PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 : G06K 9/00

A1

(11) International Publication Number: WO 91/10207

(43) International Publication Date: 11 July 1991 (11.07.91)

(21) International Application Number: PCT/US91/00072

(22) International Filing Date: 3 January 1991 (03.01.91)

(30) Priority data:
460,755 4 January 1990 (04.01.90) US

(71) Applicant: TEKNEKRON COMMUNICATIONS SYSTEMS, INC. [US/US]; 2121 Allston Way, Berkeley, CA 94704 (US).

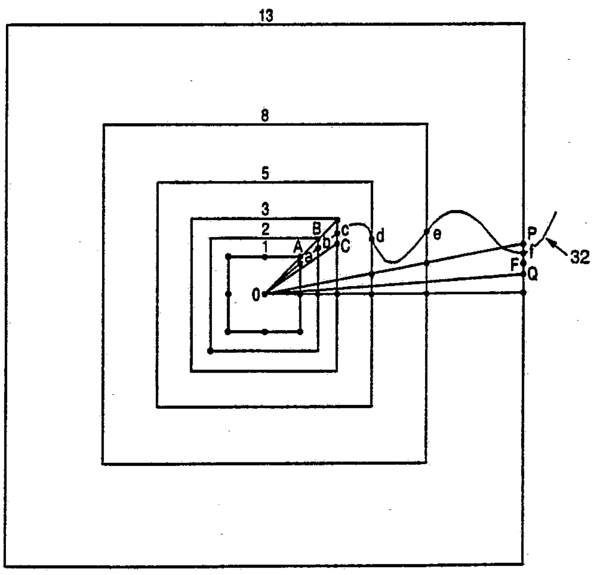
(72) Inventors: BERGER, Toby ; 422 Highland Road, Ithaca, NY 14850 (US). MILLER, Daniel, H. ; 207 Stanford Avenue, Kensington, CA 94708 (US).

(74) Agents: YIN, Ronald, L. et al.; Limbach, Limbach & Sutton, 2001 Ferry Building, San Francisco, CA 94111 (US).

(81) Designated States: AT (European patent), AU, BE (European patent), CA, CH (European patent), DE (European patent), DK (European patent), ES (European patent), FR (European patent), GB (European patent), GR (European patent), IT (European patent), JP, LU (European patent), NL (European patent), SE (European patent).

Published
With international search report.
With amended claims.

(54) Title: A METHOD AND AN APPARATUS FOR ELECTRONICALLY PROCESSING A TRANSACTION WITH A HUMAN SIGNATURE



(57) Abstract

In the present invention a method of compressing a signature signal is disclosed. The signature signal (32) is divided into a plurality of signature segment signals where each segment is encoded by using a modified ring-encoding method, such that the total number of grid points along the perimeter of all the rings can be stored in an 8-bit byte. In one embodiment of the method of the present invention, a Fibonacci series (1, 2, 3, 5, 8, 13) is used to determine the relative spacing of the rings. The present invention also discloses a method and apparatus for electronically processing a POS transaction with a human signature for verification of the transaction, as well as for request of extension of credit by a credit company. The signature (32) is electronically captured, compressed and combined with a transaction signal which is representative of the transaction. The record signal is then processed by the credit company for verification of the transaction or for requesting extension of credit.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	**ES**	Spain	**MG**	Madagascar
AU	Australia	**FI**	Finland	**ML**	Mali
BB	Barbados	**FR**	France	**MN**	Mongolia
BE	Belgium	**GA**	Gabon	**MR**	Mauritania
BF	Burkina Faso	**GB**	United Kingdom	**MW**	Malawi
BG	Bulgaria	**GN**	Guinea	**NL**	Netherlands
BJ	Benin	**GR**	Greece	**NO**	Norway
BR	Brazil	**HU**	Hungary	**PL**	Poland
CA	Canada	**IT**	Italy	**RO**	Romania
CF	Central African Republic	**JP**	Japan	**SD**	Sudan
CG	Congo	**KP**	Democratic People's Republic of Korea	**SE**	Sweden
CH	Switzerland			**SN**	Senegal
CI	Côte d'Ivoire	**KR**	Republic of Korea	**SU**	Soviet Union
CM	Cameroon	**LI**	Liechtenstein	**TD**	Chad
CS	Czechoslovakia	**LK**	Sri Lanka	**TG**	Togo
DE	Germany	**LU**	Luxembourg	**US**	United States of America
DK	Denmark	**MC**	Monaco		

