INTERNATIONAL STANDARD

ISO/IEC 9282-2

First edition
1992-05-15

Information processing — Coded representation of pictures —

Part 2:
Incremental encoding of point lists in a 7-bit or 8-bit environment

Traitement de l'information — Représentation codée de l'image —

Partie 2: Codage en accroissement des listes de points dans un environnement à 7 ou à 8 éléments

Reference number
ISO/IEC 9282-2:1992(E)

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 9282-2 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*.

ISO/IEC 9282 consists of the following parts, under the general title *Information processing — Coded representation of pictures*:

— *Part 1: Encoding principles for picture representation in a 7-bit or 8-bit environment*

— *Part 2: Incremental encoding of point lists in a 7-bit or 8-bit environment*

Annex A forms an integral part of this part of ISO/IEC 9282. Annex B is for information only.

Introduction

This International Standard consists of a series of parts designed to facilitate the interchange and representation of pictures where the interchange is by means of data communications or the exchange of storage media.

ISO 9282 is structured as a multi-part standard. At present, it has two parts :

Part 1: Encoding principles for picture representation in a 7-bit or 8-bit environment.

Part 2: Incremental encoding of point lists in a 7-bit or 8-bit environment.

Other parts may be added later.

Standard methods are given for picture coding in order to assist in coding system design and to prevent a proliferation of different unrelated coding techniques.

Information processing - Coded representation of pictures -

Part 2 :
Incremental encoding of point lists in a 7-bit or 8-bit environment

1 Scope

This part of ISO/IEC 9282 defines

- a set of compression techniques to be used in coding point lists handled by computer graphics applications or by specific applications such as telewriting;

- the data structures to be used to represent the compressed point lists.

A point list specifies a sequence of points, each of them being represented by two coordinates as defined in ISO/IEC 9282-1. A point list contains at least two points but this part of ISO/IEC 9282 is more appropriate when numerous points with short displacements between them are to be coded and, furthermore, when it is required to code the point list in a compact way.

This part of ISO/IEC 9282 does not deal with the presentation semantics of the data; these are defined in the related International Standards.

This part of ISO/IEC 9282 applies either to the encoding of a datastream directly invoked through the identification method of ISO/IEC 9281 or to the subset of a picture which is represented as a point list embedded within a datastream structured in accordance with ISO/IEC 9282-1. Depending on the environment into which the incremental encoding is used, i.e. ISO/IEC 9281 or ISO/IEC 9282-1, different switching mechanisms allow to identify which compression technique is used, they are described in the relevant sub-clauses of this part of ISO/IEC 9282.

It is still possible to code a point list in the displacement mode as defined in ISO/IEC 9282-1 when there is no specific requirement in terms of compression ratio. Any functional standard using ISO/IEC 9282 for the representation of a syntax is free to use either the displacement mode or the incremental mode; ISO/IEC 9282-1 and this part of ISO/IEC 9282 allow for the representation of the displacement mode together with one or several incremental modes.

2 Conformity

A picture coding method is in conformity with this part of ISO/IEC 9282 if the incremental encoding technique used meets the following requirements :

- an incremental mode identification is in conformity with clause 5;

- the incremental mode used is in conformity with one of the modes described in clauses 6,7 or 8.

3 Normative references

The following standards contains provisions which, through reference in this text, constitute provisions of this part of ISO/IEC 9282. At the time of publication, the editions indicated where valid. All standard are subject to revision, and parties to agreements based on this part of ISO/IEC 9282 are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. Members of ISO maintain registers of currently valid International Standards.

ISO/IEC 9281-1 : 1990, *Information technology - Picture coding methods - Part 1 : Identification.*

ISO/IEC 9281-2 : 1990, *Information technology - Picture coding methods - Part 2 : Procedure for registration.*

ISO/IEC 9282-1 : 1989, *Information processing - Coded representation of pictures - Part 1 : Encoding principles for picture representation in a 7-bit or 8-bit environment.*

CCITT T.150, *Telewriting terminal equipment.*

4 Definitions and notation

4.1 Definitions

For the purpose of this part of ISO/IEC 9282, the following definitions apply :

4.1.1 basic grid unit : A binary fraction that identifies the accuracy of coordinates.

4.1.2 bit combination; byte : An ordered set of bits that represents an opcode or an operand or is used as a part of the representation of an opcode or an operand.

4.1.3 code : A set of unambiguous rules that establishes the one to one relationship between each opcode or operand of the set and their coded representation by one or more bit combinations.

4.1.4 differential chain mode : A coding method used in incremental mode, identifying differences between steps (increments).

4.1.5 domain ring : A mechanism for defining the accuracy (number of bits) of the coordinate data.

4.1.6 nibble : An ordered set of bits that represents a step increment in the step increment coding.

4.1.7 opcode : A one or multi-byte representation that identifies a function required by a picture or a computer graphics interface. An opcode may be followed by zero or more operands.

4.1.8 operand : A single or multiple coded representation used to specify the parameters required by an opcode.

4.1.9 pixel, picture element : The smallest element of a display surface can be independently assigned a colour or intensity.

4.1.10 ring : A square defined by its radius and angular resolution factor, used for encoding increments in the incremental mode.

