METHOD OF ENCODING GRAPHICAL DATA

[72] Inventor: George Y. Kurata, San Diego, Calif.
[73] Assignee: Typagraph Corporation, San Diego, Calif.
[22] Filed: Mar. 18, 1970
[21] Appl. No.: 20,494

[52] U.S. Cl. 444/1, 340/172.5
[51] Int. Cl. G06f 3/14
[58] Field of Search.....340/172.5; 346/29

[56] References Cited
UNITED STATES PATENTS

3,434,113	3/1969	Wiley et al.	340/172.5
3,199,111	8/1965	Jennings et al.	346/29
3,377,622	4/1968	Burch, Jr. et al.	340/172.5
3,430,207	2/1969	Davis.....	340/172.5
3,438,003	4/1969	Bryan	340/172.5

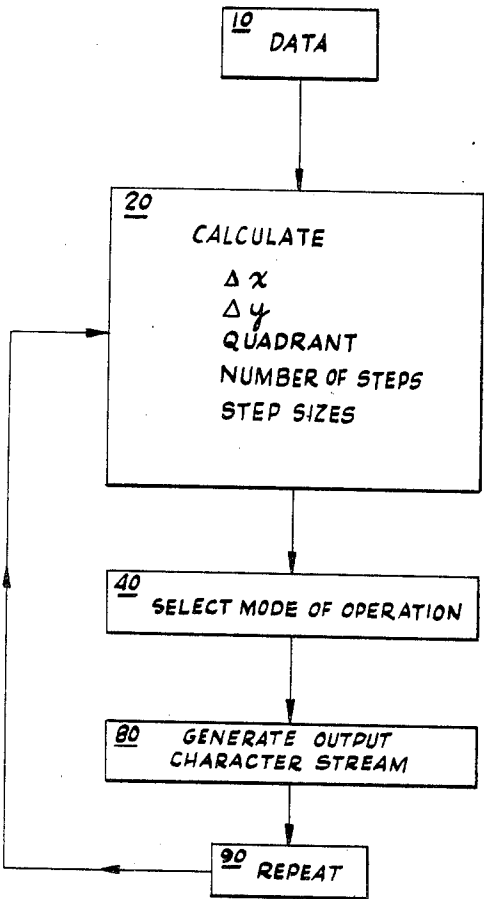
3,496,547 2/1970 Gorrell et al. 340/172.5

Primary Examiner—Paul J. Henon
Assistant Examiner—Sydney R. Chirlin
Attorney—Fulwider, Patton, Rieber, Lee & Utecht

[57] ABSTRACT

A method of encoding graphical data to be displayed by an output terminal where the output terminal has plural modes of operation and procedures are undertaken to select, using past history and future anticipated operation, the fastest and most efficient mode of operation. In the faster modes, part of the information defining the location to be reached and the symbol to be displayed are stored and remembered. In these modes less information need be transmitted to the terminal to execute a move and successive moves are thereby executed faster. The optimum mode may be selected by examining previous location information to predict a possible mode change.

34 Claims, 6 Drawing Figures



	MODE/QUAD	dx	dy	SYMBOL
MODE 1	STORED	STORED	STORED	VARIABLE
MODE 2	STORED	STORED	VARIABLE	STORED
MODE 3	STORED	VARIABLE	STORED	STORED
MODE 4	STORED	VARIABLE	VARIABLE	STORED

Fig. 1

CHARACTER STREAM

MODE 1	1 FLAG	2 MODE/ QUAD	3 dx	4 dy	5 SYMBOL	6 SYMBOL	7 SYMBOL	8 SYMBOL	9 SYMBOL
MODE 2	1 FLAG	2 MODE/ QUAD	3 dx	4 dy	5 SYMBOL	6 dy	7 dy	8 dy	9 dy
MODE 3	1 FLAG	2 MODE/ QUAD	3 dx	4 dy	5 SYMBOL	6 dx	7 dx	8 dx	9 dx
MODE 4	1 FLAG	2 MODE/ QUAD	3 dx	4 dy	5 SYMBOL	6 dx	7 dy	8 dx	9 dy

Fig. 2

INVENTOR.
GEORGE Y. KURATA
 BY *Zulwider, Patton, Rieber*
Lee, and Utecht
 ATTORNEYS