A METHOD AND AN APPARATUS FOR ELECTRONICALLY PROCESSING A TRANSACTION WITH A HUMAN SIGNATURE

Technical Field

The present invention relates to a method and an apparatus for electronically processing a transaction having a human signature for verification of the transaction. More particularly, the present invention relates to a method and an apparatus whereby the human signature is transformed into an electronic signal which is then compressed. The compressed signature signal is combined with an electronic signal representative of the transaction to form a record signal.

Background Of The Invention

Electronic recording of Point Of Sale (POS) transactions is well-known in the art. However, POS transactions of the type involving a human signature, such as those requesting extension of credit, have required a large amount of storage to capture the human signature in electronic form. The large amount of storage required for a signature signal is acceptable in certain industries where the transaction volume is low. For example, in the package delivery industry where a human signature is recorded along with the delivery of a package, the human signature is converted into an electronic signal which is recorded and stored.

However, in POS transactions involving requests for credit extension where a large number of transactions are involved, the storage requirement for record signals, each comprised of a signature signal and a transaction signal, becomes enormous.

Modified ring encoding is one of the techniques that has been used to encode weather maps, contour maps and other pictorial line-drawings. See, for example, "A Rate And Distortion Analysis Of Chain Codes For Line Drawings" by David L. Neuhoff and Kenneth G. Castor, IEEE Transactions On Information Theory, Vol. IT-31, No. 1, January 1985. However, the application of the modified ring-encoding technique to signature compression in a manner to store signature signals in a well-defined bit field has not been known heretofore.

Summary Of The Invention

In the present invention, a method for electronically processing a transaction having a human signature for verification of the transaction is disclosed. The method comprises the steps of electronically capturing the transaction data at the point of transaction to form a transaction signal. The human signature is electronically captured at the point of transaction to form a signature signal. The signature signal is compressed to form a compressed signal signature. The compressed signal signature is combined with the transaction signal to form a record signal. The record signal is processed for verification of the transaction.

An apparatus to carrying out the foregoing method is also disclosed.

Brief Description Of The Drawings

Fig. 1 is a perspective view of an apparatus of the present invention.

Fig. 2 is a graphical image of one embodiment of the encoding technique of the present invention, showing the encoding of a portion of a human signature.

Fig. 3 is a schematic diagram showing the encoding of segments of a human signature.

Fig. 4 is a graphical image of another embodiment of the encoding technique of the present invention, showing the encoding of a portion of a human signature.

Fig. 5 is a schematic diagram of the conditional coding of the segments of a signature in accordance with the method of the present invention.

Fig. 6 is a schematic drawing showing encoding of a signature in accordance with two different methods and the method of the present invention for selecting and encoding the results of these two different methods.

Fig. 7 is a schematic drawing showing the encoding of the segments of a human signature, along with their associated timing marks.

Detailed Description Of The Drawings

Referring to Fig. 1, there is shown a perspective view of an apparatus 10 of the present invention. The apparatus 10 comprises a machine 14, such as a cash register, to record a Point Of Sale (POS) transaction. A typical POS transaction is a retail transaction. The machine 14 generates a paper copy 12 of the transaction. The machine 14 also generates a transaction signal, which is representative of the transaction recorded on the paper 12, and is transmitted along a first cable 16 to a first computer 18. Thus, the transaction signal includes alphanumeric data related to the transaction, such as date, description of goods, quantity and amount of sales.

The apparatus 10 of the present invention is suited for use in a POS transaction of the type requiring a human signature for verification of the transaction, as well as for requesting extension of credit for the transaction by a credit company. When a paper copy 12 of the POS transaction is generated, the paper copy 12 is placed on an electronic tablet 24. The electronic tablet 24 can electronically capture images of drawings drawn thereon. The purchaser applies his/her signature on the paper copy 12 over the tablet 24. Thus, the human signature is recorded on the paper copy 12 and is simultaneously captured electronically by the tablet 24. The human signature captured by the electronic tablet 24 in the form of a signature signal is sent over a second cable 26 to the first computer 18.