4.2 Notation

4.2.1 7-bit byte

The bits of a 7-bit byte are identified by b_7, b_6, b_5, b_4, b_3, b_2 and b_1 where b_7, is highest order, or most significant bit and b_1 is the lowest order, or least significant bit.

The bit combinations are identified by notations of the form x/y, where "x" is a number in the range 0 to 7 and "y" a number in the range 0 to 15.

The correspondence between the notations of the forms "x/y" and the bit combinations consisting of the bits b_7 to b_1, is as follows :

- x is the number represented by b_7, b_6 and b_5 where these bits are given the weights 4, 2 and 1, respectively.

- y is the number represented by b_4, b_3, b_2 and b_1 where these bits are given the weights 8, 4, 2 and 1, respectively.

4.2.2 8-bit byte

The bits of an 8-bit byte are identified by b_8, b_7, b_6, b_5, b_4, b_3, b_2 and b_1, where b_8 is the highest-order, or most-significant bit and b_1 is the lowest-order, or least-significant bit.

In this part of ISO/IEC 9282, the bit combinations are identified by notations of the form xx/yy, where xx and yy are numbers in the range 00 to 15. The correspondence between the notations of the form xx/yy and the bit combinations consisting of the bits b_8 to b_1, is as follows :

- xx is the number represented by b_8, b_7, b_6 and b_5, where these bits are given the weights 8, 4, 2 and 1, respectively ;

- yy is the number represented by b_4, b_3, b_2 and b_1 where bits are given the weights 8, 4, 2 and 1, respectively.

4.2.3 Byte interpretation

Bits within a byte may be interpreted to represent numbers in binary notation by attributing the following weights to the individual bits :

Bits of a 7-bit code		b_7	b_6	b_5	b_4	b_3	b_2	b_1
Bits of a 8-bit code	b_8	b_7	b_6	b_5	b_4	b_3	b_2	b_1
Weight	128	64	32	16	8	4	2	1

Using these weights, the bit-combinations of the 7-bit code represent numbers in the range 0 to 127 and the bit-combinations of the 8-bit code represent numbers in the range 0 to 255.

4.3 State variables

State variables are used to control the encoding of a datastream. Such state variables are set to a value by an application at the time a coding/decoding process is initialized.

Depending on the incremental method under which the encoding defined in this part of ISO/IEC 9282 is used, a state variable may remain fixed or may dynamically be modified within the encoded datastream.

The overall list of state variables defined in ISO/IEC 9282-2 is given in annex A.

5 Incremental encoding structure

The incremental coding modes allow for the coding of point lists in a compact manner. This part of ISO/IEC 9282 specifies three incremental coding modes :

a) the STEP INCREMENT CODING (SIC) which makes assumption on the smoothness of boundaries of natural objects.

b) the DIFFERENTIAL CHAIN CODING (DCC) which codes incremental displacements on a ring around successive points.

c) the ZONE CODING (ZC) which reduces redundancy of handwritten drawings by simulating the sampling of points at a fixed rate.

Specifications of the incremental coding modes are described in the clauses hereafter.

This part of ISO/IEC 9282 does not intend to make comparisons between the different incremental modes in terms of compression ratio, coding/decoding process and types of applications into which each of them is the most appropriate.

According to the definitions given in ISO/IEC 9282-1 a point list is coded as an operand; it is then preceded either by an opcode or by an operand in a datastream.

5.1 Incremental mode identification

The incremental identification is dependent on the environment into which incremental data are to be used.

If an incremental mode is directly invoked through ISO/IEC 9281-1, the incremental mode is identified by a coding method identifier (CMI); CMI's will be assigned by the registration authority for picture coding methods when they are required. Figure 1 shows the incremental coding structure used with ISO/IEC 9281-1.

