(12) UK Patent Application (19) GB (11) 2 147 474 A

(43) Application published 9 May 1985

(21) Application No 8406187

(22) Date of filing 9 Mar 1984

(30) Priority data

(31) 58/183071	(32) 3 Oct 1983	(33) JP
58/183072	3 Oct 1983	
58/183073	3 Oct 1983	
58/183074	3 Oct 1983	
58/183075	3 Oct 1983	

(51) INT CL⁴
H04N 1/411

(52) Domestic classification
H4F EM S25R S30K S42C S42M S7

(56) Documents cited
None

(58) Field of search
H4T
H4F
B6W

(71) Applicant
Shaken Co Ltd (Japan),
26–13 Minamiotsuka 2 Chome, Toshima-ku, Tokyo, Japan

(72) Inventors
Hiroyuki Shibata
Masatake Takashima
Shinichiro Fukuda

(74) Agent and/or Address for Service
Batchellor Kirk & Eyles,
2 Pears Tree Court, Farringdon Road, London EC1R 3DS

(54) Method of processing character or pictorial image data

(57) A method of processing character data includes approximating the outline of a character with a set of functional curves or straight lines, then storing the outline specifying information to compress the amount of data, and decoding the compressed data to reproduce the character image. Sample points on the outline are selected so that the vector joining adjacent points deviates from the outline by less than a predetermined amount. Vectors having greater than a predetermined length are deemed to correspond to straight lines and are encoded as such. Vectors shorter than the predetermined length correspond to curved portions and are encoded by segmental polynomial approximation.

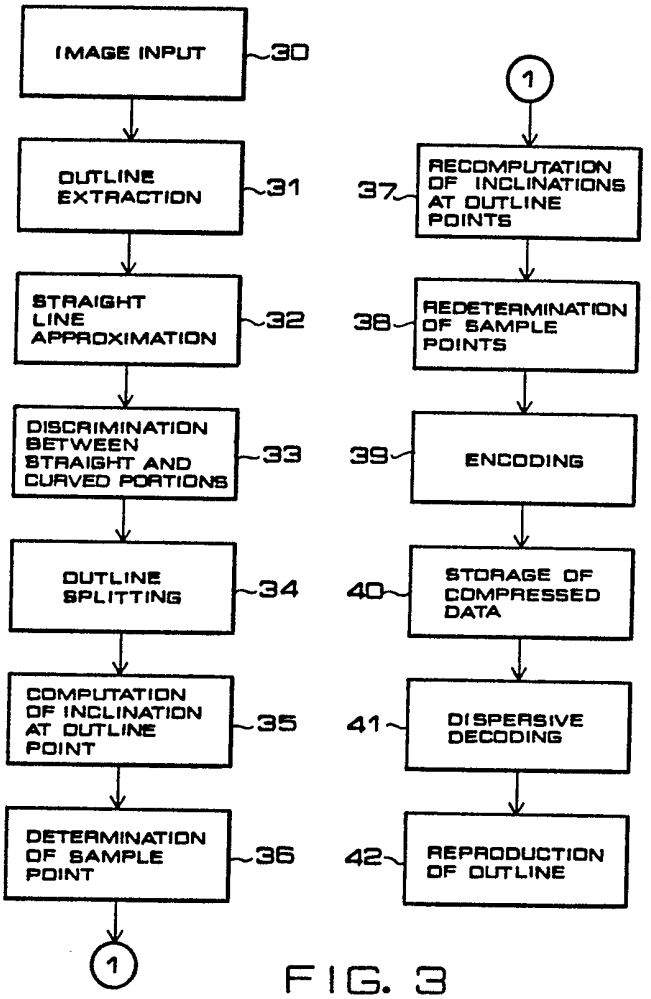


FIG. 3

GB 2 147 474 A

2147424

FIG. 1

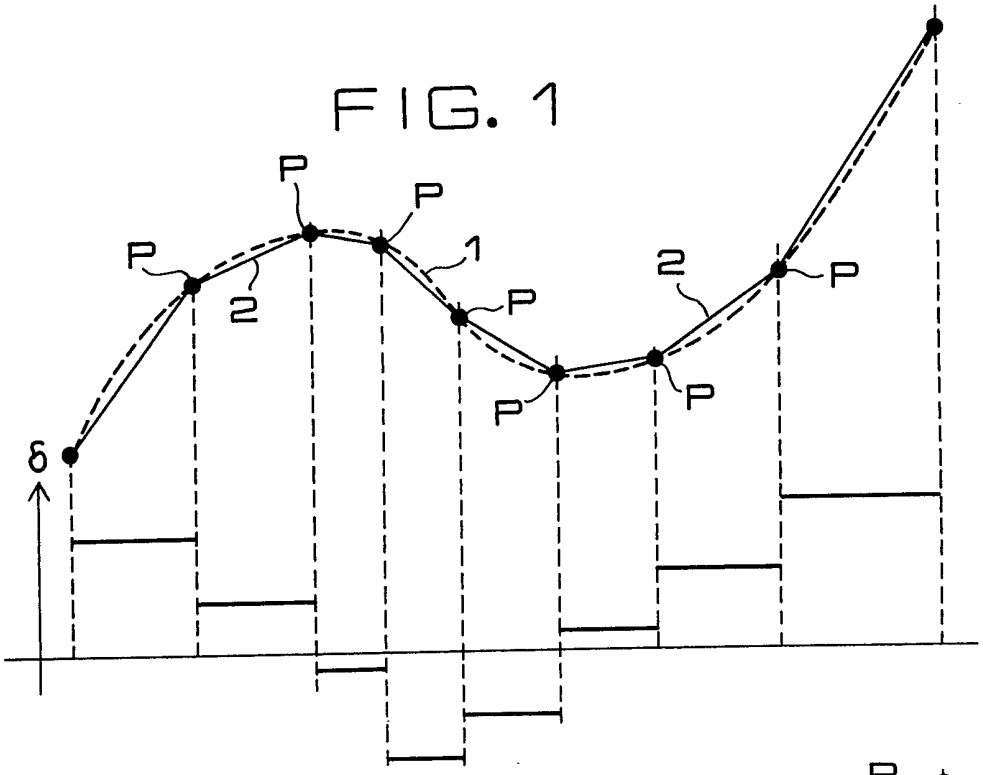
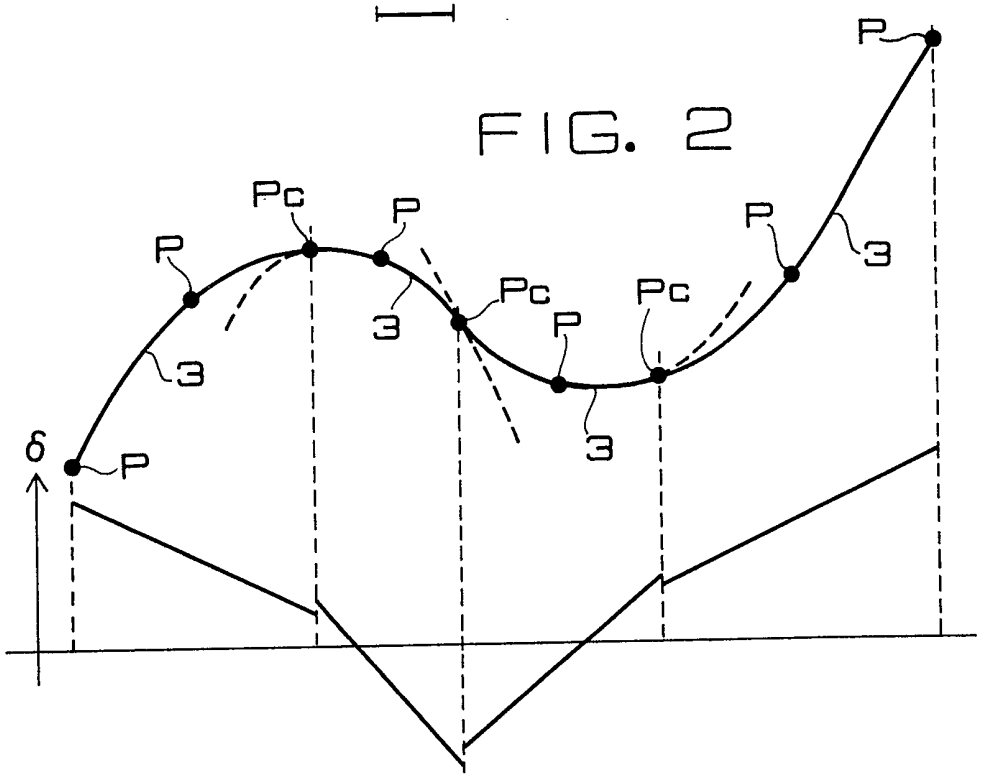