Fig. 3

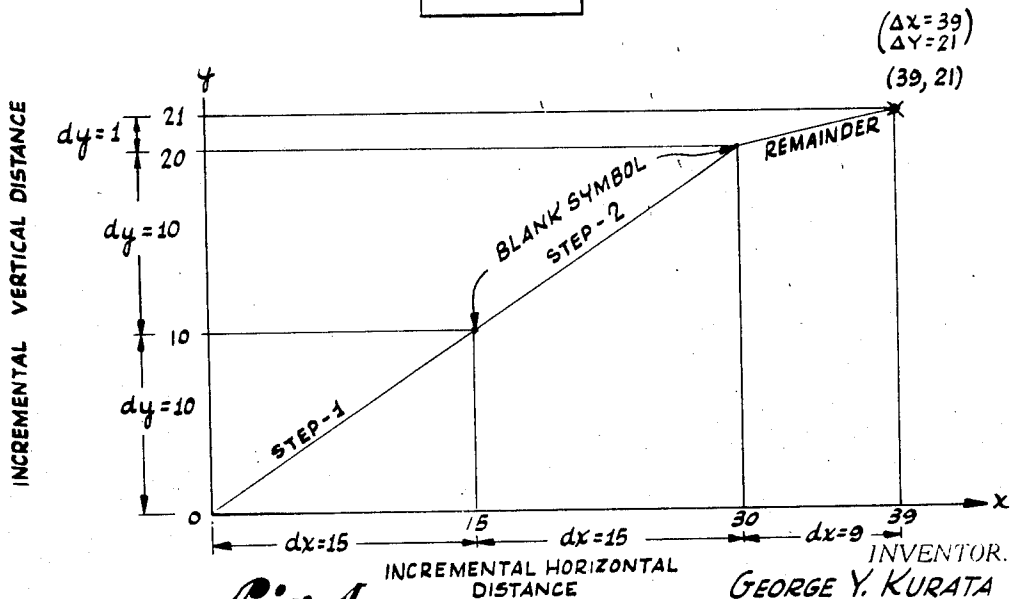
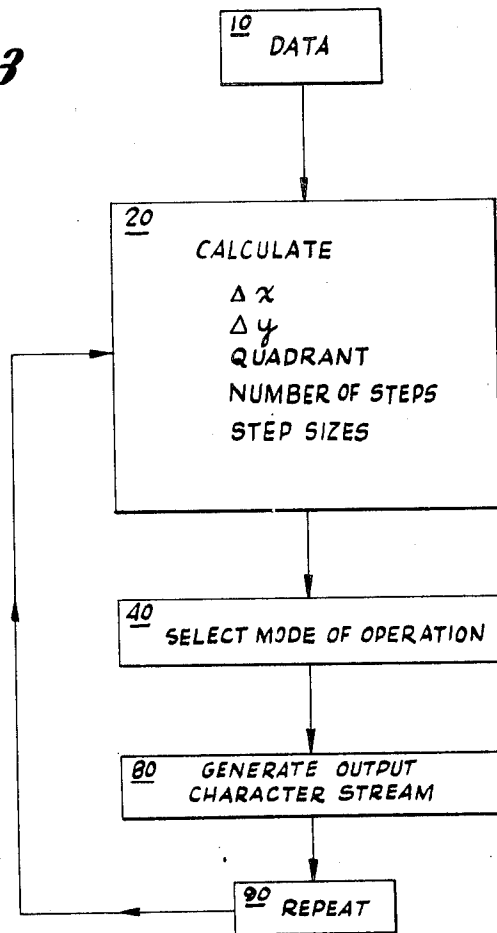
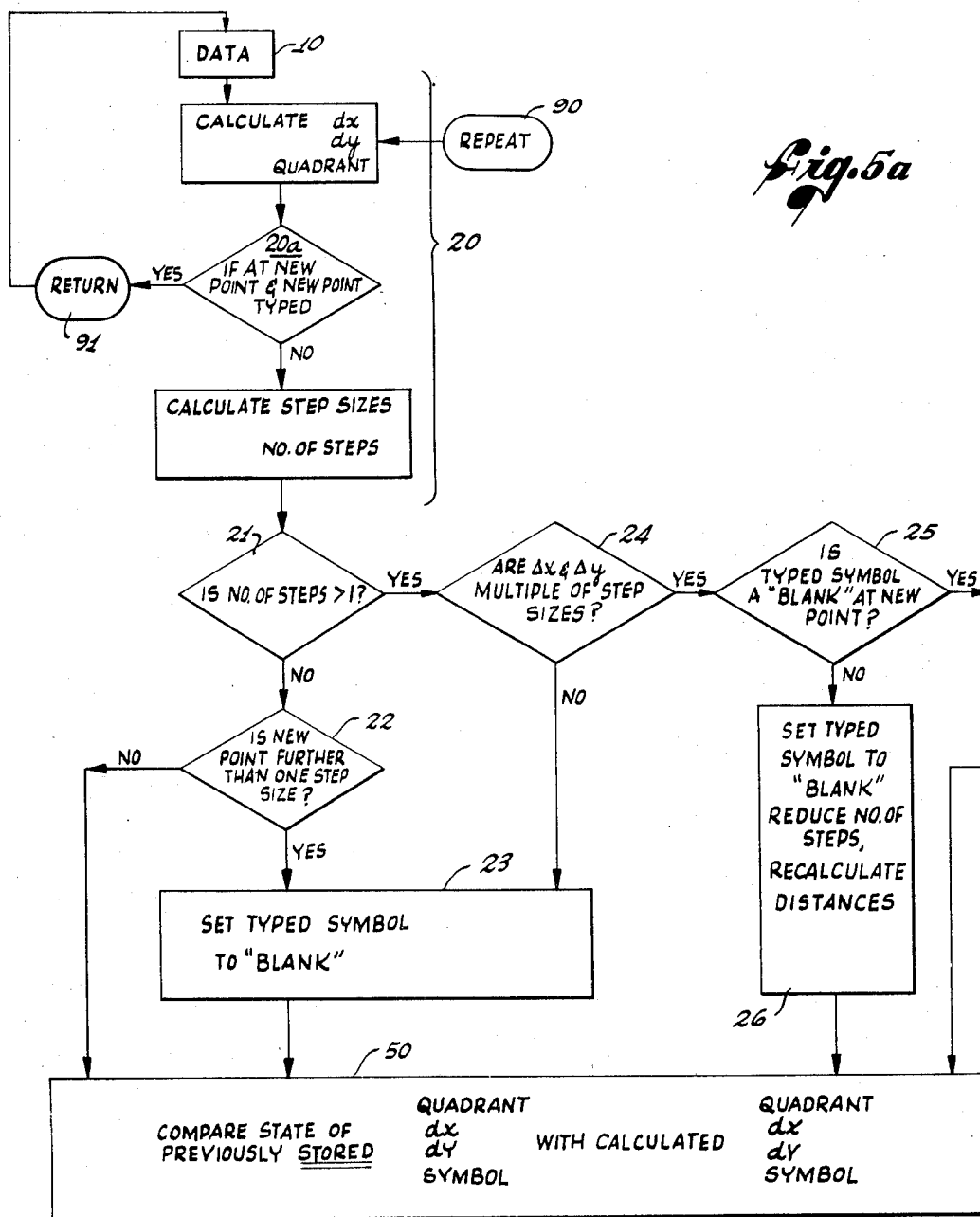