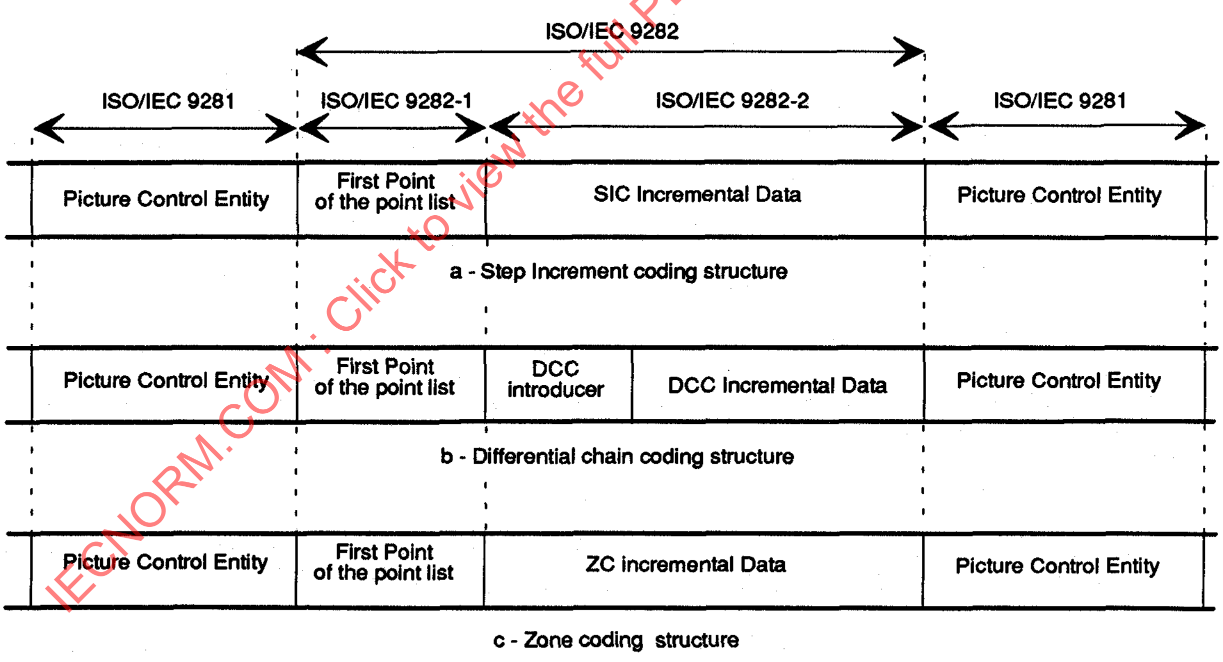


Figure 1 - Incremental coding structure used with ISO/IEC 9281

NOTE - The DCC introducer is not really necessary for the purpose of ISO/IEC 9282-2, it is maintained for compatibility with CCITT recommendation T.150 on telewriting terminal equipment.

If an incremental mode is invoked through ISO/IEC 9282-1, an incremental coding announcer is used within a point list to announce that the points which follow are coded in incremental mode; furthermore an incremental coding identifier identifies which incremental mode is used. Figure 2 shows the incremental coding structure used with ISO/IEC 9282-1.

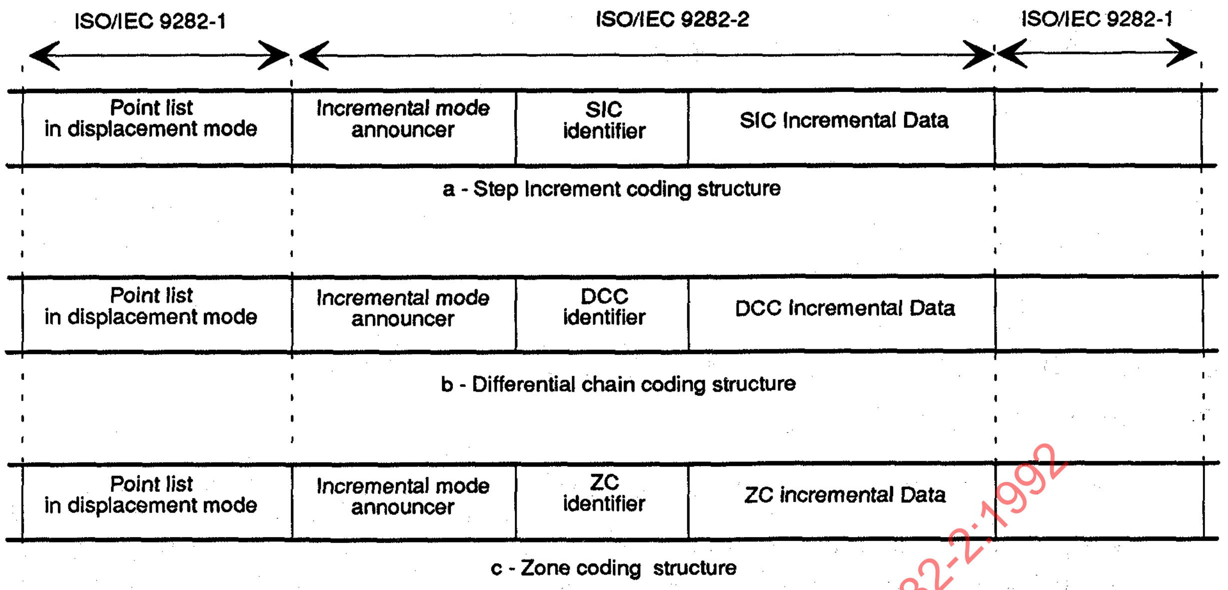


Figure 2 - Incremental coding structure used with ISO/IEC 9282-1

The point list in the displacement mode should, at least, include one point, the first point of the point list.

5.2 Encoding of the first point of a point list

The first point of a point list or the sequence of points which precede the incremental data are coded in the displacement mode as described in 6.3.6 of ISO/IEC 9282-1.

5.3 Encoding of the incremental mode introducer

An incremental mode introducer is composed of two elements :

a) An INCREMENTAL MODE ANNOUNCER which indicates that the sequence of points which follows in the point list is coded in one of the incremental modes.

b) An INCREMENTAL MODE IDENTIFIER which specifies what incremental mode is used.

The value of the INCREMENTAL MODE ANNOUNCER is conventionally MINUS ZERO, according to the description made in ISO/IEC 9282-1, but its encoding is dependent on the format into which coordinates are coded :

1) If coordinates are coded with real numbers using the basic format : the INCREMENTAL MODE ANNOUNCER is a signed integer coded in the basic format on a single byte with the value MINUS ZERO.