FIG. 2



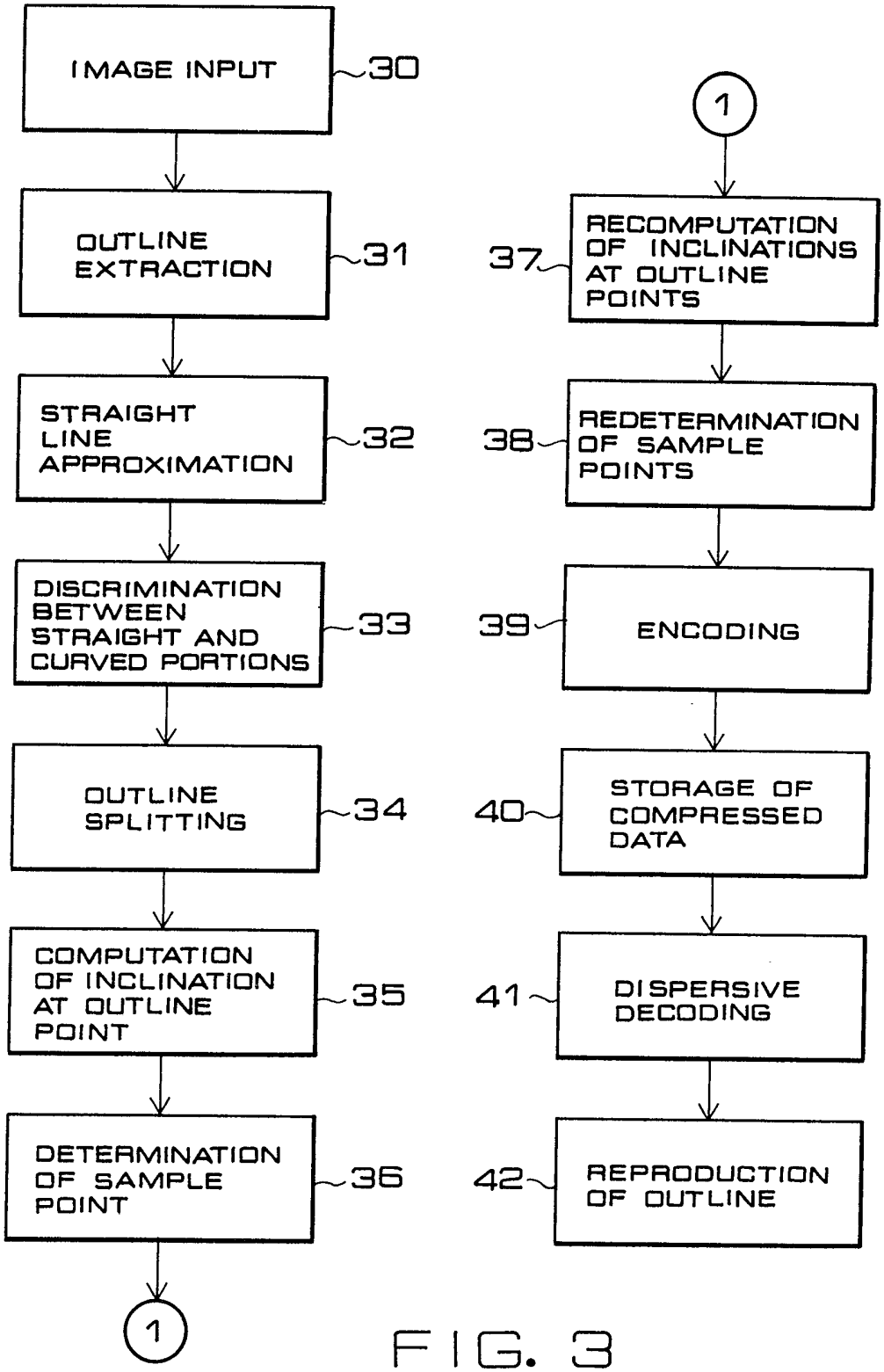

IMAGE INPUT
30
OUTLINE EXTRACTION
31
STRAIGHT LINE APPROXIMATION
32
DISCRIMINATION BETWEEN STRAIGHT AND CURVED PORTIONS
33
OUTLINE SPLITTING
34
COMPUTATION OF INCLINATION AT OUTLINE POINT
35
DETERMINATION OF SAMPLE POINT
36
1
1
RECOMPUTATION OF INCLINATIONS AT OUTLINE POINTS
37
REDETERMINATION OF SAMPLE POINTS
38
ENCODING
39
STORAGE OF COMPRESSED DATA
40
DISPERSIVE DECODING
41
REPRODUCTION OF OUTLINE
42