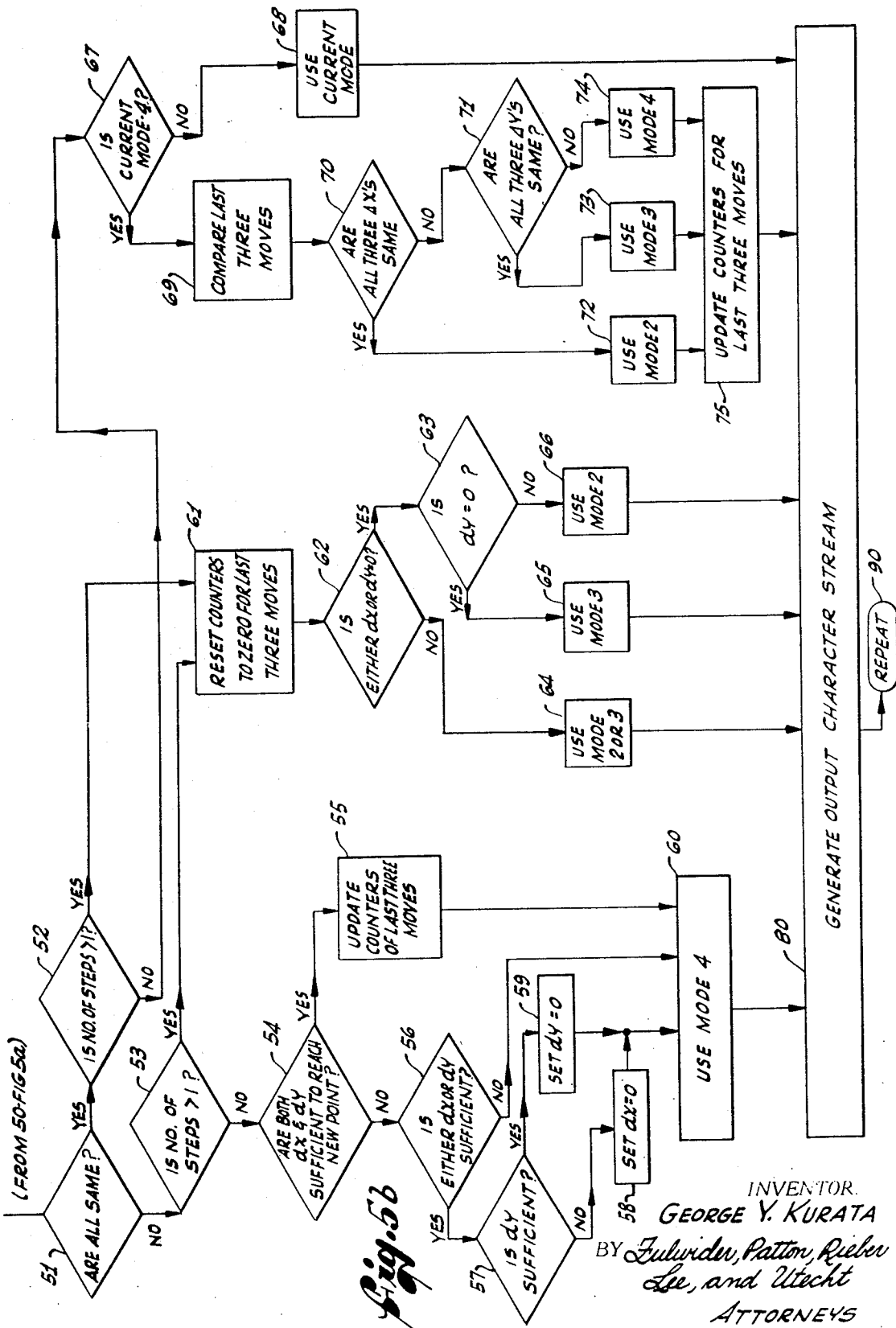
Fig. 4

INVENTOR.
 GEORGE Y. KURATA
 BY *Fulwider, Patton, Reber*
Lee, and Utecht
 ATTORNEYS



(TO 51-FIG 5B)

INVENTOR.
 GEORGE Y. KURATA
 BY *Zulwider, Patton, Riebs*
Lee, and Utecht
 ATTORNEYS



METHOD OF ENCODING GRAPHICAL DATA

BACKGROUND OF THE INVENTION

The present invention relates generally to a method of encoding graphical data and, more particularly, to a method by which a terminal output apparatus, controlled by a digital computer, can efficiently display such data as printed text or as graphical plots or both.

There presently exist many well-known terminal output devices capable of presenting encoded, computer-generated data in a form intelligible to a human operator. Thus, for example, a computer-controlled typewriter is often used to present the computer-generated data in easily readable, typewritten text. Similarly, a computer-controlled digital plotter is often used to present computer-generated data in graphical or pictorial form. Unfortunately, however, neither terminal output device above can additionally adequately perform the function of the other. Thus, because of its preset character spacing and restricted directional movement, the typewriter cannot function well as a graphical plotter. In a similar manner, the digital plotter can, at best, only draw or plot characters and symbols, and thus is unable to function additionally, in a practical sense, to type textual material.

There has recently been developed, however, an output terminal device which can both type text and plot graphically or intermix the two. For purposes of the present application this terminal output device having a combined typing and plotting capability will be referred to as a combined terminal.

Briefly, the combined terminal comprises a typewriter mechanism having independent, bidirectional control of the typewriter platen and of the type carrier head. Thus, the vertical or y-axis position of a character to be typed may be controlled by movement of the platen while horizontal or x-axis position may be controlled by movement of the type carrier head.

In its preferred form, the combined terminal has several significant features. First, it is an incremental device. That is, the terminal moves from its present x,y coordinate to the next, from point to point without referencing to an original origin for each new point. Other devices of the prior art typically move a discrete distance from an original origin for each successive new point.

Second, a point in the terminal coordinate system is defined with respect to the preceding point by three characteristics in addition to the desired symbol to be placed at the point—namely x value (dx), y value (dy), and quadrant. Thus, three characters define the location of any point in the coordinate system with respect to the preceding point. Many conventional plotters, on the other hand, must receive four characters to identify the next desired point—namely, x value, y value, whether x is positive or negative, and whether y is positive or negative. By eliminating the need for one transmitted character, the terminal thus operates faster than plotters using four characters.

Third, the combined terminal can operate in several modes. To decide which mode is used for a given character stream, the terminal can look ahead and select the optimum mode. In certain cases it can look back and, based on previous moves, predict the optimum mode to switch to.

SUMMARY OF THE PRESENT INVENTION

Briefly, and in general terms, the present invention involves a method of information handling for operating a computer-commanded, combined typing and plotting terminal, in a fast and efficient manner.

The present invention further relates to a method for encoding computer-generated data to be transmitted to and outputted by a combined typing and plotting terminal.

In addition, the present invention relates to a method for encoding computer-generated data so that a maximum quantity of information is contained in a minimum number of characters, i.e., to compress the data stream and increase information density.

Also, the present invention relates to a method of selecting from several operating modes for an output terminal, the mode that is optimum as to speed and efficiency.

In addition, the present invention relates to a method for moving in an x-y coordinate system expeditiously from point to point.

According to the present invention the movement from a present point to a desired new point involves calculating the quadrant of the new point with respect to the present point, calculating the horizontal incremental distance (Δx) along the x-axis to the new point and calculating the vertical incremental distance (Δy) along the y-axis to the new point, as well as ascertaining the desired symbol to be typed. Next, the number of steps in one direction (preferably the longest of Δx or Δy) is calculated where a step may comprise as many increments (dx or dy) as the physical constraints of the output terminal permits, for example 15. Subsequently, the number of steps in the remaining direction is calculated. If more than one step (i.e., more than 15 increments) is required in any given direction, then the symbol typed at the end of the first step is set to a "blank" so that only a blank space is typed. Successive steps are taken in plural moves until arriving at a preliminary location within one step in both x and y directions, of the new point. In many situations, a change of mode is then made from the plotting mode used to reach the preliminary location to a mode for moving the final distance to the new point and typing or otherwise displaying the symbol.

The output terminal can operate in any one of four different modes. Each mode has certain information (from among quadrant, dx , dy , and symbol characters) in storage which is remembered. Thus, only the variable characters need be transmitted to the terminal with a resultant shorter character stream and increased speed of operation of the terminal.

Certain modes are faster and more efficient than others, and certain procedures are taken, based on past history and anticipated future operation, to operate in the optimum mode.

The methods contemplated are well adapted for machine implementation by a digital computer or the like. The methods may be performed by special purpose computer hardware or by a general purpose computer which has been programmed by appropriate software.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a table showing which of the MODE/QUAD, dx , dy , and SYMBOL characters are stored and which are variable for each of MODES 1 through 4 in the method of the present invention;

FIG. 2 shows representative character streams for each of MODES 1 through 4;

FIG. 3 is a block diagram of the basic method for encoding data, in accordance with the present invention;

FIG. 4 is a graphical illustration showing the plotting of one representative point;

FIG. 5a is a detailed block diagram of a first part of the method of FIG. 3; and

FIG. 5b is a detailed block diagram of the remainder of the method of FIG. 3.

DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring now to the drawings, and more particularly to FIG. 1 thereof, there is shown a tabulation of the modes in which a combined typing and plotting terminal output device may operate. Four characters are needed to operate in each mode. The first character, "MODE/QUAD," defines both the mode of operation and the quadrant of the next point (one of four quadrants of a conventional Cartesian coordinate system). The second character dx defines the incremental distance along the x-axis for a given move, while the third character dy defines the incremental distance along the y-axis for the same move. The fourth character "SYMBOL" defines the symbol to be typed or otherwise displayed at the end of the same move at the desired point.

In each mode certain characters are stored as a given value. With the stored characters thus remembered only the variable characters need be transmitted to the "Typagraph." In MODE 1, the MODE/QUAD, dx and dy characters are stored and the SYMBOL is variable. In MODE 2, the MODE/QUAD, dx , and SYMBOL characters are stored, while dy is variable. In MODE 3, the MODE/QUAD, dy and SYMBOL characters are stored while dx is variable. Finally, in MODE 4, the MODE/QUAD and SYMBOL characters are stored while dx and dy remain variable.

FIG. 2 shows representative character streams as they would be transmitted to the "Typagraph" for each of MODES 1 through 4. Each character stream is initiated by a FLAG character. The FLAG character is a normally little-used character which is given a special assignment of commanding a plotting sequence. Thus, in MODE 1, characters 2 through 4 represent respectively the stored MODE/QUAD, dx and dy while character 5 represents the desired SYMBOL. Thereafter, characters 6 through 9 each represent any desired SYMBOLS. The character stream for MODE 1 thus commands typing for five different symbols each positioned in a manner defined by the stored MODE/QUAD, dx and dy characters from each preceding symbol. In a similar manner, note that in MODE 2, the variable character dy is shown as character 4 and thereafter as characters 6 through 9. In MODE 3, variable character dx is shown as character 3 and as characters 6 through 9. Finally, in MODE 4 both dx and dy characters are variable, dx being shown as characters 3, 6 and 8, while dy is shown as characters 4, 7 and 9.

It is evident from the above that the nine character stream of MODE 4 contains less total point information than the nine character streams of each of MODES 1 through 3. Thus, the stream of MODE 4 represents three consecutive points, the first defined by characters 3 and 4, the second defined by characters 6 and 7, and the third defined by characters 8 and 9. By comparison, the character stream of MODE 3, for example, represents five consecutive points shown by the five following pairs of characters: 3-4, 6-4, 7-4, 8-4 and 9-4. Since any stream of nine characters is transmitted to the "Typagraph" at the same speed, clearly the "Typagraph" will perform more moves in a given time interval in MODE 3 than in MODE 4. The method of encoding data, in accordance with the present invention, seeks to place a maximum amount of information in a given character stream so that a terminal device will operate in a fast and efficient manner.

FIG. 3 shows a basic method for encoding data in accordance with the invention. The data so encoded may then be transmitted to an output terminal for appropriate display. One suitable output terminal is the "Typagraph" terminal previously described.

The data 10 can have been generated by a computer as a solution to a given problem and can be, for example, a listing of information to be represented graphically such as points of a curve. The points may each be specified in terms of horizontal (x) and vertical (y) distance from a common origin.

The data 10 is first operated on and quantized. A maximum and minimum value for the variables of each axis (x and y) is ascertained. The axes are laid down and labeled and grid marks are inserted. The starting point for the type carrier head is selected. Information handling and data processing methods to perform these operations are well known in the art and are not treated further herein.

The encoding method of FIG. 3 is a reiterative process for encoding data to be transmitted to the "Typagraph" for display in graphical form. Basically, in moving from a given point on the graph to a desired new point at which a symbol is to be typed, the horizontal and vertical distances (Δx and Δy) and the quadrant of the desired new point with respect to the given point are calculated during calculation operation 20. Further, the number of steps required to reach the desired new point and the size of the steps is determined. Thereafter, in the selection operation 40 a proper mode is selected to efficiently move the first step toward the desired new point. An ap-

propriate character stream is generated by operation 80 to command the first move. In this regard, the characters are preferably assembled in a desired large quantity prior to being transmitted to the terminal device. At the point thus reached intermediate the given point and the desired new point, a "blank" symbol set in the above character stream is typed so that a blank space appears on an appropriate recording medium. The above method is then repeated in a repeat operation 90 the necessary number of times to reach a last intermediate point located within one step in each direction from the desired new point. The distance between this last intermediate point and the desired new point is termed a remainder. The method is then repeated a last time, movement is executed to the new point and the designated symbol is typed.

One reason for the need to calculate successive steps to the new point is because of mechanical constraints in the terminal device. In the aforementioned combined terminal, the platen and type carriage are driven by bidirectional stepping motors. A finite time after receiving a character stream this terminal must type a symbol. Within this time the stepping motors can be advanced a given number of increments, for example 15. Thus, if the desired new point is greater than 15 increments from the previous point, a "blank" must be typed at the end of the first move of 15 increments. A step thus consists of any desired distance up to a maximum distance of fifteen increments in each direction (x and y).

It is convenient to consider the total distance to be traversed between the given point and the new point as comprising a given number of steps plus a remainder, the remainder being the final distance, less than a full step, to reach the new point. The steps and remainder are traversed in a series of moves. A prime object of the method of the present invention is to arrive within one step of the new point in an optimum mode. While it would be possible to step directly to the new point in a plotting mode, the combined terminal would lay down a "blank" at the desired new point. Thereafter, a new character stream having dx and dy both zero and specifying the symbol to be typed would have to be generated. As will be clear from the ensuing discussion, it requires fewer characters to move within one step of the new point and then generate a new character stream commanding the remainder in x , the remainder in y , and the symbol to be typed.

In accordance with the method of the present invention, after calculating the total incremental distance (Δx and Δy) between the given and desired new points, the number of steps is calculated by dividing the maximum step size (assumed to be 15 for the present terminal device), into the longest of Δx or Δy and truncating to the right of the decimal point. While it is possible to set the step size at any value less than the maximum, the explanation to follow will use the maximum. Thus, for a step size of 15, if the longest of Δx and Δy lies in the range 16 through 29 increments, then the number of steps is one. For values of Δx or Δy in the range 30 through 44, the number of steps is two, and so forth. Thus, for example, if Δx is 39, the number of steps required is two and the remainder is nine increments. If the longest of Δx and Δy is 15 or less, then it is convenient to regard this distance simply as a remainder.