2) If coordinates are coded with real numbers using the bitstream format : the INCREMENTAL MODE ANNOUNCER is a signed integer coded in the bitstream format with the value MINUS ZERO. It is represented on one or several bytes according to the SIGNED INTEGER LENGTH state variable as defined in ISO/IEC 9282-1.

3) If coordinates are coded using interleaved coordinate pairs : the first coordinate pair is coded as a sequence of two coordinates with the value MINUS ZERO. It is represented on one or several bytes according to the INTERLEAVED LENGTH state variable value as defined in ISO/IEC 9282-1.

NOTE - In the two's complement notation, the value zero has a positive sign and MINUS ZERO does not exist. For the purpose of that part of ISO/IEC 9282 MINUS ZERO shall be represented by a sequence of bits, the most significant bit being "1" or sign bit, followed by all "0's" bits.

The INCREMENTAL MODE IDENTIFIER is coded as an unsigned integer according to ISO/IEC 9282-1, it can take one of the following values :

0 : Step Increment Coding

1 : Differential Chain Coding

2 : Zone Coding

>2 : Reserved

6 Step Increment Coding mode description

6.1 Step Increment Coding outline

The step Increment Coding provides a method of coding short point displacements which may be used to describe natural objects. An assumption is made that the boundary of a natural object has a certain degree of smoothness so that it is less frequently required to radically change directions. Increments in the same direction of displacement or with only a small angular displacement can be efficiently coded but major changes of direction require a somewhat longer encoding.

In this mode, points lie at the line intersection of a grid, the cell of which are sized dx and dy along the x and y directions, respectively, the step size for increments (dx, dy) is assigned globally for a string of incremental codes. The first point of the point list is coded in the DISPLACEMENT MODE from the origin then the successive points are defined in terms of displacement directions from the previous point. Eight directions are coded, they are given in figure 3 :

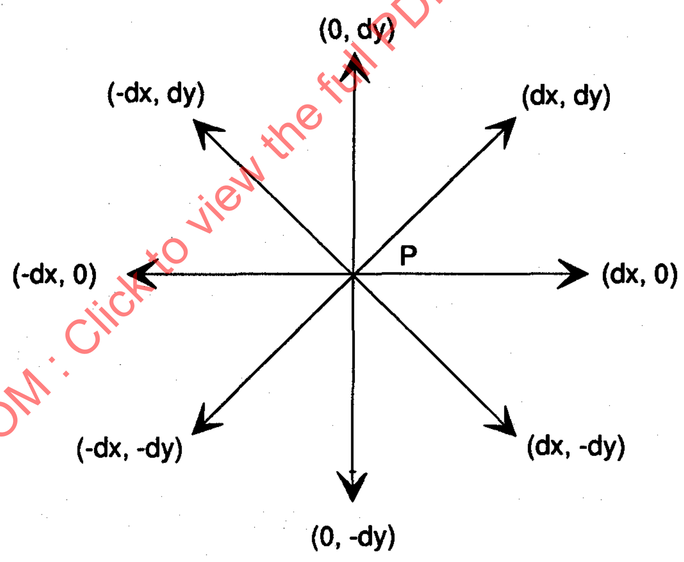


Figure 3 - Displacement directions for step increment code

The step size for increments dx and dy represent distances in the coordinate space; it is coded as a pair of coordinates. Step direction may be varied within strings of incremental codes.

6.2 Step Increment Coding encoding

The step size is the first part of the incremental data; it is mandatory and coded as a pair of coordinates in the interleaved format in conformity with ISO/IEC 9282-1. Both coordinates are always positive.

The step size is followed by a sequence of step increments. Each increment is encoded as a two bit nibble, with three nibbles packed within one byte. These three nibbles occupy the lower six bits of a byte in accordance with the bitstream format described in ISO/IEC 9282-1. This is illustrated in figure 4 :

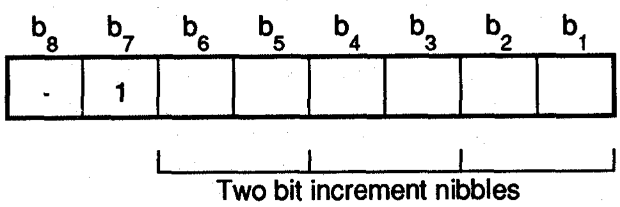


Figure 4 - Step code bit assignment

The interpretation of the two bit nibbles is as shown in table 1 :

Table 1 - Nibble value assignment

NIBBLE VALUE		INTERPRETATION
0	0	Interpret the following nibble as modify parameter instruction
0	1	Advance the point a distance dx in the x direction
1	0	Advance the point a distance dy in the y direction
1	1	Advance the point a distance dx in the x direction and dy in the y direction

When a nibble value of (0,0) is encountered, the next nibble is interpreted as a modify parameter instruction as shown in table 2.

The nibble which follows the (0,0) value is interpreted as a step operation.

Table 2 - Modification instruction

NIBBLE VALUE		MODIFY PARAMETER INSTRUCTION
0	0	Incremental data terminator
0	1	Change sign of dx
1	0	Change sign of dy
1	1	Change sign of dx and dy

If required, the last byte of incremental data is stuffed with "0" bits.

7 Differential Chain Coding mode description

7.1 Differential Chain Coding outline

The data in this mode do not reflect actual coordinates, but identify points on a Ring. A Ring is a set of points on a square which centre is the previously identified point. The first centre point is encoded in displacement mode according to ISO/IEC 9282-1.