FIG. 3

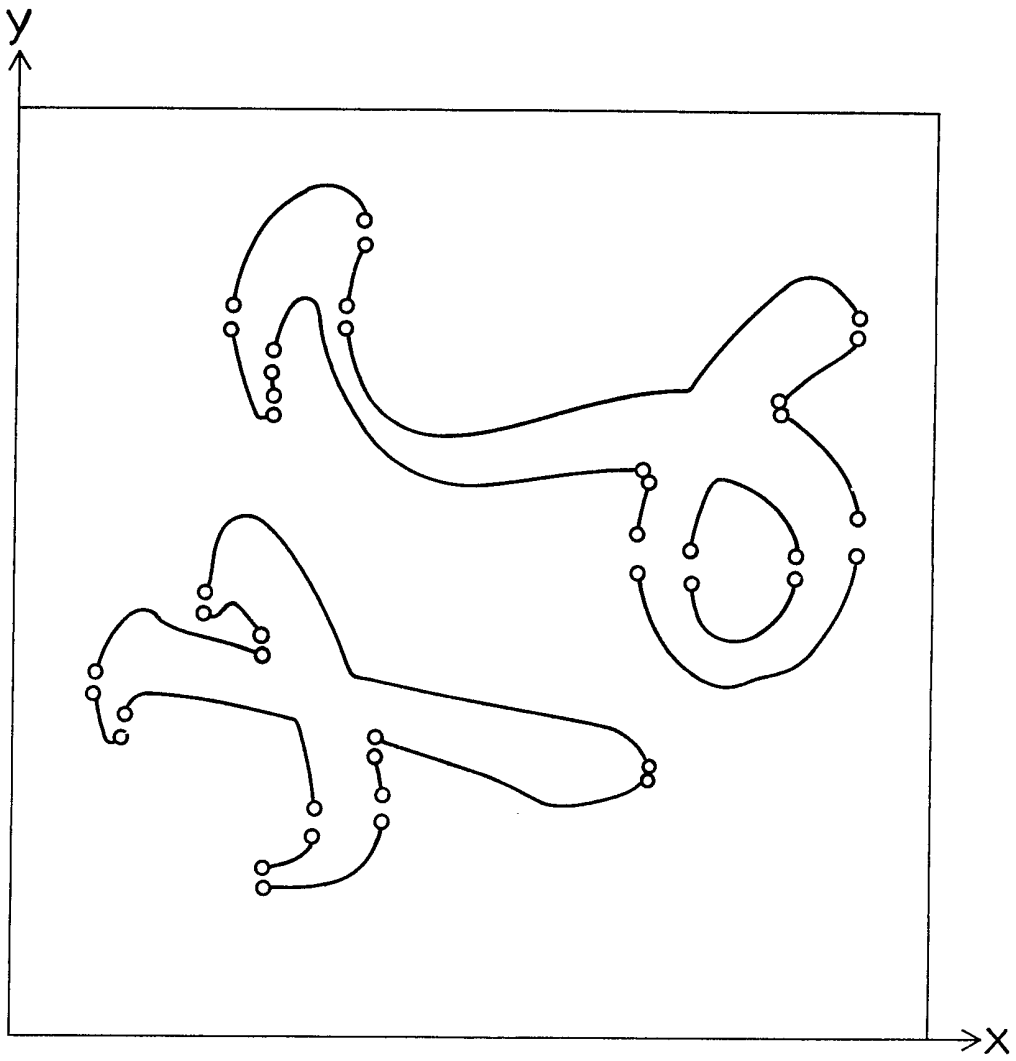


FIG. 4

FIG. 5

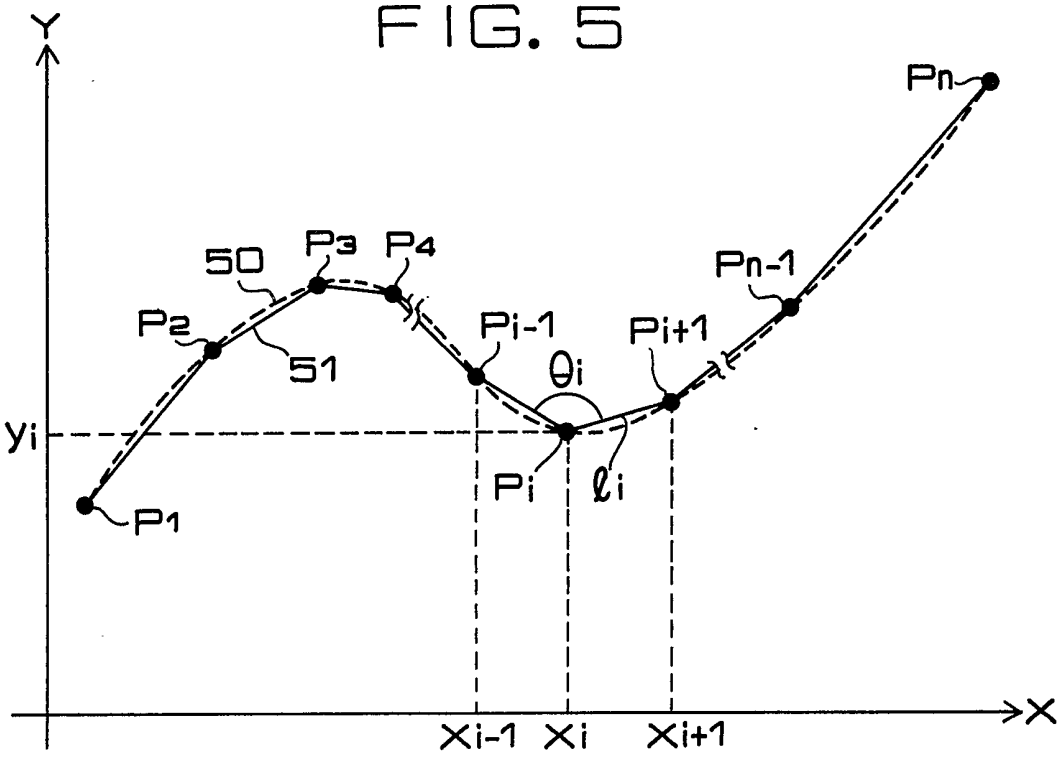


FIG. 10

BLOCK HEADER
SEGMENT HEADER
SEGMENT INFORMATION
SEGMENT HEADER
SEGMENT INFORMATION
≀
SEGMENT HEADER
SEGMENT INFORMATION

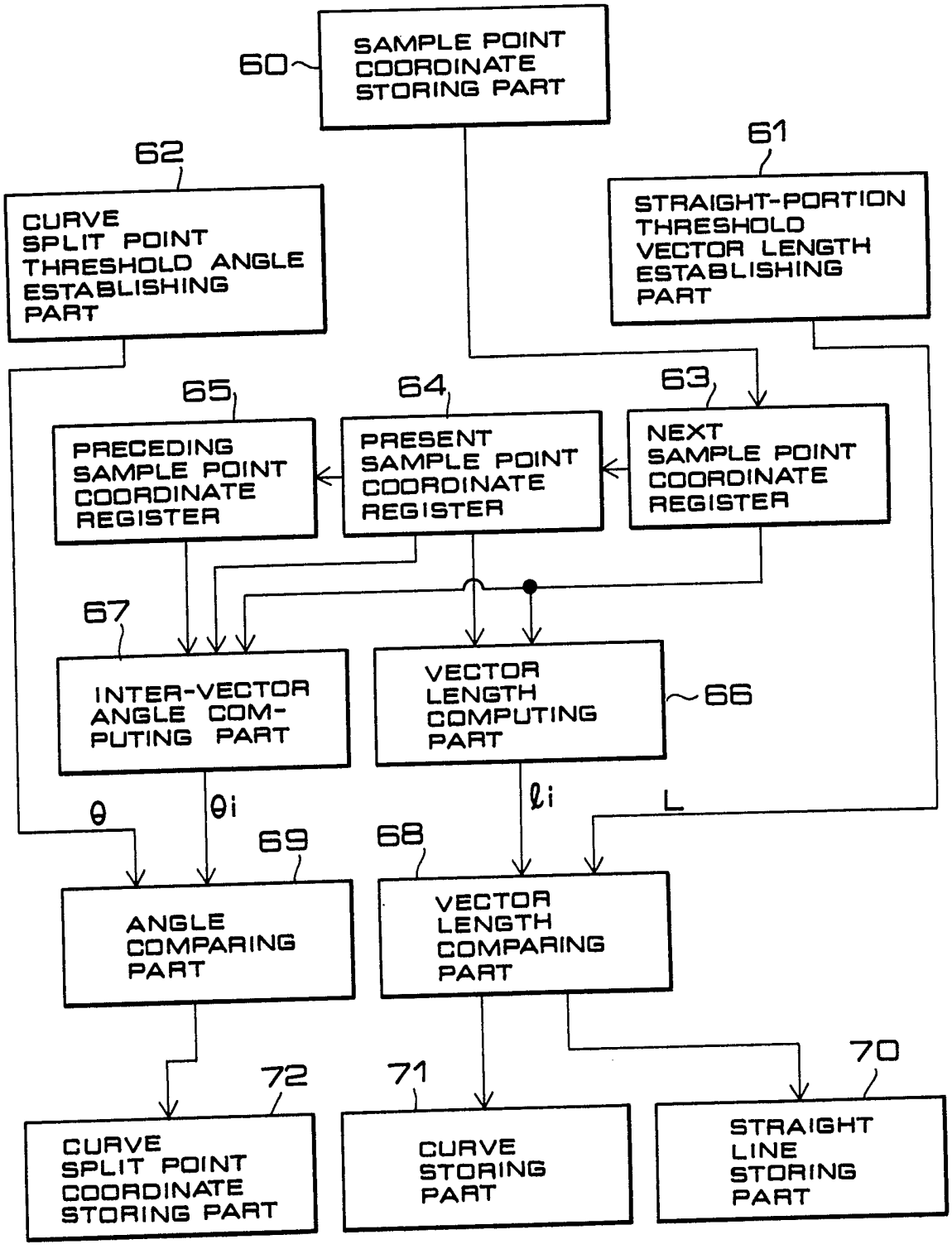

60
SAMPLE POINT COORDINATE STORING PART
62
CURVE SPLIT POINT THRESHOLD ANGLE ESTABLISHING PART
61
STRAIGHT-PORTION THRESHOLD VECTOR LENGTH ESTABLISHING PART
65
PRECEDING SAMPLE POINT COORDINATE REGISTER
64
PRESENT SAMPLE POINT COORDINATE REGISTER
63
NEXT SAMPLE POINT COORDINATE REGISTER
67
INTER-VECTOR ANGLE COMPUTING PART
VECTOR LENGTH COMPUTING PART
66
θ
θi
ℓi
L
69
68
ANGLE COMPARING PART
VECTOR LENGTH COMPARING PART
72
71
70
CURVE SPLIT POINT COORDINATE STORING PART
CURVE STORING PART
STRAIGHT LINE STORING PART

FIG. 6

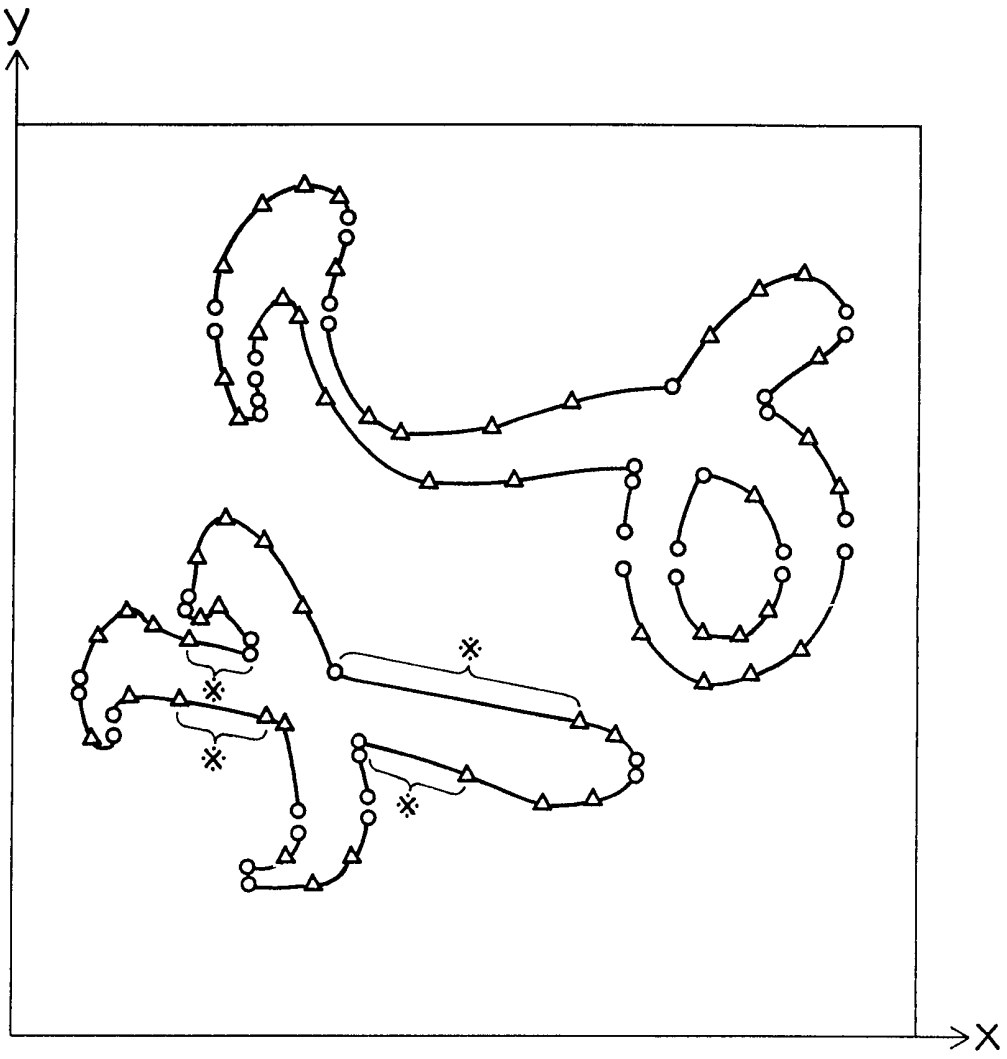


FIG. 7

2147474

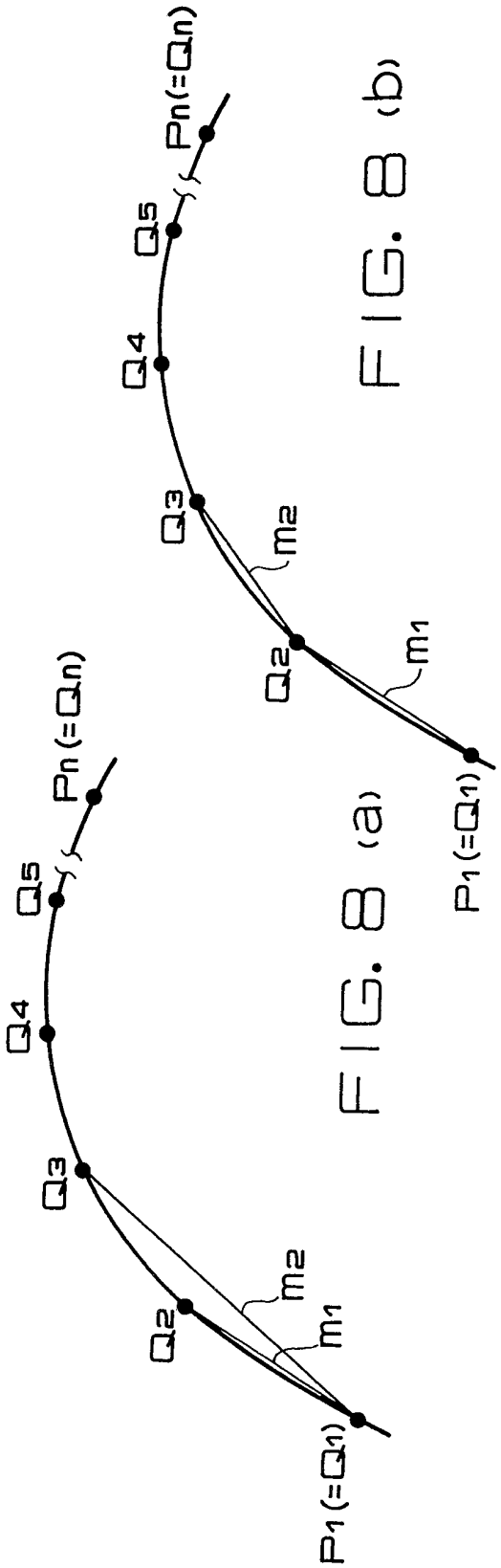


FIG. 8 (a)