To calculate the distance to be traveled in the direction of the shortest of Δx and Δy for each step of the longest, divide the shortest by the number of steps in the longest and truncate to the right of the decimal point. Thus, extending the above examples, if Δx is 39 and Δy is 21, then for each step (dx) of 15 in x , y should be stepped (dy) 10. Thus, the first step would move $dx=15$ and $dy=10$; the second step would move another $dx=15$ and another $dy=10$, leaving a remainder of $dx=9$ and $dy=1$. The path followed to reach the new point (starting at the origin 0,0) is shown in FIG. 4. Note that each move, whether a step or a remainder, is stated in terms of a dx and a dy with respect to the immediately preceding location.

The quantities calculated in operation 20 are used in operation 40 to select the optimum mode of operation of the "Typagraph." The method of FIG. 3 including the selection of modes is shown in greater detail in FIGS. 5a and 5b. The en-

coding methods illustrated in FIGS. 3, 5a and 5b are directed primarily to analysis and encoding of data for ultimate plotting in a graphical display and are thus concerned primarily with plotting MODES 2, 3 and 4 as opposed to text-typing MODE 1.

In the method of FIGS. 5a and 5b it is necessary to compare the state of the stored characters (the stored characters being shown in FIG. 1 for various modes) with the calculated dx , with the calculated dy , with the quadrant, and with the symbol to be typed. If any of these four characters is different in the next step to be performed from the value stored for the immediately preceding step, then it becomes necessary to change that one of the stored characters and transmit the new character. This change of character is incorporated in a new character stream.

Prior to making this comparison in comparison operation 50 certain decisions have to be made. One pass through the loop of FIGS. 3, 5a and 5b operates on only one step or less, i.e., one move. Thus, if the total distance (Δx or Δy) to the desired new point is more than one step size, then a "blank" symbol must be typed at the end of each step prior to reaching the desired new point. Thus, in operation 21 whether the number of steps is greater than one is first determined. If the number of steps is not greater than one, i.e., if the larger of Δx and Δy is 29 or less, then next it is determined in operation 22 whether the new point is further than the size of one step, i.e., further than either dx or dy . If it is not, then the comparison operation 50 is performed. If, however, it is further than the size of one step, then the symbol to be typed at the end of the first step is set to a "blank" at 23. The operations 23, 25 and 26 will be treated subsequently.

As seen in FIG. 5b, the comparison operation 50 results in a decision at operation 51 that either all stored characters are the same as the new ones or that one or more are different. From this point three general paths may be taken. The first path is taken irrespective of the decision at operation 51 if the number of steps required to reach the desired new point is greater than one as determined at operations 52 or 53. In this first path beginning at operation 61 the mode is ultimately set in either of the faster MODES 2 or 3 at operations 64, 65 or 66.

As to the second path, if the number of steps required to reach the new point is not greater than one, (and if the stored characters were found to be the same as the new characters, then an attempt is made, beginning at operation 67, to use one of the faster MODES 2 or 3 based on the probability, determined from past experience, that MODE 2 or 3 will be proper for the next succeeding new point.

In the third path, if the number of steps required to reach the new point is not greater than one, and if the stored characters were found not to be identical with the new ones, then beginning with operation 54 the mode will be set to MODE 4. However, if one but not both of dx and dy is sufficient to reach the new point (thus requiring one step plus a remainder to reach the new point), then the sufficient one will be set to zero for the single step. Thereafter, in picking up the remainder on the next move the sufficient one of dx or dy will be generated and placed in a counter. It is then contemplated that the next two new points will have an identical sufficient incremental value. If such is the case then these consecutive identical values will be used ultimately to justify a change to a faster mode.

Considering now the second path above, it will be recalled that MODE 4 is the slowest mode. In MODE 4 both dx and dy are variables while in MODE 2 only dy is variable (and dx is stored) and in MODE 3 only dx is variable (and dy is stored). Thus, if moving from a given point, to a new desired point, to yet another new desired point, and so on in MODE 4 it would be desirable to switch to a faster MODE 2 or 3. One method to accomplish this would involve looking ahead through data 10 to ascertain what presently unknown new points are to be plotted and from this information selecting the optimum mode. However, this method requires extensive additional hardware and software and is rather costly.

According to the present invention, the dx values and dy values for a given number of previous moves are maintained in counters (not shown) and are compared in operation 69. If the counters show that one of dx or dy has not changed for the given number of previous moves, then it is reasonable to assume that the odds are it will remain the same for the next move. Thus, if dx , for example, remained constant three consecutive times, then the mode would be changed to MODE 2, dx would be stored and only dy varied. While the present embodiment checks the previous three moves it is understood that this is simply by way of practical example and that any desired number of previous moves may be checked. However, as a greater number of moves are checked and found constant, the odds increase that the next move will remain constant, but obviously the change to the faster mode is delayed. The usefulness of the present method is clearly seen if one is plotting, for example, a y variable such as current as a function of constant steps in x of voltage.

Looking further at FIG. 5b, the attempt to change modes begins after finding at 67 that the current mode is MODE 4. The preceding three moves are checked in operation 69. If the three dx 's are the same, MODE 2 is used in operation 72. If the dx 's are not the same but the dy 's are found in operation 71 to be the same then MODE 3 is selected by operation 73. If, too, the dy 's are not the same, then MODE 4 is retained in operation 74. After each move the counters are updated by operation 75. Of course, if operation 67 determines the current mode to be other than MODE 4, then the current mode is retained in operation 68.

Turning now to the first path above, it is seen that when the number of steps exceeds one (as determined by operations 52 or 53) then MODE 2 or 3 is used. It will be recalled from above that the present embodiment does not look ahead within data 10 to the next new point, i.e., a point not yet subjected to calculation operation 20, to determine optimum mode. However, to the extent that a known desired new point has been ascertained (by operation 20 in FIGS. 3 and 5a) to be more steps than one step away, then such knowledge definitely indicates the desirability of using MODE 2 or 3. Accordingly, in this case the decision to change to a faster mode is based on the calculation of total distance, i.e., the number of moves, to the new point. Thus, after resetting to zero the counters keeping track of the last three moves in operation 61, if neither of dx nor dy is zero (as ascertained by operation 62) then either MODE 2 or 3 is used in operation 64. If, however, dx or dy is zero as ascertained in operations 62 and 63 then a particular one of MODES 2 or 3 is used. If dx is zero, MODE 2 is used in operation 66 while a zero for dy results in a change to MODE 3 in operation 65.

The third path is entered when the number of steps is not greater than one and some change has to be made (since the comparison in 50 found that the stored and calculated characters were not all the same). The third path ultimately uses MODE 4 at 60 to execute any move. Thus, if both dx and dy are sufficient to reach the desired new point, then after updating the counters keeping track of the last three moves (at 55) MODE 4 is employed. Similarly, if neither dx nor dy is sufficient to reach the new point as determined in 54, then MODE 4 is employed. However, if one of dx or dy is sufficient to reach the desired new point and the other is not, then the sufficient one is set to zero and is picked up in the remainder rather than in the first step.