A Ring is characterized by its radius (*R* in Basic Grid Units), it Angular Resolution (by a factor *p*) and its Direction (D). The maximum number of points on a Ring is 8*R*. The actual number of points on a Ring with a given Angular Resolution factor *p* follows from :

$$N = 8R / 2^{p} \text{ with } p = 0, 1, 2, 3$$

N must be even. If *N* is odd, the encoded operand (the point list) must be discarded. If *N* is even for the first part of the operand and *N* is odd for the remaining part, the remaining part (with *N* being odd) is discarded.

The points on the Ring are numbered, starting at the Direction point, form 0 to *M* -1 for the upper part of the Ring and from -1 to -*M* for the lower part of the Ring, with *M* = *N* / 2.

Figure 5 shows a Ring with radius *R* = 3 and angular resolution factor *p* = 0, respectively 1.

Figure 5 - Some examples of rings with point numbering

The Direction of a Ring is identified by the position of the point with number ZERO. The initial position of this point is on the positive x-axis, while the Cartesian axes are drawn through the centre point of the Ring. The Direction of the Rings following the initial one is dependent on the Direction of the increments. This Direction is determined in the following way :

If P1 is the previous centre point and the current centre point is P2 (P2 is a point on the Ring with the centre in P1), the position of the point with number ZERO on the Ring, with P2 as centre point, is opposite to point P1, this is the Direction of the Ring. So the Direction of the Ring is dependent on the writing Direction as indicated by the last increment. The position of the increment on the new Ring (centre P2) is described as the difference between the position of point P2 on the previous Ring and position of the new point P3 on the current Ring.

In the DCC only the difference between points on the consecutive Rings are coded. Or to state it in another way, the Direction of the Ring is dependent on the Direction of the line to be displayed. As shown in figure 6, the position of point P3 is defined by the difference :

$$P3 - P2 = -1$$

P3 and P2 being point numbers on the two Rings numbered as given in figure 5.

The Direction (position of the point with number ZERO) is identified by D.

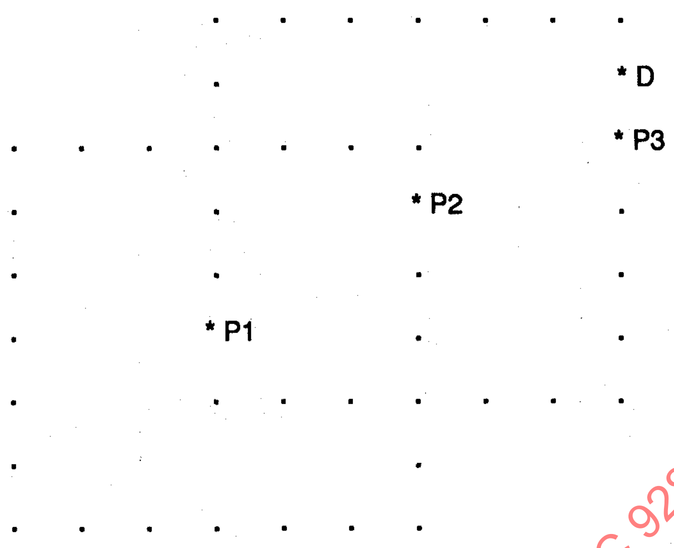


Figure 6 - Change of Direction with *R* = 3

The basic radius of the Ring, as used in the incremental mode, is dependent on the granularity code as set with the COORDINATE GRANULARITY CODE state variable.

The basic radius follows from :

$$\text{basic radius} = 2^{(-8-\text{granularity code})}$$

The basic radius is expressed in Basic Grid Units (BGUs).

The basic radius must be equal to or greater than ONE. If the basic radius is less than ONE, the value ONE is assumed for the basic radius.

The basic radius, as derived from the granularity code, may be changed to any value by a primitive, which defines the Ring size. The angular resolution factor may be changed by a primitive, which defines the angular resolution factor.

The default value for the radius is the basic radius. The default value for *p* is 0 and *p* can only be 0, 1, 2 or 3.

The encoding used in incremental mode makes use of the DCC property by using variable length code-words (Huffman Code). The encoding also allows changing of the radius and the Angular Resolution factor. The radius can have a value of *R*, 2*R*, 4*R* or 8*R*, where *R* is the defined radius. The angular resolution factor *p* can be 0, 1, 2 or 3.

The Huffman Code table used in the incremental mode is a fixed length table. To allow the encoding of more points on a Ring two Escape codes are defined. With these Escape codes the points outside the Huffman Code table can be addressed. The end of incremental mode data is indicated by an End of Block value in the Huffman Code table.

The Huffman code is given in table 3.