FIG. 8 (b)

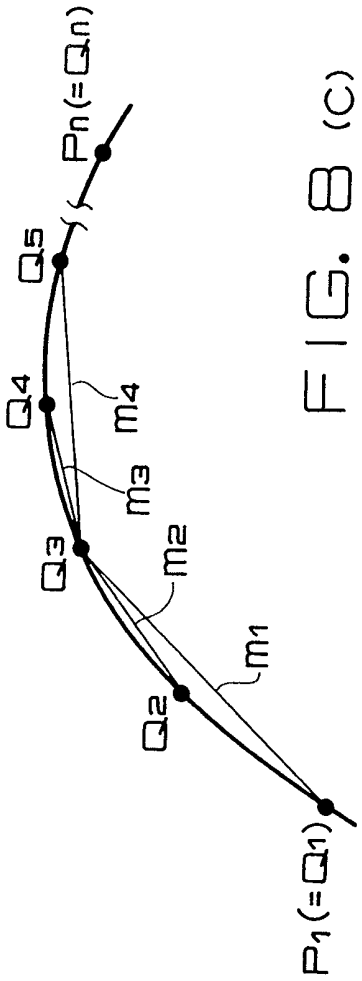


FIG. 8 (c)

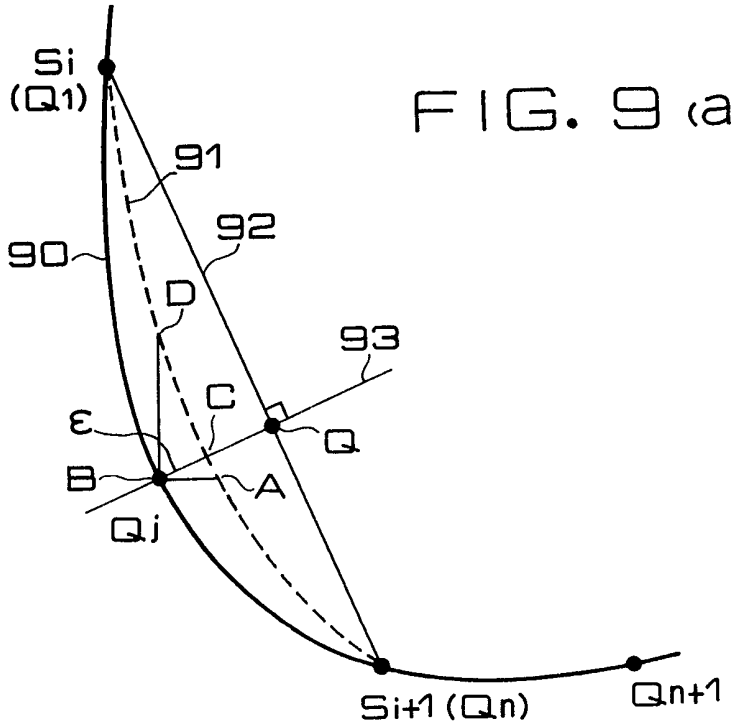


FIG. 9 (a)

FIG. 9 (b)

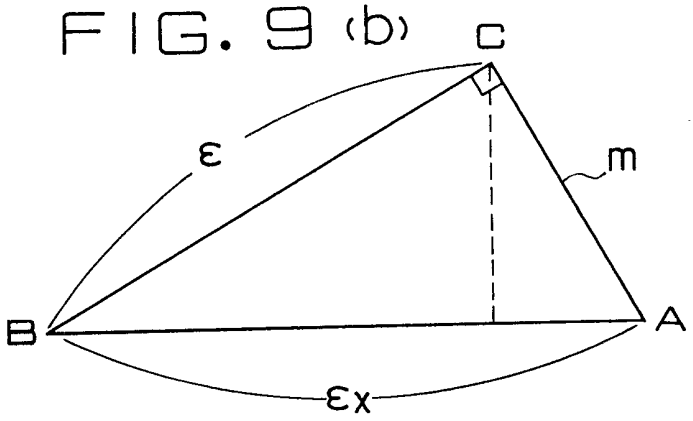


FIG. 11 (a)

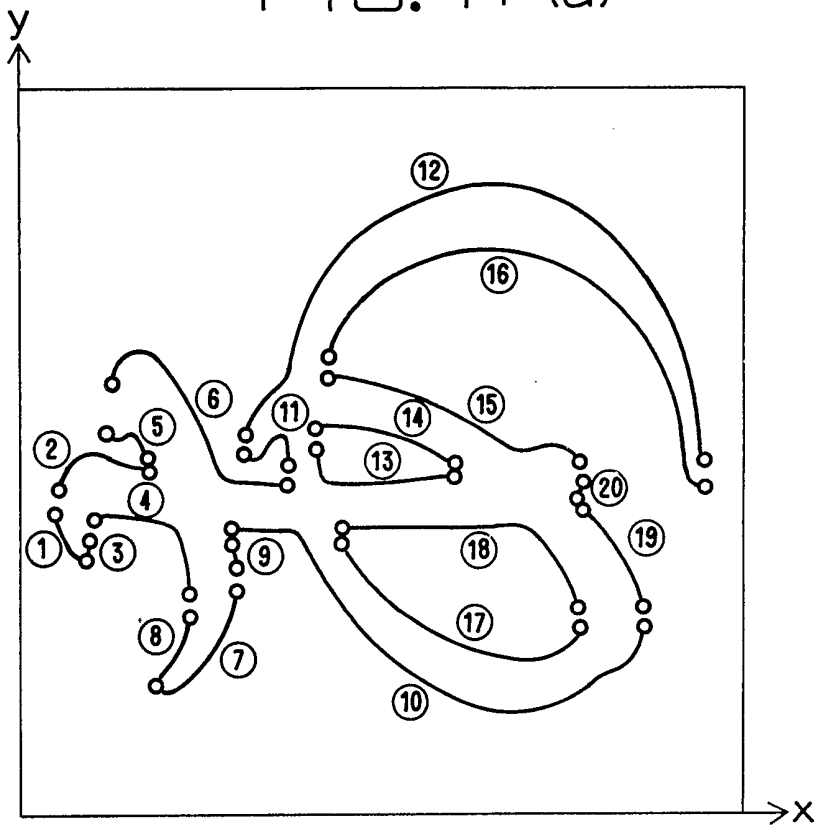
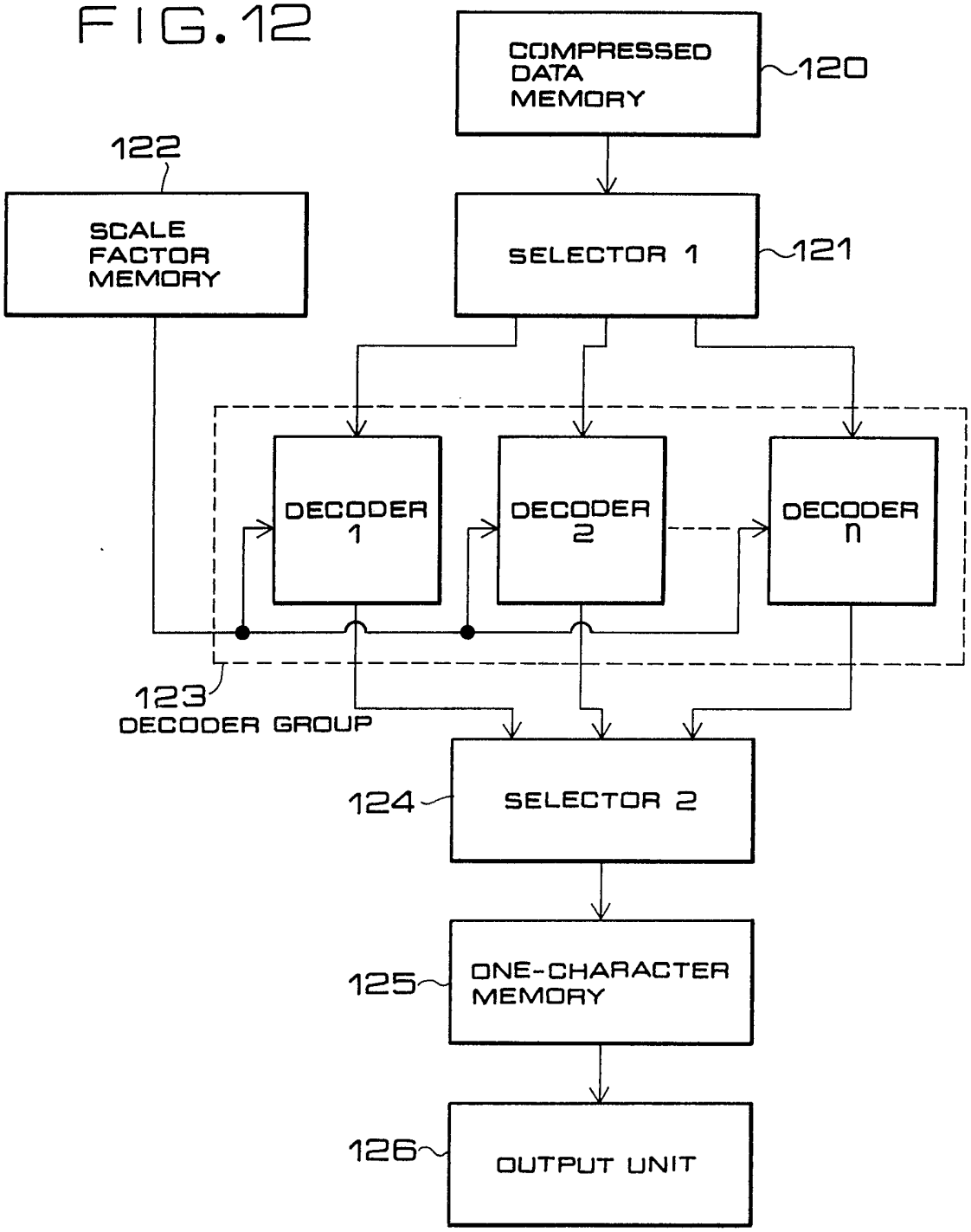