Assume, for example, that the desired new point was spaced from the present point by $\Delta x=16$ and $\Delta y=14$. Thus, there is a first step of $dx=15$ and $dy=14$ and a remainder of $dx=1$ and $dy=0$ as determined by operation 20. The operations 54, 55 and 56 determine for the first move that the $dx=15$ is not sufficient to reach the point $\Delta x=16$, $\Delta y=14$ while $dy=14$ is sufficient. Therefore, according to the invention, instead of moving the first step of $dx=15$ and $dy=14$, rather dy is set to 0 as required by operations 57, 58 and 59. Thus, the first move is $dx=15$ and $dy=0$. As a result, the remainder becomes $dx=1$ and $dy=14$. After the last move has been made to the desired point, the values $dx=1$, and $dy=14$ are the last values stored in

the counters. If, on the next new point, a $\Delta y=14$ reoccurs, as is very possible, then there will be one $\Delta y=14$ already in the counters and the odds are thus better for ultimately obtaining enough consecutive Δy increments as determined by operation 69 to change to a faster mode. Thus, by storing the original $\Delta y=14$, which would not normally have been stored, the later comparison by operation 69 will show three consecutive identical increments earlier.

In all of the three general paths above after selecting a mode, the next step is to output the resulting character stream by operation 80. Thereafter the method is repeated and begins again at 90 (FIG. 5a). If, thereafter, by operations 20 and 20a it is found that the point actually has been reached and typed, then a return at 91 is made to begin the process anew for the next desired new point. If it is calculated in operations 20 and 20a that at least one step remains to reach the original desired new point or that a remainder has to be picked up, then the method of FIGS. 3, 5a and 5b is repeated for each step and for the remainder. Thus, to move a distance greater than one step size, characters are generated to move the output terminal more than once and to type a blank symbol for each move but the last.

In FIG. 5a the operations 24, 25 and 26 handle the special case where both Δx and Δy are even multiples of the step sizes (if, too, the number of steps is greater than one)—for example, where the desired new point is $\Delta x=45$ and $\Delta y=30$. There the step size in x is 15 and in y is 10. Moreover, 45 and 30 are even multiples of 15 and 10 respectively. Thus, without more, the terminal would move in three steps to the desired new point $\Delta x=45$, $\Delta y=30$ and type a blank at each step including the desired new point. Thereafter, a new character stream with Δx and Δy both zero and having the required symbol would be transmitted. It requires fewer characters to include the symbol in the character stream commanding the final move to the desired new point. Therefore, in operation 26 after setting the typed symbol to a blank the number of steps is reduced artificially by one and the distances recalculated so that the terminal stops one step from the desired new point and the artificial remainder is picked up on the last pass. Of course, if the actual symbol to be typed at the desired new point is a "blank" (as determined by operation 25) then operation 26 is eliminated.

Thus, there has been disclosed a method for efficiently encoding data to be transmitted to an output terminal. It will be clear to one skilled in the art that the above-described method may be implemented using a general purpose digital computer. One example of typical software suitable for machine implementation by a general purpose digital computer is disclosed in the Appendix attached hereto which is a Fortran program listing for a Model 940 Time Sharing Computer System (manufactured by Xerox Data Systems, Inc.) which will perform in accordance with the method of the invention.

It will be apparent from the foregoing that, while particular forms of the invention have been illustrated and described, various modifications can be made without departing from the spirit and scope of the invention. Accordingly, it is not intended that the invention be limited, except as by the appended claims.

I claim:

1. For machine implementation, a method of encoding graphical data to be displayed by an output terminal, wherein the terminal responds to a stream of characters representing said data, comprising the operations of:

1. calculating the number of moves required for the terminal to reach the next desired graphical point,
2. selecting from several operating modes the optimum mode for moving toward said point, in each of said several modes at least one character being stored and permitting a character stream to subsequently omit the stored character; and
3. generating a character stream to operate the terminal in the optimum mode to obtain maximum information density in the character stream.

2. The method of claim 1 wherein said optimum mode is selected for executing a first move toward said graphical point, the method further including:

repeating operations (1) through (3) to command an additional move if said first move is insufficient to reach said point.

3. The method of claim 1 wherein:

the calculated number of moves represents a given number of steps in two coordinate directions, the size of each step not exceeding the maximum possible number of increments movable by the terminal in one move, and a remainder, the size of the remainder not exceeding the above said number of increments.

4. The method of claim 3 wherein a symbol is to be displayed at said point and the symbol and the increments of the remainder are encoded together in the character stream and are executed together in the final move.

5. The method of claim 4 wherein at least one step precedes execution of the remainder and for each step the symbol encoded in the character stream represents a blank so that after each step no symbol is visually displayed by the output terminal.

6. The method of claim 1 wherein the operation of selecting the optimum mode further includes:

examining a given number of preceding moves to find for each location in the character stream any characters which are the same for each move; and

after finding the same character in said given number of preceding moves, changing to a mode capable of transmitting a greater information density than the current mode.

7. The method of claim 6 wherein the process of examining said given number of preceding moves is performed for three such moves.

8. The method of claim 3 wherein the operation of selecting the optimum mode includes:

examining the total number of steps and selecting the optimum mode based on the number of steps.

9. The method of claim 5 wherein the operation of selecting the optimum mode includes:

setting to zero the step in one coordinate direction if the incremental distance represented by said step is sufficient to reach said point; and

encoding said incremental distance represented by said step in said character stream only when moving through said remainder.

10. A reiterative process for machine implementation wherein graphical data in the form of points at which symbols are to be plotted, each point being expressed in terms of x and y coordinate distances from a common origin is encoded to be displayed by an incremental graphical display terminal receptive to characters expressing point location by x and y displacements from the immediately preceding point and by a quadrant direction from the immediately preceding point and by a symbol to be displayed at the print location and wherein the display terminal has plural operating modes, in each mode at least one respective character being stored and permitting a character stream to subsequently omit the stored character, said process comprising the operations of:

1. analyzing the data for a desired new point and converting it into a form acceptable to the display terminal by selecting new characters defining a first move toward the desired new point,
2. comparing for each location in the character stream the new characters with the stored values of the characters defining the immediately preceding move to ascertain if any of the new characters are different from those preceding,
3. selecting an operating mode for said first move which compresses the new characters in a character stream of as few characters as possible to obtain maximum information density, and

4. repeating operations (1) through (3) to command an additional move if said first move is insufficient to reach the desired new point.

11. The process of claim 10 wherein the operation of analyzing the data further includes:

- calculating the net distances Δx and Δy in each coordinate axis direction from the immediately preceding point to the new point;
- determining the quadrant of the new point with respect to the preceding point;
- determining a step size in x and y (dx and dy) for said first move of the display terminal; and
- calculating the number of steps required to move within one step size of the new point leaving a remainder to be moved of less than one step size.