Table 3 - Huffman Code table for incremental mode

Code no	Length	Code-word	Point number
1	2	00	0
2	2	10	1
3	2	01	-1
4	4	1100	2
5	4	1101	-2
6	6	111000	3
7	6	111001	-3
8	6	111010	4
9	6	111011	-4
10	8	11110000	5
11	8	11110001	-5
12	8	11110010	6
13	8	11110011	-6
14	8	11110100	7
15	8	11110101	-7
16	8	11110110	8
17	8	11110111	-8
18	10	1111100000	9
19	10	1111100001	-9
20	10	1111100010	10
21	10	1111100011	-10
22	10	1111100100	11
23	10	1111100101	-11
24	10	1111100110	12
25	10	1111100111	-12
26	10	1111101000	13
27	10	1111101001	-13
28	10	1111101010	14
29	10	1111101011	-14
30	10	1111101100	15
31	10	1111101101	-15
32	10	1111101110	16
33	10	1111101111	-16
34	10	1111110000	17
35	10	1111110001	-17
36	10	1111110010	18
37	10	1111110011	-18
38	10	1111110100	19
39	10	1111110101	-19
40	10	1111110110	C1
41	10	1111110111	-20
42	10	1111111000	C2
43	10	1111111001	C3
44	10	1111111010	C4
45	10	1111111011	C5
46	10	1111111100	C6
47	10	1111111101	IM-ESC 1
48	10	1111111110	IM-ESC 2
49	10	1111111111	End of block

The (End of Block) code from the Huffman Code table identifies the end of the incremental mode data. Remaining bits in the last incremental mode data byte have no meaning; they will be ignored.

The Incremental Mode escape codes <IM-ESC 1> and <IM-ESC 2> are used to extend the addressable number of points, e.g. points outside the range - 20 to 19. The code <IM-ESC 1> adds + 20 or - 20 to the following code depending on the sign of that following point. The code <IM-ESC 2> adds + 40 or - 40 to the following code, depending on the sign of the following point. The escape codes can follow each other in any desired order. The following examples demonstrate some possible combinations, [n] is a point number.

<IM-ESC 1>[1] = point number 21
<IM-ESC 1>[-1] = point number-21
<IM-ESC 2>[14] = point number 54
<IM-ESC 2>[-12] = point number-52
<IM-ESC 1> <IM-ESC2>[6] = point number (20 + 40 + 6) = 66
<IM-ESC 2> <IM-ESC1>[-18] = point number (-40 -20 -18) = -78

The codes C1 up to C6 are used to change the parameters that define the Ring to be used. The values of R are taken from the range R_0, $2R_0$, $4R_0$ and $8R_0$, where R_0 is the value of the Ring radius before entering the incremental mode. The values of p are taken from the range 0, 1, 2 and 3. The functions of these codes is as follows :

a. C1

Change the Ring parameters, R and p, to the next higher value e.g. if the radius is R_0, the next higher is $2R_0$, if $p = 0$, the next higher value is 1. R cannot become greater than $8R_0$ and p cannot become greater than 3. For example if the current Ring radius is $8R_0$ or the current $p = 3$, the code <C1> has no effect.

b. C2

Change the Ring parameters, R and p, to the next lower value. The effect of the code <C2> is the inverse of code <C1>. R cannot become smaller than R_0 and p cannot become smaller than 0. For example if the current radius is R_0 or the current p is 0, the code <C2> has no effect.

c. C3

Change the Ring radius R to the next higher value. The code <C3> has no effect if the current radius = $8R_0$.

d. C4

Change the Angular Resolution factor p to the next higher value. The code <C4> has no effect if the current $p = 3$.

e. C5

Change the Ring radius R to the next lower value. The code <C5> has no effect if the current radius = R_0.

f. C6

Change the Angular Resolution factor p to the next lower value. The code <C6> has no effect if the current $p = 0$.

In addition, those codes (C1 to C6) set the position of the point with number ZERO on the positive x-axis while the Cartesian axes are drawn through the centre point of the Ring.

7.2 Differential Chain Coding encoding

The incremental data are coded in the Bitstream format as defined in ISO/IEC 9282-1.

The incremental mode uses variable length code-words. The code-words are packed in consecutive bits of the incremental mode bytes, starting from the higher numbered bit to the lower numbered bit. Code-words are not aligned on byte boundaries; a code-word begins at the first data bit (bit b_6 to b_1 of each byte) which follows the last bit of the preceding code-word. If the code-word does not fit in one byte, the most significant part is packed in the first byte, the remaining part is packed in the second byte and so on.

The end of incremental mode data is identified by the End of Block code. Remaining bits in the last incremental mode data byte have no meaning, they will be ignored.

8 Zone Coding mode description

8.1 Zone Coding outline

This coding method reduces redundancy of handwritten drawings. In this method a trace is coded as a sequence of vectors, the beginning of a trace is the starting point of the first vector ; the end point of a vector constitutes the starting point for the next vector.

The location of the present point is defined by a single valued function of previously coded point.

The end point position is found through a three step approximation :

- the quadrant number difference dq;
- the zone description number difference dk;
- the relative address within the zone (Ax, Ay).

A vector D_i is defined by :

$$D_i = P_i - P_{i-1}$$

$$= (dx_i, dy_i)$$

$$= (x_i - x_{i-1}, y_i - y_{i-1})$$

Where P_i is the i - th coordinate pair to be coded and P_{i-1} the previous coordinate pair.