FIG. 11 (b)

BLOCK DATA (12)
BLOCK DATA (10)
BLOCK DATA (16)
≀
BLOCK DATA (20)
BLOCK DATA (3)

FIG.12


COMPRESSED DATA MEMORY
120
122
SCALE FACTOR MEMORY
SELECTOR 1
121
DECODER 1
DECODER 2
DECODER n
123
DECODER GROUP
124
SELECTOR 2
125
ONE-CHARACTER MEMORY
126
OUTPUT UNIT

2147474

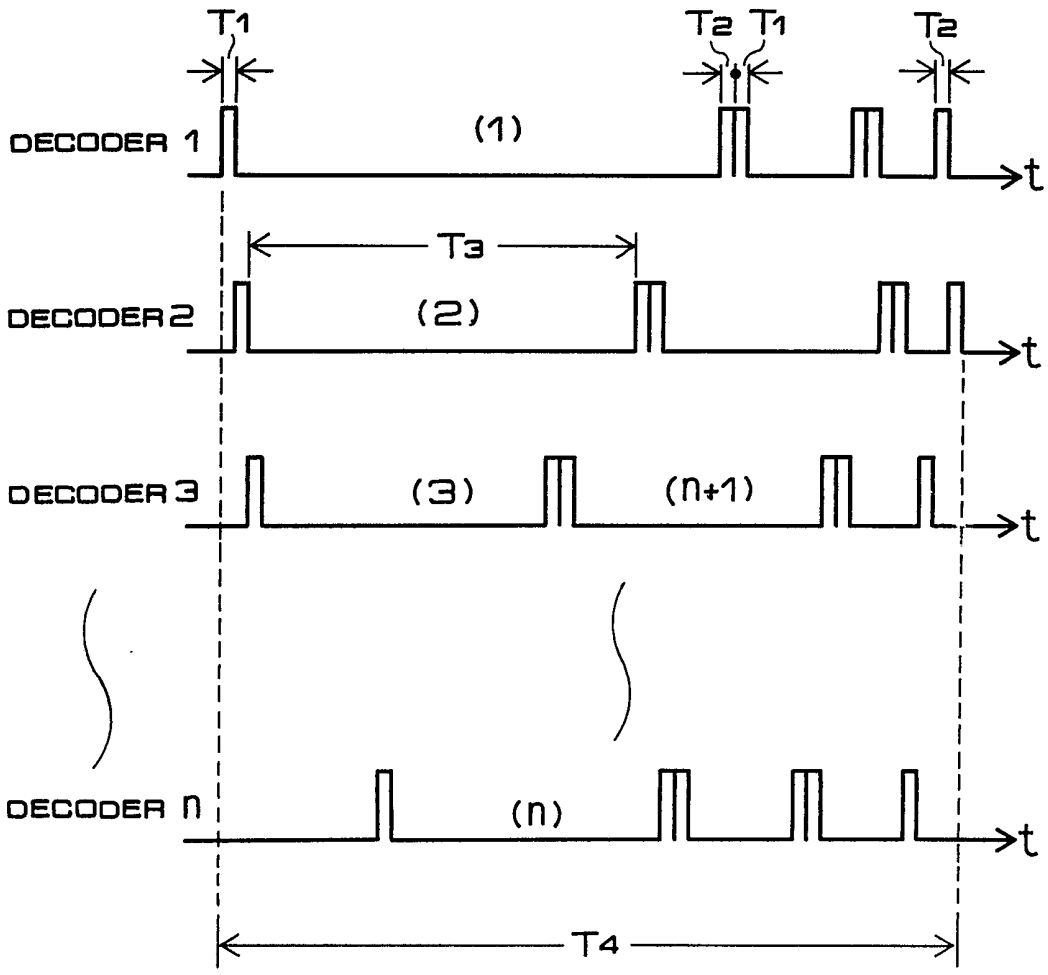


FIG. 13

SPECIFICATION

Method of processing character or pictorial image data

The present invention relates to a method of compressing the data of a character, pictorial image or the like (hereinafter referred to as character) and, more particularly, to a method of processing character data (such as Japanese text characters) by approximating the outline of a character with a set of functional curves or straight lines, then storing the outline-specifying information to compress the amount of data, and decoding the compressed data to reproduce the character image.

It is well known that binary data obtained by resolving a character into dots has an extremely high redundancy. In order to reduce such redundancy, there have been proposed a variety of data compression methods heretofore.

One of the prior data compression techniques is a so-called outline method which compresses the amount of data by ascertaining the shape of a character in accordance with its outline and storing the information required to specify the outline.

In the process of data compression carried out on the basis of such an outline method, two known techniques which have been employed are those of straight line (vector) approximation shown in Fig. 1 and n-degree curve approximation shown in Fig. 2.

The straight line approximation illustrated as an example in Fig. 1 is based on the technique disclosed in Japanese Patent Laid-Open nos. 149522/1979 and 79154/1980, according to which the data compression is performed in such a manner that an outline 1 of a character plotted by a dotted line is first approximated with a set of two-dimensional vectors 2 represented by straight lines, and the information for specifying each vector (position of initial point, length and inclination, or horizontal and vertical components) is stored to serve as data.

Another example of n-degree curve approximation illustrated in Fig. 2 is the method contrived by the present applicant in Japanese Patent Application No. 116160/1980 (Laid-Open no. 39963/1982), according to which the amount of data is compressed by storing the coordinates of a group of points P established suitably on the outline of a character, and a desired outline is approximated with a set of n-degree curve elements 3 which connect (n + 1) points in succession.

It is a feature of data compression systems based on such outline methods that, when reproducing a character image by decoding the compresssed data thereof a variety of scale factors can be obtained in the reproduced image by interpolation or conversion of a vector scale factor.

In such prior methods as those mentioned above, however, there is a basic disadvantage that an optimal result is not guaranteed with regard to the smoothness of the outline (continuity in inclination of the outline), as is obvious from the example of Fig. 1 where the inclination angles δ of the line segments at the terminal points P of the individual vectors are discontinuous, and also from the example of Fig. 2 in which the inclination angles δ of the left and right line segments on the two sides of each connecting point Pc of the n-degree curves 3 are discontinuous.

In contrast therewith, the outline of a character is generally continuous itself and further has such contour characteristics that its first degree derived function (the gradient of the outline) varies continuously with the exception of some peculiar points such as the intersections of constituent lines of the character and the tapered ends of portions which are known as 'hane' in Japanese.

Consequently, it is unavoidable that these conventional data compression methods give rise to some problems such as difficulties in attaining satisfactory compressed data faithful to the original character outline and also in accurately eliminating the unnaturalness (e.g. discontinuity in contour) of the character image reproduced on the basis of such compressed data.

In an attempt to solve the problems mentioned above, the present applicant has previously developed an improved method disclosed in Patent Application No. 16884/1982 (Laid-Open No. 134745/1982).

However, the improved data compression method still has the following disadvantages.

(1) As a result of the attempt to approximate the whole of the outline at once, discontinuous shapes are prone to be reproduced with a deviation from the original outline at a point of connection between a straight portion and a curved portion. In order to avoid such drawbacks, a large number of sample points need to be established with the requirement of division into many polynomials to execute the approximation, so that the amount of compressed data tends to be increased.

(2) Due to the necessity for complex computations to obtain the compressed dat to be stored, a considerable period of time is required to produce the compressed data.

The present invention seeks to provide an improved outline data compressing method, and to provide a character data processing method adapted to produce satisfactory compressed data

faithful to a character outline despite taking less time to compute the compressed data.

The invention further seeks to provide a character data processing method capable of achieving a high data compression rate in addition to accurate storage of the contours of a character outline.

The invention further seeks to provide a character data processing method which, in the mode of reproducing a character image by decoding the compressed data, is capable of faithfully reproducing the character at any of various scale factors to obtain a desired character image with a smooth outline.

Accordingly the present invention provides a method of processing character or pictorial image data to specify the outline of the character or pictorial image developed on x- and y-coordinates so that the encoded data may be subsequently decoded to reproduce the original character or pictorial image, said method comprising: splitting said outline into blocks each defined by univalent function involving x as a variable: computing, for each of said blocks, a plurality of vectors which extend sequentially along the outline of the character or pictorial image and are so established as to be of the maximum possible length consistent with the requirement that the deviation between each vector and said outline is maintained to be less than a predetermined allowable error; and treating as a straight line any outline portion with a vector longer than a predetermined threshold and encoding said portion by a code adapted for a straight line, while treating as a curve any outline portion with a vector shorter than said predetermined threshold and encoding said portion by segmental polynomial approximation.

Some embodiments of the invention will now be described by way of example with reference to the accompanying drawings.

Figures 1 and 2 illustrate data compression based on the prior outline method;

Figure 3 is a flow chart summarising the method of data according to this invention;

Figure 4 illustrates an exemplary outline split into blocks each defined by a univalent function involving x as a variable;

Figure 5 graphically shows a straight line approximation;

Figure 6 is a block diagram of an embodiment contrived to carry out discrimination between a straight portion and a curved portion of an outline according to this invention;

Figure 7 illustrates the result of such discrimination executed by the method of this invention;

Figure 8 plots the process of computing the inclination at each of outline points by the method of this invention;

Figure 9 shows how sample points are determined in this invention;

Figure 10 shows an exemplary format for storage of block data;

Figure 11 illustrates how block data are stored according to this invention;

Figure 12 is a block diagram of an exemplary constitution for carrying out reproduction of block data according to this invention; and

Figure 13 is a timing chart showing the operation performed in Fig. 12.

The method of data processing, with which this invention is concerned, will now be summarised with reference to the flow chart shown in Fig. 3.