12. The process of claim 11 including utilizing a mode in which either dx or dy is stored if the number of steps exceeds a given number.

13. The process of claim 12 wherein said given number is one.

14. The process of claim 11 wherein the operation of analyzing the data further includes:

- setting the symbol to a blank if the distance to the new point is greater than one step size.

15. The process of claim 14 wherein the operation of analyzing the data further includes:

- ascertaining whether said Δx and Δy are both even multiples of the step sizes.

16. The process of claim 15 wherein the operation of analyzing the data further includes:

- setting the symbol to a blank if Δx and Δy are not both even multiples of the step sizes.

17. The process of claim 15 wherein the operation of analyzing the data further includes:

- setting the symbol to a blank if Δx and Δy are both even multiples of the step sizes; and
- reducing the number of steps by one to avoid overshooting the new point.

18. The process of claim 11 wherein the operation of selecting an operating mode includes:

- comparing a given character for each of several preceding moves to determine for each location in the character stream if the character is identical for each move; and
- utilizing an operating mode in which the given character is stored if the character is determined to be identical.

19. The process of claim 18 wherein the given character is compared for each of three preceding moves.

20. The process of claim 11 wherein the operation of selecting an operating mode includes:

- examining the current mode to determine whether either dx or dy is stored.

21. The process of claim 20 wherein the operation of selecting an operating mode further includes:

- using the current mode if, upon examining the current mode, one of dx and dy is found to be stored; and
- attempting to change modes to a mode in which one of dx or dy is stored if, upon examining the current mode, neither one of dx or dy is found to be stored.

22. The process of claim 21 wherein the operation of attempting to change modes includes:

- comparing for each location in the character stream a given character for each of several preceding moves to determine if the given character is identical for each move;
- utilizing an operating mode in which the given character is stored if the character is found to be identical; and
- using the current mode if the character is not found to be identical.

23. The process of claim 22 wherein the operation of comparing a given character compares the character for three preceding moves.

24. The process of claim 11 wherein the operation of selecting an operating mode includes:

- resetting counters keeping track of the last three moves to zero, if the number of steps is greater than one; and

using a mode in which one of dx or dy is stored.

25. The process of claim 24 wherein the operation of selecting an operating mode further includes:

- determining whether either dx or dy is zero;
- if either dx or dy is zero, then using a mode having the zero character of dx or dy stored; and
- if neither dx or dy is zero, then using a mode in which either dx or dy is stored.

26. The process of claim 11 wherein the operation of selecting an operating mode includes:

- setting to zero the step in one coordinate direction if the net incremental distance represented by said step is sufficient to reach said point; and
- encoding said incremental distance represented by said step in said character stream only when moving through said remainder.

27. For machine implementation, a method of encoding graphical data to be displayed by an output terminal, wherein the terminal responds to a stream of characters representing said data, comprising the operations of:

1. calculating the number of moves required for the terminal to reach the next desired graphical point;
2. selecting from several operating modes the optimum mode for moving toward said point, the operation of selecting the optimum mode including:
 - examining a given number of preceding moves to find for each location in the character stream any characters which are the same for each move; and
 - after finding the same character in said given number of preceding moves, changing to a mode capable of transmitting a greater information density than the current mode; and
3. generating a character stream to operate the terminal in the optimum mode.

28. The method of claim 27 wherein the process of examining said given number of preceding moves is performed for three such moves.

29. The method of claim 27 wherein:

- the calculated number of moves represents a given number of steps in two coordinate directions, the size of each step not exceeding the maximum possible number of increments movable by the terminal in one move, and a remainder, the size of the remainder not exceeding the above said number of increments.

30. The method of claim 29 wherein a symbol is to be displayed at said point and the symbol and the increments of the remainder are encoded together in the character stream and are executed together in the final move.

31. The method of claim 30 wherein at least one step precedes execution of the remainder and for each step the symbol encoded in the character stream represents a blank so that after each step no symbol is visually displayed by the output terminal.

32. The method of claim 29 wherein the operation of selecting the optimum mode further includes:

- examining the total number of steps and selecting the optimum mode based on the number of steps.

33. The method of claim 31 wherein the operation of selecting the optimum mode further includes:

- setting to zero the step in one coordinate direction if the incremental distance represented by said step is sufficient to reach said point; and
- encoding said incremental distance represented by said step in said character stream only when moving through said remainder.

34. For machine implementation, a method of encoding graphical data to be displayed by an output terminal comprising the operations of:

1. calculating the number of moves required for the terminal to reach the next desired graphical point, the calculated number of moves representing a given number of steps in two coordinate directions, the size of each step not exceeding the maximum possible number of increments movable by the terminal in one move, and a

11

- remainder, the size of the remainder not exceeding the above said number of increments, and wherein a symbol is to be displayed at said point;
2. selecting from several operating modes the optimum mode for moving toward said point; and 5
3. generating a character stream to operate the terminal in the optimum mode, wherein at least one step precedes execution of the remainder and for each step the symbol encoded in the character stream represents a blank so that after each step no symbol is visually displayed by the output terminal, with the symbol and the increments of the 10

12

- remainder being encoded in the character stream and executed together in the final move; and
- wherein the operation of selecting the optimum mode includes:
- setting to zero the step in one coordinate direction if the incremental distance represented by said step is sufficient to reach said point; and
- encoding said incremental distance represented by said step in said character stream only when moving through said remainder.

* * * * *

15

20

25

30

35

40

45

50

55

60

65

70

75

UNITED STATES PATENT OFFICE
CERTIFICATE OF CORRECTION

Patent No. 3,646,571 Dated FEBRUARY 29, 1972

Inventor(s) KURATA, GEORGE Y.

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

Following the Claims (Columns 11 and 12),
add the Appendix, pages (a) through (g) as
shown on the accompanying prints.

Signed and sealed this 23rd day of January 1973.

(SEAL)
Attest:

EDWARD M. FLETCHER, JR.
Attesting Officer

ROBERT GOTTSCHALK
Commissioner of Patents

MISSING PAGE TEMPORARY NOTICE

PATENT # 3646571 FOR ISSUE DATE 1-23-73

HAS BEEN SCANNED, BUT WITH MISSING PAGE(S). UPON RECEIVING OF MISSING PAGE(S), THE ENTIRE DOCUMENT WILL BE RESCANNED. PLEASE CALL IMAGE DATA ADMINISTRATION STAFF OF 557-6154 IF YOU HAVE A QUESTION. ASK FOR DAVE GROOMS, ANITA YOUNG OR POLA JONES.

THIS NOTICE IS FOR THE MISSING PAGE CONTAINING:

Page (a) of calc

UNITED STATES PATENT OFFICE
CERTIFICATE OF CORRECTION

Patent No. 3,646,571 Dated February 29, 1972

Inventor(s) KURATA, GEORGE Y.