The quadrant number q_i of the i-th vector is defined as :

$q_i = 1$ for $dx \geq 0$, $dy \geq 0$

$q_i = 2$ for $dx < 0$, $dy \geq 0$

$q_i = 3$ for $dx < 0$, $dy < 0$

$q_i = 4$ for $dx \geq 0$, $dy < 0$

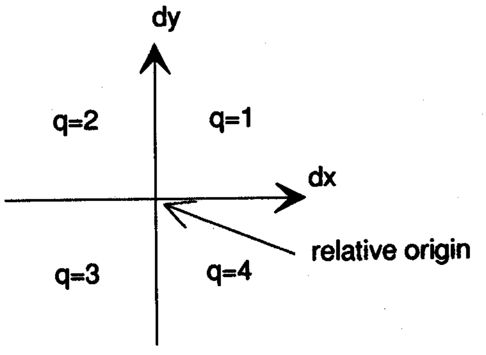


Figure 7 - Definition of quadrant number

The quadrant number difference dq_i is defined as :

$$dq_i = q_i - q_{i-1}, \text{ where } q_0 = 1, \text{ for simplicity.}$$

The space of vectors without signs is divided into square zones which are numbered counter clockwise, as shown in figure 8 :

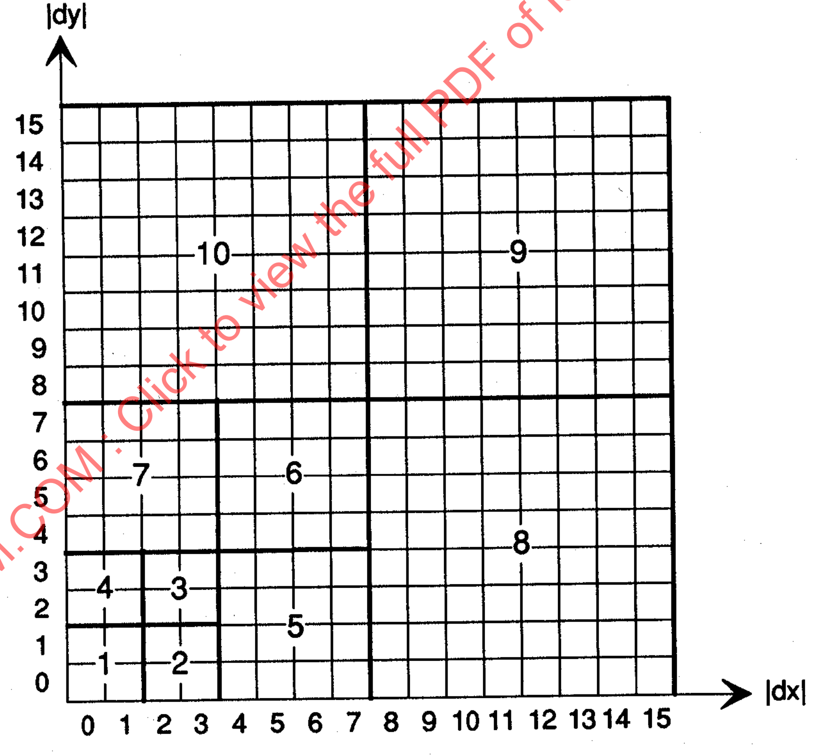


Figure 8 - Zone division and zone designation number

The zone width is taken as the power of two. Thus the width of the k-th zone is defined as :

$W(k) = 2$ for $k = 1$

$W(k) = 2 \times 2^{[(k-2)/3]}$ for $k > 1$, where [ ] is the Gaussian integer notation.

The k-th zone Z_k is defined as :

a) for $k = 1$:

$$Z_k = [\ |dx|\ ,\ |dy|\ ]\ ;\ 0 \le |dx| \le W(k)\text{-}1,\ 0 \le |dy| \le W(k)\text{-}1$$

b) for $k > 1$:

1) for $k = 0$ (mod 3)

$$Z_k = [|dx|\ ,\ |dy|\ ]\ ;\ W(k) \le |dx| \le 2\ W(k)\text{-}1,\ W(k) \le |dy| \le 2W(k)\text{-}1$$

2) for $k = 1$ (mod 3)

$$Z_k = [|dx|\ ,\ |dy|]\ ;\ 0 \le |dx| \le W(k)\text{-}1,\ W(k) \le |dy| \le 2W(k)\text{-}1$$

3) for $k = 2$ (mod 3)

$$Z_k = [|dx|\ ,\ |dy|\ ]\ ;\ W(k) \le |dx| \le 2\ W(k)\text{-}1,\ 0 \le |dy| \le W(k)\text{-}1$$

The zone number difference dk_i is defined as :

$$dk_i = k_i - k_{i\text{-}1}$$

Where k_i is the zone number obtained by i-th vector, and $k_0 = 1$, for simplicity.