An input character image is resolved into dots in the form of an x-y matrix (30), and the outline of the character image thus resolved is extracted (31). Then the extracted outline is approximated with straight-line vectors in such a manner that the positional deviation of each vector from the outline becomes smaller than an allowable error (32). The approximated outline is processed for discrimination between straight portions and curved portions in accordance with the lengths of the individual vectors (33), and the split points of the curved portion are determined together with the intersection angle of the mutually adjacent vectors (34). The shape of the outline segment corresponding to the straight portion is represented by an n-degree polynomial ($n = 1$) while the shape of the outline segment corresponding to the curved portion is approximated with a curve by a single or plural n-degree polynomial ($n = 2, 3$). The curve approximation is carried out by first computing the inclination at each of the outline points forming the outline (35), and then computing an n-degree polynomial on the basis of such inclination and the coordinates of the two ends of the segment which is to be approximated with a curve. Subsequently, with extension of the said segment, a computation is executed to define a sample segment in such a manner that its length becomes maximal within a range where the deviation between the outline and the n-degree polynomial for approximating the said segment is maintained to be smaller than the allowable error, and the two ends of the said segment are determined as sample points (36). Sample segments posterior thereto are defined sequentially in the same way as the above and, after obtaining an n-degree polynomial for approximating each of such sample segments, a first-degree derived function of each n-degree polynomial is found to compute the inclination again at each outline point (37). Then a sample point is determined in the same manner as the foregoing on the basis of the newly computed inclination at each outline point and the coordinate values thereof, and an n-degree polynomial more faithful to the outline is computed (38). The coefficients and the degrees in the n-degree polynomial thus obtained are encoded (39). Furthermore, in order to achieve efficient reproduction of the

character outline thereafter, respective block data are stored in sequence from the block having the longest distance between the initial and terminal points in its x-direction (40). The compressed data of the character outline thus stored are decoded dispersively by a plurality of decoders (41), and the desired character outline is reproduced on the basis of the results of such decoding (42).

Hereinafter the data processing performed in each step will be described in detail.

[Image input (30)]

A character or pictorial image is resolved into x-y matrix dots through raster scanning by means of a scanner or the like, and bit pattern data acquired therefrom is fed as original character data to be processed.

[Outline extraction (31)]

The outline is obtained by detecting dot positions (outline points) where the binary data corresponding to the image-resolved dots change from "0" to "1" or from "1" to "0" in the x- or y-direction. The outline thus obtained is split into blocks each defined by a univalent function involving x as a variable, whereby a set of blocks are specified.

Fig. 4 illustrates how the outline extracted from the character data resolved into x-y matrix dots is split into univalent-function blocks (each extending from "o" to "o").

[Straight line approximation (32)]

In one block on the outline, straight line approximation is performed with a multiplicity of vectors so established as to be maximal in length within a range where the deviation from the outline is maintained to be smaller than a predetermined allowable error.

Fig. 5 shows an exemplary straight line approximation executed in a block [P_1, P_n) with a set of two-dimensional vectors 51 for a character outline 50 represented by a dotted line. The connecting points P_1, P_2, . . . , P_{n-1}, Pn of the individual vectors 51 are stored as sample points in a sample point coordinate storing part 60 which will be described later.

[Discrimination between straight portion and curved portion of character outline (33)]

As mentioned already, the outline of a character generally has straight portions and curved portions. According to the prior art which attempts to approximate the entire outline at a time, a large number of sample points need to be established in the vicinity of the connecting points in the outline where straight portions and curved portions are contiguously existent, and further division into many approximate expressions is necessary, so that the amount of data tends to increase resulting in an undesirable reduction in the compression rate.

In view of the circumstances described above, the present invention has been contrived in order to eliminate such disadvantages by discriminating between the straight portion and the curved portion of the outline in accordance with the lengths of the respective vectors obtained by the aforesaid straight line approximation, and then processing the straight portion and the curved portion individually.

Fig. 6 is a block diagram of an exemplary system designed to implement the discrimination between a straight portion and a curved portion of a character outline. In this diagram, there are shown a sample point coordinate storing part 60 for storing the sample points of each block obtained by the straight line approximation (32); a straight-portion threshold vector length establishing part 61 for establishing a threshold length L to discriminate a vector; a curve split point threshold angle establishing part 62 for establishing a threshold angle ϕ to discriminate a curve split point; a next sample point coordinate register 63 for holding the coordinates (x_{i+1}, y_{i+1}) of an (i + 1)th sample point P_{i+1}; a present sample point coordinate register 64 for holding the coordinates (x_i, Y_i) of an i-th sample point P_i; a preceding sample point coordinate register 65 for holding the coordinates (x_{i-1}, y_{i-1}) of an (i − 1)th sample point P_{i-1}; a vector length computing part 66 for computing the vector length l_i of a sample segment (x_i, x_{i+1}) from the respective coordinates held in the next sample point coordinate register 63 and the present sample point coordinate register 64; an inter-vector angle computing part 67 for computing the inter-vbector angle θi at the same point P_i from the respective coordinates held in the registers 63, 64 and 65; a vector length comparing part 68 for comparing the vector length L established by the part 61 with the vector length l_i computed by the part 66; an angle comparing part 69 for comparing the curve split point threshold angle θ established by the part 62 with the inter-vector angle θ_i computed by the part 67; a straight line storing part 70 and a curve storing part 71 for respectively storing the segments of a straight portion and a curved portion in accordance with the result of comparison performed by the vector length comparing part 68; and a curve split point coordinate storing part 72 for storing the i-th sample point P_i as a curve split when the result of comparison is $\theta > \theta_i$ in the angle comparing part 69.

In the above embodiment, the operation is performed in the following manner. First, a straight portion threshold vector length L and a curve split point threshold angle θ are established in the

parts 61 and 62, respectively. Subsequently, initial point coordinates of one block are transferred from the sample point coordinate storing part 60 to the next sample point coordinate register 63. In this stage, since no coordinate data is stored in the present sample point coordinate register 64, it is impossible to obtain the vector length in the computing part 66 which will be described later. Then the sample point coordinates stored in the next sample point coordinate register 63 are shifted to the present sample point coordinate register 64, and next sample point coordinates are newly stored in the register 63. Thereafter, the coordinates stored in the present sample point coordinate register 64 are shifted to the preceding sample point coordinate register 65, and the coordinates in the next sample point coordinate register 63 are shifted to the present sample point coordinate register 64. And subsequently the next sample point coordinates obtained from the sample point coordinate storing part 60 are stored in the register 63.

The vector length computing part 66 computes a vector length $l_i = \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2}$ from the coordinates (x_{i+1}, y_{i+1}) of the (i + 1)th sample point $P_i + 1$ in the next sample point coordinate register 63 and the coordinates (x_i, y_i) of the i-th sample point P_i in the present sample point coordinate register 64. The vector length l_i thus computed is compared by the vector length comparing part 68 with the threshold length L previously established by the straight-portion threshold vector length establishing part 61. And when $l_i > L$, the segment $[x_i, x_{i+1}]$ is regarded as a straight portion, and the coordinates of the initial and terminal points thereof are stored in the straight line storing part 70. In the case of $l_i \leqq L$, the segment $[x_i, x_i + 1]$ is regarded as a curved portion and then is stored temporarily until discrimination of the next segment. And if the next segment is a straight portion, the coordinates of the initial and terminal points of the preceding segment $[x_i, x_{i+1}]$ regarded as a curved portion are stored in the curve storing part 71. In another case where the next segment is also a curved portion, the discrimination is executed continuously with respect to the following segment and, upon detection of a change from a curved portion to a straight portion, the continuous segments of the preceding curved portion are regarded as a single curved portion, and the coordinates of its initial and terminal points are stored in the curve storing part 71.

[Outline splitting (34)]

Meanwhile, in the inter-vector angle computing part 67, the respective coordinates (x_{i-1}, y_{i-1}), (x_i, y_i), (x_{i+1}, y_{i+1}) of the sample points P_{i-1}, P_i, P_{i+1} are read out from the next sample point coordinate register 63, the present sample point coordinate register 64 and the preceding sample point coordinate register 65, and subsequently the inter-vector angle θ_i (acute angle) shown in Fig. 5 is computed. The angle θ_i thus obtained is compared by the angle comparing part 69 with the curve split point threshold angle θ established previously in the part 62. And when the result of such comparison is $\theta > \theta_i$, the sample point P_i is regarded as a new curve split point, whose coordinates are than stored in the curve split point coordinate storing part 72.

Since many sample points are usually existent in the vicinity of such curve split point, it has been customary that a long time is required for processing the data to determine a desired approximate curve. However, due to division of the outline at the curve split point thus specified, a curve approximation is executed individually for each of segments anterior and posterior to the curve split point, hence expediting the processing to shorten the time required and facilitating the determination of the approximate curve.

Fig. 7 illustrates an exemplary character outline obtained by processing the outline data of Fig. 4 in the aforesaid procedure from the straight line approximation (32) through the outline splitting (34).

In the figure, a mark "o" denotes an initial point and a terminal point of each block; a mark "Δ" denotes a sample point obtained by straight line approximation; and a mark "•" denotes a curve split point. A straight portion is marked with ※, while a curved portion is not marked with any symbol.

[Computation of inclination at outline point (35)]

In approximating the outline shape of the computed curved portion by means of n-degree polynomials (where n = 1, 2, 3), the expressions are determined uniquely when the coordinates of two points and the inclinations thereat are specified. In this case, therefore, it is necessary first to obtain the inclination at each point on the outline.

According to the present invention, the inclination at each of desired outline points is computed by extracting a predetermined number of outline points anterior and posterior to the desired outline point, and finding the respective inclinations of the line segments which connect the desired outline point, whose inclination is to be computed, with the individual extracted points.

The process of computing the inclination at each outline point according to this invention will now be described in detail with reference to Fig. 8.

For computing the inclination t_1 at an initial point P_1 (outline point Q_1) of a block $[P_1, P_n]$ on

the outline illustrated in Fig. 8 (a), first an arbitrary number of outline points existing posterior to the initial point P_1 are extracted, then the respective inclinations of the line segments between the initial point P_1 and the individual extracted points are computed, and the inclination t_1 at the initial point P_1 is computed on the basis of the respective inclinations of the line segments by the expressions which will be described later.