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

```
202 IOD=1
21 IF(MOVEY)22,23,24
22 MOVEY=-MOVEY
   IF(IOD)221,222,224
221 IOD=3
   GO TO 25
222 IF(IBK(2)-3)224,223,223
223 IOD=IBK(2)
   GO TO 25
224 IOD=4
   GO TO 25
23 IF(IOD)231,223,233
231 IF(IBK(2)-3)232,223,221
232 IF(IBK(2)-2)221,223,221
233 IF(IBK(2)-3)234,224,223
234 IF(IBK(2)-1)224,223,224
24 IF(IOD)241,242,243
241 IOD=2
   GO TO 25
242 IF(IBK(2)-2)243,241,243
243 IOD=1
25 IF(MOVEX+MOVEY+MFLG)100,100,26
26 IF(MOVEX-MOVEY)27,28,28
27 IXFLAG=0
   MAX=MOVEY
   MIN=MOVEX
   GO TO 30
28 IXFLAG=1
   MAX=MOVEX
   MIN=MOVEY.
30 DO 35 J=1,15
```

(b)

UNITED STATES PATENT OFFICE CERTIFICATE OF CORRECTION

Patent No. 3,646,571 Dated February 29, 1972

Inventor(s) KURATA, GEORGE Y.

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

```

K1=17-J
· NSTEP=MAX/ISTEP(K1)
· IF(NSTEP)100,35,40
35  CONTINUE
    K1=1
    K2=1
    NSTEP=1
    GO TO 44
40  K2=MIN/NSTEP+1
    IF(K2-16)41,41,401
401 K2=16
41  IF(IXFLAG)42,44,42
42  IDX=ISTEP(K1)
    IDY=ISTEP(K2)
    GO TO 441
44  IDY=ISTEP(K1)
    IDX=ISTEP(K2)

441 IF(IROT)442,45,45
442 IX=IDX
    IDX=IDY
    IDY=IX
    IX=MOVEX
    MOVEX=MOVEY
    MOVEY=IX
45  ISTEP=1
    MDE=-1
    ISYM=ISYMB
    IXX=IDX+1
    IYY=IDY+1
    IXZ=1
    IYZ=1
```

(c)

UNITED STATES PATENT OFFICE
CERTIFICATE OF CORRECTION

Patent No. 3,646,571 Dated February 29, 1972

Inventor(s) KURATA, GEORGE V.

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

```
IX=NSTEP*IDX
IY=NSTEP*IDY
IF(MOVEX-IX)100,450,4501
450 IXZ=0
4501 IF(MOVEY-IY)100,451,452
451 IYZ=0
452 ISUFF=IXZ+IYZ
IF(NSTEP-1)100,4531,453
453 ISTEP=0
ISYM=IBLNK
NSTEP=NSTEP-1
IF(ISUFF)100,4530,4536
4530 IF(ISYMB-IBLNK)4533,4536,4533
4531 IF(ISUFF)100,4536,4532
4532 MDE=0
IF(MDE)45321,45321,4536
45321 ISYM=IBLNK
GO TO 4536
4533 ISUFF=2
IX=NSTEP*IDX
IY=NSTEP*IDY
IF(NSTEP-1)100,4535,4534
4534 NSTEP=NSTEP-1
GO TO 4536
4535 ISTEP=1
4536 IF(IBK(1))4537,4537,4538
4537 LSTEP=-1
GO TO 454
4538 LSTEP=1
454 IF(IBK(2)-IQD)4549,4540,4549
4540 IF(IBK(1)-2)4542,4546,4541
```

(d)

UNITED STATES PATENT OFFICE
CERTIFICATE OF CORRECTION

Patent No. 3,646,571 Dated February 29, 1972

Inventor(s) KURATA, GEORGE Y.

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

4541 IF(IBK(1)-3)4548,4544,4542
4542 IF(IBK(3)-IDX)4548,4543,4548
4543 IF(IBK(1)-4)4544,4546,4548
4544 IF(IBK(4)-IDY)4548,4545,4548
4545 IF(IBK(1)-1)4548,4547,4546
4546 IF(MDE)45462,45461,45462
45461 ISYM=IBLNK
45462 IF(IBK(5)-ISYM)4548,4547,4548
4547 KOK=1
GO TO 4552
4548 KOK=0
4549 KOK=-1
4552 IF(MFLG)100,48,4553
4553 MFLG=0
IF(KOK)46,46,47
46 IF(ISTP)100,461,463
461 LTSP=-1
IF(IDX*IDY)100,462,483
462 IF(IDY)100,483,484
463 IF(ISUFF-1)482,464,481
464 IF(IYZ)100,465,466
465 IY=0
IYY=1
GOTO 481
466 IX=0
IXX=1
GOTO 481
47 IF(ISTP)100,461,471
471 IF(IBK(1)-2)472,472,474
472 ILX=0
ILY=0

(e)

UNITED STATES PATENT OFFICE CERTIFICATE OF CORRECTION

Patent No. 3,646,571 Dated February 29, 1972

Inventor(s) KURATA, GEORGE Y.

It is certified that error appears in the above-identified patent and that said Letters Patent are hereby corrected as shown below:

```
DO 4721 L=1,3
  ILX=ILX+IABS(IDX-LX(L))
4721  ILY=ILY+IABS(IDY-LY(L))
      IF(ILX*ILY)100,4722,482
4722  KOK=-1
      IF(ILX)100,484,473
473   IF(ILY)100,483,482
474   IF(ISUFF)100,48,475
475   IF(IBK(1)-3)100,466,465
48    MDE=IBK(1)
      GO TO 49
481   LSTP=0
482   MDE=2
      GO TO 49
483   MDE=3
      GO TO 49
484   MDE=4
49    NOUT=1
      IF(MOFLAG)491,490,491
490   NOUT=2
491   IF(IBK(1)-MDE)492,4911,492
4911  IF(KOK)492,493,50
492   LSTP=-1
493   IDEC(1)=MOFLG
      IDEC(2)=MDOD(MDE,IOD)
      IDEC(3)=IDEL(IXX)
      IDEC(4)=IDEL(IYY)
      IDEC(5)=ISYM
      II=0
      DO 494 I=NOUT,5
      II=II+1
```

(f)

MISSING PAGE TEMPORARY NOTICE

PATENT # 3646571 FOR ISSUE DATE 1-23-73

HAS BEEN SCANNED, BUT WITH MISSING PAGE(S). UPON RECEIVING OF MISSING PAGE(S), THE ENTIRE DOCUMENT WILL BE RESCANNED. PLEASE CALL IMAGE DATA ADMINISTRATION STAFF OF 557-6154 IF YOU HAVE A QUESTION. ASK FOR DAVE GROOMS, ANITA YOUNG OR POLA JONES.

THIS NOTICE IS FOR THE MISSING PAGE CONTAINING:

Page (g) of c of c