The origin of the relative address in each zone is the lower left corner. The relative address in the k-th zone, (Ax, Ay) is defined as :

a) For $k = 1$:

Ax = dx, Ay = dy

b) For $k > 1$:

1) for $k = 0$ (mod 3) :

Ax = |dx| - $W(k)$, Ay = |dy| - $W(k)$

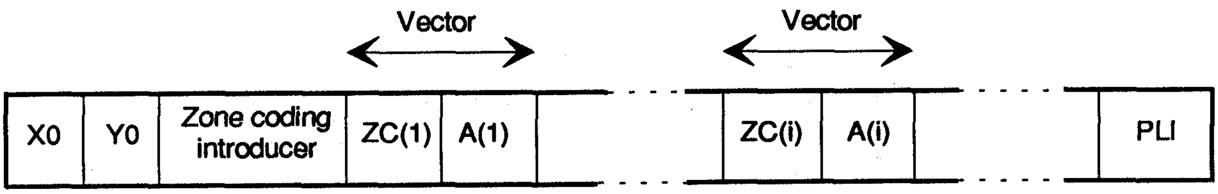
2) for $k = 1$ (mod 3) :

Ax = |dx|, Ay = |dy| - $W(k)$

3) for $k = 2$ (mod 3) :

Ax = |dx| - $W(k)$, Ay = |dy|

All the successive points are represented by zone codes (ZC) and relative addresses (Ax, Ay) ; the full data format of a stroke is illustrated in figure 9.



X0, Y0	:	Starting point address
ZC (i)	:	Zone code of the i-th vector
A (i)	:	Relative address of the i-th vector
PLI	:	Pen Lift indicator (end of incremental data)

Figure 9 - Incremental Zone Coding format

The zone code is defined in table 4.

The table specifies a zone code number 1 ... 30 and a bit combination for 30 combinations of dq and dk.

Table 4 - Zone code table

Zone code no	dq	dk	Length of the code (bit)	Code (the left bit is LSB)
1	0	0	2	01
2	3	0	4	00 01
3	1	0	4	11 11
4	0	3	4	00 10
5	0	1	4	10 11
6	0	-3	4	11 10
7	3	3	5	10 01 1
8	0	-1	5	00 11 1
9	3	-1	6	10 01 01
10	3	-3	6	10 00 01
11	2	0	6	00 11 01
12	1	3	6	10 10 01
13	1	1	6	10 00 11
14	1	-3	6	10 10 11
15	0	4	6	10 00 10
16	0	2	6	00 00 11
17	0	-2	6	00 00 01
18	3	2	7	10 00 00 1
19	3	1	7	10 01 00 1
20	2	3	7	10 10 10 0
21	1	2	7	10 10 00 1
22	1	-1	7	00 11 00 1
23	1	-2	7	10 01 00 0
24	0	6	7	00 00 00 1
25	0	-4	7	00 11 00 0
26	0	-6	7	10 10 00 0
27	3	6	8	10 10 10 10
28	2	1	8	10 00 00 01
29	2	-1	8	10 10 10 11
30	2	-3	8	00 00 00 01
PLI			3	11 0
EFZ			6	00 00 10
NULL			8	00 00 00 00

Legend :
PLI : Pen Lift indicator
EFZ : Escape from zone code

A vector end point position of which the combination dq and dk is not defined in table 4 is coded EFZ (Escape From Zone Code) followed by an absolute coordinate pair.

The end of a trace is indicated by PLI (Pen Lift Indicator) following the last (relative or absolute) address.

8.2 Encoding of the Zone Coding Incremental mode

The zone coding incremental uses one state variable, the GRID RESOLUTION that defines the number of steps into which the unit screen is divided; dx and dy are obtained by dividing the width and the height of the unit screen by the GRID RESOLUTION, respectively.

All successive points are represented by their respective zone code (ZC) and by their relative address (Ax,Ay) in the zone.

A zero vector (0,0) is not coded ; it is also possible that the zone vector ($| x_i - x_{i-1} | \leq 1, | y_i - y_{i-1} | \leq 1$) is rejected before being coded.

The zone code is coded as a variable length word, immediately followed by a relative address (Ax, Ay) ; The bit length L of each relative address component is defined by :

$L = 2 \log_2 W(k)$, where $W(k)$ stands for the width of the zone which contains the point to be coded.

The first point (xo, yo) is coded as a pair of coordinates in accordance with the description of ISO/IEC 9282-1.

ANNEX A

(normative)

List and definition of state variables

A.1 State variables used for the Differential Chain Coding

ANGULAR RESOLUTION FACTOR : Defines the actual number of points on a Ring through the relation :

$$N = 8R / 2^p \text{ with } p = 0, 1, 2, 3$$

With N = number of points on a Ring
R = radius expressed in basic grid units (BGU)
p = angular resolution factor

COORDINATE GRANULARITY CODE : A signed integer which specifies the smallest non zero value that can be expressed. It is the smallest exponent that is permitted in the coded representation of a coordinate or size value. This value is called the basic grid unit (BGU)

COORDINATE MAGNITUDE CODE : Specifies the largest possible magnitude of coordinates and size values that can be encoded.

RADIUS : An unsigned integer which specifies the Ring size in basic grid units.

A.2 State variables used for the Zone Coding incremental mode

GRID RESOLUTION : All coordinates of incremental data are quantized relative to a measurement grid in the unit area.

The resolution of this grid (GRID RESOLUTION) is determined by the expression :

$$\text{GRID RESOLUTION} = 2^n \times 2^n$$

with n = GRID RESOLUTION FACTOR

GRID RESOLUTION FACTOR : The GRID RESOLUTION FACTOR is an unsigned integer which defines the GRID RESOLUTION as follows :

n = 6 means : GRID RESOLUTION = 64 x 64
n = 7 means : GRID RESOLUTION = 128 x 128

The minimum value of n is 6 and the maximum 11.