In an exemplary case of finding the inclination with extraction of two outline points, first the inclination m_1 of a line segment $\overline{Q_1 Q_2}$ and the inclination m_2 of a line segment $\overline{Q_1 Q_3}$ are computed. The segment inclination m_1 can be obtained from the coordinates (x_1, y_1) and $x_2, y_2)$ of the two points as

$$m_1 = \frac{y_2 - y_1}{x_2 - x_1}$$

Meanwhile, the inclination m_2 is also obtainable in the same manner.

On the basis of the segment inclinations m_1 and m_2, the inclination t_1 at the outline point Q_1 is determined by the equation

$$t_1 = \tan\left(\frac{\tan^{-1} m_1 + \tan^{-1} m_2}{2}\right)$$

The inclination at the terminal point can be determined in the same manner.

For computing an inclination t_2 at the next outline point Q_2 as shown in Fig. 8 (b), first a predetermined number of outline points existing anterior and posterior to the point Q_2 are extracted, then the respective inclinations of the line segments between the outline point Q2 and the individual extracted points are computed, and the inclination t_2 at the outline point Q_2 is obtained therefrom.

For example, in the case of finding the inclination with extraction of two points relative to the desired outline point, one anterior point and one posterior point are extracted with respect to the outline point Q_2, and the inclination t_2 is determined by the following equation:

$$t_2 = \tan\left(\frac{\tan^{-1} m_1 + \tan^{-1} m_2}{2}\right)$$

In such a case where the number of outline points extractable is less than a designated value, a maximum number of points selectable within the designated range are extracted to compute the inclination.

In the case of computing an inclination t_3 at an outline point Q_3 as shown in Fig. 8 (c), the designated number of outline points are existent posterior and anterior to the point Q3. Therefore, the inclinations m_1, m_2, m_3 and m_4 of the line segments between the outline point Q_3 and the individual points are computed, and the inclination t_3 at the outline point Q_3 is determined by the following equation:

$$t_3 = \tan\left(\frac{\tan^{-1} m_1 + \tan^{-1} m_2 + \tan^{-1} m_3 + \tan^{-1} m_4}{4}\right)$$

Thus, with sequential advance along the outline points, the inclination at each point can be computed in the same procedure as the above.

[Determination of sample points (36)]

Upon completion of computing the inclination at each of the outline points, an approximate curve is defined by two points on the outline, and the deviation between the approximate curve and the outline is computed with respect to each outline point.

Supposing now that the deviation in the approximate curve segment being processed is smaller than an allowable error, the procedure is advanced to the next outline point to repeat the same processing, thereby determining a sample segment where the deviation is the longest within the limit of an allowable error, and thus the entire outline is divided into sample segments. And the outline is approximated according to n-degree polynomials by such sample segments.

Hereinafter the procedure of computing the approximate curve of n-degree polynomials and the deviation will be described with reference to Fig. 9.

First, on the basis of the coordinate values stored in the straight line storing part 70 and the

curve storing part 71, the segment to be approximated is judged whether it is a straight portion or a curved portion. And when the segment is regarded as a straight portion, linear (first degree) expressions representing the segment and the coordinates of its initial point are computed and encoded as will be described later.

When the segment is a curved portion, the following procedure is executed.

In Fig. 9 (a), $S_i(Q_1)$ denotes an initial point of the i-th sample segment to be processed. An approximate curve is computed sequentially with a candidate sample segment defined between the initial point S_i (Q_1) and a candidate sample point S_{i+1} (Q_n). There are further shown an outline point Q_j designated by B; an intersection point C of an approximate curve 91 and a straight line $\overline{BQ}$ 93 which orthogonally traverses a straight line 92 between the two ends of the segment to be approximated with a curve and passes through the point B (Q_j); a deviation $\overline{BA}$ in the x-direction between the outline 90 at the point Q_j and the approximate curve 91; a deviation $\overline{BC}$ between the outline 90 and the approximate curve 91; and a deviation $\overline{BD}$ in the y-direction between the outline 90 and the approximate curve 91. Since the n-degree polynomial in the segment $[Q_1, Q_2]$ determines a straight line uniquely, an approximate curve of n-degree polynomial is obtained from the next segment $[Q_1, Q_3]$.

Hereinafter an example will be described in connection with the case of a cubic curve. An approximate curve f(x) of the above segment can be specified by introducing into the following third-degree equation the initial point S_i (Q_1), the inclinations t_i and t_n of the cancidate sample point Q_n, and the coordinate values (x_i, y_i) and (x_n, y_n) previously computed.

$$f(x) = y_i' + b_i' (x - x_i') + c_i' (x - x_i')^2 + d_i' (x - x_i')^3$$

$$b_i' = t_i'$$

$$c_i' = \frac{\frac{3(y_n - y_i')}{x_n - x_i'} - 2t_i' - t_n}{x_n - x_i'}$$

$$d_i' = \frac{t_i' + t_n - \frac{2(y_n - y_i')}{x_n - x_i'}}{(x_n - x_i')^2}$$

Subsequently to finding the individual coefficients to determine the approximate curve f(x) in the manner mentioned, a deviation ε between the approximate curve f(x) and the outline is computed.

The deviation ε is obtained at each of the outline points Q_j $(2 \leqq j \leqq n-1)$ existing in the said segment.

In the figure, when finding the deviation ε $(= \overline{BC})$ between the outline 90 and the approximate curve 91, the present invention expedites the processing by computing the deviation ε affinely.

Fig. 9(b) is an enlarged view of an error portion between the outline 90 and the approximate curve 91 shown in Fig. 9(a). Supposing now that a line segment $\overline{CA}$ is parallel with a straight line 92, the inclination m of the line segment $\overline{CA}$ is considered to be equal to that of the straight line 92. Then, at the outline point B (Q_j), distances εx and εy from the approximate curve 91 in the x- and y-directions are computed, whereby the desired deviation ε can be obtained approximately as

$$\varepsilon \fallingdotseq \frac{m}{\sqrt{m^2 + 1}} \varepsilon x \qquad (\text{when } |m| > 1)$$

$$\varepsilon \fallingdotseq \frac{1}{\sqrt{m^2 + 1}} \varepsilon y \qquad (\text{when } |m| \leq 1)$$

If the approximate deviation ε between the outline 90 and the approximate curve 91 is smaller than an allowable error, the segment is extended with the next outline point Q_{n+1} determined provisionally as a candidate sample point, and an approximate curve is newly computed with respect to the segment (Q_1, Q_{n+1}) in the same manner as the foregoing. Thus, the deviation ε is examined with respect to each of the entire outline points Q_j existing in the said segment.

And when the deviation has exceeded the allowable error at any outline point, the candidate sample point Q_n immediately anterior thereto is selected as an established sample point S_{i+1}, so that the i-th sample segment [S_i, S_{i+1}] is determined.

Thereafter, by establishing the sample point S_{i+1} as an initial point of the next sample segment, subsequent sample segments are specified with sequential determination of sample points in the same procedure as the above.

In the exemplary case mentioned above, the process is so carried out when the deviation ε has exceeded the allowable error within the next candidate sample segment (Q_1, Q_{n+1}), the candidate sample point Q_n immediately anterior thereto is selected as an established sample point to determine a sample segment.

However, a modified procedure may be executed in such a manner that, when the deviation has exceeded the allowable error, the candidate sample point Q_{n+1} at the moment is stored temporarily, and the deviation ε is further appreciated with respect to a candidate sample segment after several candidate sample points (Q_{n+2}, Q_{n+3}, ...). And if the deviation still exceeds the allowable error even after a predetermined number of points, the said point Q_n is selected as an established sample point to determine the sample segment.

In the above case, when the allowable-error condition is satisfied in the candidate sample segment after several points, this segment is established as a new sample segment [Q_1, Q_n], and the aforementioned appreciation is executed repeatedly while advancing the sample point therefrom. Due to such preliminary recognition in advance, it becomes possible to further reduce the number of sample segments, thereby increasing the data compression rate eventually.

[Recomputation of inclinations at outline points (37)]

When the outline has thus been approximated with third-degree approximate curves in a plurality of sample segments respectively, the initial point coordinates of the sample segments and the degrees of coefficients of the curves to approximate the sample segments are stored as outline data, whereby compressed codes are achievable. However, the inclination obtained in [Computation of inclination at outline point (35)] is the average of those at a predetermined number of points anterior and posterior to the desired outline point, so that there may be a considerable difference from the actual inclination. In view of such circumstances, according to the present invention contrived in order to attain more faithful approximation to the outline, a first-degree derived function f′ (x) of the approximate curve f(x) in each sample segment is formed, and the inclination at each of the outline points in the sample segment is computed again.

[Redetermination of sample points (38)]

On the basis of the inclination at each outline point and the coordinate values thereof obtained by the data processing in [Recomputation of inclinations at outline points (37)], a sample segment is determined while establishing the sample points again in the same manner as in [Determination of sample points 36]. The sample points thus established are approximated in the said sample segment with an approximate curve f(x) which is more faithful to the outline.

[Encoding (39)]

The initial point coordinates of the respective sample segments in the straight and curved portions of the outline and the coefficients and degrees of the approximate curve f(x) are encoded and then stored as a set of block data arranged separately for the individual blocks, thereby providing compressed data faithful to the character outline.

Fig. 10 shows a preferred example of one-block data storage format applied in carrying out the present invention.

In this format, a block header stores the terminal point coordinates of one block and the number of sample segments existing in one block; a segment header stores the x- and y-coordinates of the initial point of one sample segment and the degree of an approximate curve f(x); and segment information stores the coefficient of the approximate curve f(x) specified by the above degree.

One sample segment data is formed of the segment header and the segment information, and sample segment data corresponding to the stored number of sample segments are arrayed sequentially to constitute one block data.

[Storage of compressed data (40)]

The block data encoded in conformity to the above format are stored separately for the individual blocks so as to attain efficient reproduction of the character outline as will be described later, wherein the data storage is effected sequentially in the order of the time length required for decoding the individual block data.

Fig. 11(a) illustrates an exemplary Japanese character "あ" where compressed data have been obtained with respect to individual blocks (number from 1 through 20 for explanation) of the character outline. The data are stored in sequence from the one that requires the longest time (e.g. the block have the longest distance between the initial and terminal points in the x-direction). In Fig. 11(a), block 12 requires the longest decoding time, which is followed by block 10, block 16 and so forth in sequence. And block 3 requires the shortest decoding time. Accordingly, the compressed data of the character outline obtained finally are stored as a set of block data shown in fig. 11(b).

[Dispersive decoding (41)]

The procedure described hereinabove is concerned with compression of the character outline data. Now an explanation will be given on how to reproduce the original outline on the basis of the outline data compressed by the aforementioned data processing.

In order to perform fast reproduction of the outline, the block data to be processed are transferred sequentially to a plurality of decoders, and any decoder having completed the task is assigned to process the next block data. In this arrangement, the time required for decoding the data of each block is a matter of concern. That is, if a long time is needed for decoding the data of one block, it is natural that the processing time for reproduction of the outline is also rendered long. Therefore, in case the block data of a long decoding time is processed late, there occurs a state where the corresponding decoder alone is still kept in operation while all other decoders have already finished the processing. And such undesired state brings about a reduction in the operational efficiency of the decoders.

For the purpose of avoiding occurrence of such a phenomenon, sequential data storage is performed according to this invention in such a manner that the compressed data of the block requiring the longest decoding time is first stored as mentioned above, and the entire block data thus stored are processed by a plurality of decoders, whereby a high processing efficiency is ensured to accomplish fast reproduction of the character outline.

The detailed procedure will be described below with reference to Figs. 12 and 13.

Fig. 12 is a block diagram of an exemplary embodiment contrived to optimally carry out the outline reproduction in the present invention. In the diagram, there are shown a compressed data memory 120 for storing individual block data in the sequence of the time length required for decoding the data of each block; a selector 121 for selecting a decoder and transferring the data of one block thereto; a scale factor memory 122 for storing a desired scale factor inputted from a separate source; a decoder group 123 comprising n-units of decoders for reproducing the outline from the block data at a desired scale factor; a selector 124 for selecting a decoder, which has finished the processing out of the decoder group 123, and transfers the decoded outline image data to a one-character memory therefrom; a one-character memory 125 for storing the outline data fed from the selector 124; and an output unit 126 for printing or displaying the character on the basis of the outline image data of one character completely stored in the one-character memory 125.

The above embodiment performs its operation in the following manner.

The data stored in the compressed data memory 120 are transferred to the selector 121 in a predetermined sequence of the time length required for decoding one block (e.g. in the sequence of block 12, block 10, block 16 and so forth in Fig. 11). Then the selector 121 sequentially selects the decoders out of the decoder group 123 and further transfers the block data to the selected decoders. Upon completion of such data transfer, each decoder starts reproduction of the outline on the basis of the block data thus received and the scale factor data stored separately in the scale factor memory 122. The selector 124 serves to select the decoder having finished the processing for outline reproduction, and transfers the reproduced outline image data to the one-character memory 125. Any decoder having completed the reproduction of one block sends a command to the selector 121 for transfer of block data, whereby the data

of the next one block is transferred thereto.

Fig. 13 is a timing chart representing exemplary processing states of the decoder group 123 shown in Fig. 12. In the chart: there are shown a time T_1 of transferring the block data from the selector 121 to each decoder; a time T_2 of transferring the decoded outline image data via the selector 124 to the one-character memory 125; a time T_3 of decoding the block data in each decoder; and a time T_4 of processing the one-character data.

After the data of n-blocks transferred to the decoders 1 through n begin to be processed, the next (n + 1)th block data is transferred to the decoder having required the shortest processing time (in this example, decoder 3), and the successive data processing is executed continuously.

Since the respective block data to be transferred are arrayed in such a sequence that the required processing time becomes gradually shorter as mentioned already, the individual decoders can terminate processing the data of one character almost simultaneously. It follows that the decoders operate substantially equally without any wasteful idle time to consequently minimize the entire period of time required for decoding the character data.

[Reproduction of outline (42)]

As described above, the outline image data of individual blocks reproduced by the decoders are stored sequentially in the one-character memory 125 to constitute complete outline image data of one character. And the desired character is reproduced as in the prior art by feeding the outline image data from them one-character memory 125 to an output unit 126 such as a laser beam printer, CRT photocomposer or display device.

As a result of experimentally applying the above-described character data processing method of this invention to a Japanese Ming-style hiragana character "あ" composed of 800 × 800 dots, it has been verified that a data compression rate of 1.21% is attainable with one dot determined as an allowable error to a desired character image.

Although in the foregoing description a hiragana character "あ" has been explained as an example, it is obvious that the same processing can be performed with regard to any other character such as Chinese character, alphabetical character or to any of various marks, symbols and line drawings.

Thus, according to the method of this invention mentioned hereinabove, it becomes possible to quickly obtain storage data faithful to the smoothness of a character outline at a sufficiently high compression rate and further to realize fast reproduction of a high-quality character or pictorial image at a desired scale factor.

CLAIMS

1. A method of processing character or pictorial image data to specify the outline of the character of pictorial image developed on x- and y- coordinates so that the encoded data may be subsequently decoded to reproduce the original character or pictorial image, said method comprising: splitting said outline into blocks each defined by a univalent function involving x as a variable: computing, for each of said blocks, a plurality of vectors which extend sequentially along the outline of the character or pictorial image and are so established as to be of the maximum possible length consistent with the requirement that the deviation between each vector and said outline is maintained to be less than a predetermined allowable error; and treating as a straight line any outline portion with a vector longer than a predetermined threshold and encoding said portion by a code adapted for a straight line, while treating as a curve any outline portion with a vector shorter than said predetermined thresold and encoding said portion by segmental polynomial approximation.

2. A data processing method according to claim 1, wherein the polynomial approximation for treating the outline portion as a curve comprises: dividing said portion into a plurality of groups in accordance with the angles of intersection of the mutually adjacent vectors; and carrying out segmental polynomial approximation independently with regard to the outline segments of each group.

3. A data processing method according to claim 1 or 2, said method further comprising: after determination of an outline segment to be encoded by said segmental polynomial approximation, computing inclinations t_1, t_n at the two end points P_1, P_n of said segment and the coordinates (x_1, y_1), (x_n, y_n) of the end points a certain fixed segment which is set in said outline segment; then computing to an n-degree polynomial f(x) (where n = 1, 2, 3) to approximate said segment on the basis of said inclinations t_1, t_n and coordinates; and storing the coefficient and the degree of said n-degree polynomial as outline data to specify said segment.

4. A data processing method according to claim 3, in which the inclinations t_1 and t_n, are obtained by computing the average of the respective inclination at each end point and a predetermined number of outline points preceding and following said end point, and applying said averages, as the inclinations t_1 and t_n.

5. A data processing method according to claim 3, wherein said f(x) is expressed as

$$f(x) = y_i + b_i(x - x_i) + c_i(x - x_i)^2 - d_i(x - x_i)^3$$

and the coefficients thereof are computed from the following equations:

$$b_i = t_i$$

$$c_i = \frac{3\,(y_n - y_i)}{x_n - x_i} - 2t_i - t_n \Big/ x_n - x_i$$

$$d_i = \frac{t_i + t_n - \frac{2(y_n - y_i)}{x_n - x_i}}{(x_n - x_i)^2}$$

6. A data processing method according to claim 1 or 2, said method further comprising: after determination of an outline segment to be encoded by said segmental polynomial approximation, computing an n-degree polynomial $f_1(x)$ (where n = 1, 2, 3) to approximate the shape of an outline segment on the basis of inclinations t_1 and t_n at the two end points of a certain fixed segment which is set in said outline segment and the coordinates thereof; subsequently computing, from the first degree derived function $f'_1(x)$ of said n-degree polynomial $f_1(x)$, the inclination at each of the entire outline points existent within said segment, then computing an n-degree polynomial $f_2(x)$ (where n = 1, 2, 3) to approximate a segment different from the foregoing segment on the basis of said inclination at each of the outline points and said coordinates thereof; and storing the coefficient and the degree of said n-degree polynomial $f_2(x)$ as outline data to specify said segment.

7. A data processing method according to claim 1 or 2, said method further comprising: after determination of an outline segment to be encoded by said segmental polynomial approximation, computing, with respect to each of the outline points within an established segment, an error ε between the outline of a certain fixed said segment which is set in said outline segment and an approximate curve obtained by approximating said outline with an n-degree polynomial; appreciating said error ε by comparison with an allowable value and, if the error ε is smaller than said allowable value, extending said segment for renewal thereof and repeating the computation and appreciation of said error ε until the same exceeds said allowable value; and storing, as outline data to specify the extended segment, the coefficient and the degree of an n-degree polynomial f(x) (wherein n = 1, 2, 3) for said segment.

8. A data processing method according to claim 7, wherein the computation of said error ε is executed by approximation as

$$\varepsilon = \frac{m}{\sqrt{m^2 + 1}}\,\varepsilon_x \quad \text{where } |m| > 1)$$

$$\varepsilon = \frac{1}{\sqrt{m^2 + 1}}\,\varepsilon_y \quad \text{where } |m| \leq 1)$$

(in which m is the inclination of an outline segment; and ε_x and ε_y are deviations between the outline and the approximate curve in the x- and y- directions, respectively).

9. A data processing method according to claim 1 or 2, said method further comprising: splitting said outline into blocks each defined by a univalent function involving x as a variable; then computing encoded block data to specify the individual blocks; subsequently storing the

block data in an individually readable manner in the sequence of the time length required for decoding the data of each block; and after reading out the block data with regard to the individual blocks, distributing said data sequentially to a plurality of decoders so as to decode the same.

10. A method of processing character or pictorial image data substantially as herein described with reference to the accompanying drawings.

Printed in the United Kingdom for
Her Majesty's Stationery Office, Dd 8818935, 1985, 4235.
Published at The Patent Office, 25 Southampton Buildings,
London, WC2A 1AY, from which copies may be obtained.