The first computer 18 has a keyboard 20 associated therewith and a display 22. The first computer 18 executes a program, a copy of which is attached as Exhibit A. The program electronically compresses the signature signal to form a compressed signature signal.

The compressed signature signal is combined with the transaction signal from the machine 14 to form a record signal. The record signal can be stored or can be further processed by transmitting it to a second computer 30 to request verification of the transaction or for extension of credit by the credit company. The record signal can also be used to regenerate the human signature, in printed format, from the compressed signature signal, along with a printed copy of the data from the transaction signal. In this manner, a print copy of the transaction and the human signature can always be regenerated. Typically, the regenerated transaction and human signature is performed at a location remote to the POS transaction. The regeneration of the transaction and human signature remotely can be performed as soon as the record signal is received or the regeneration can be performed remotely in time. In the event the regeneration is performed as soon as the record signal is received, the regenerated human signature can be compared to a previously recorded human signature for verification purpose. A human signature regenerated remotely in time can be used, for example, to authenticate a charge on a credit card.

The various components of the apparatus 10 can be as follows: An electronic cash register 14 can be of the type, manufactured by NCR Corporation, which supplies the transaction signal to the computer 18. The electronic tablet 24 can be of the type ScriptWriter, manufactured by Data Entry Systems, to supply a signature signal to the computer 18. The computer 18 can be an IBM PC/AT or compatible thereof executing the program attached herewith as Exhibit A. Another embodiment of the invention includes a computer with a built-in writing pad, or an electronic cash register connected to a writing pad. Typically, signature compression is accomplished in the writing pad, and a separate computer is not needed.

The program executed by the computer 18 in the apparatus 10 of the present invention divides the signature signal into a plurality of segments. Each of those segments is encoded in eight bits. The encoding of each of the segments is in accordance with a variation of the modified ring-encoding technique.

The operation of the program will now be described with respect to the illustration shown in Fig. 2, wherein a portion of a signature signal 32 is shown. A grid is superimposed over the portion of the signature signal 32. In one example the grid has a resolution of 50 lines per inch. The starting point of the first segment is designated as "0" and is the origin of the graph. A plurality of squares centered at the origin and whose right-hand vertical size passes through the coordinates (0,1), (0,2), (0,3) (0,5), (0,8) and (0,13) are constructed. Although these are squares, we shall refer to them hereinafter as rings and in particular as the 1-ring, 2-ring, 3-ring, 5-ring, 8-ring and 13-ring, respectively. In fact, any set of closed geometries of increasing size can be used. Thus, the 1-ring is spaced one unit or 1/50 inch away from the starting point (0,0). The 2-ring, similarly, is spaced at 2/50 inch away from (0,0). The 13-ring, the most distant ring from (0,0) is spaced at a distance of 13/50 inch away from the coordinate (0,0). As can be seen from the foregoing, the spacing of the rings from one another and from the origin is not uniform. The particular example shown hereinabove was selected based upon a Fibonacci series of 1, 2, 3, 5, 8 and 13. Of course, any other sequence can be used. As will be shown, this particular choice of ring spacings has important advantages.

Each of the rings has the number of grid points along its perimeter equal to eight times the spacing of that ring from the origin. Thus, the 13-ring has

104 (8x13) grid points along its perimeter. The 8-ring has 64 (8x8) grid points along its perimeter. The sum of all of the grid points is equal to:

$$(1+2+3+5+8+13) \times 8 = 256$$

256 grid points have the distinct advantage that they can be encoded in a single 8-bit byte.

To segment the signature signal 32 into a plurality of segments and to encode each of those segments, the following method is employed:

1. The origin is first located.
2. The points at which the signature signal 32 intersect each of the rings, i.e. a,b,...f, is calculated.
3. The grid points, i.e. A,B...F, which are closest to the points of intersection, a,b...f, of the signature signal 32 through each of the rings are also calculated.
4. Proceeding with the outermost ring, i.e., 13-ring, points P and Q which denote points on the 13-ring at a distance of one-half spacing of the grid point from the grid point F calculated in accordance with step three (3) above is determined. A cone-shape region determined by the lines PO and QO is then formed.
5. Each of the points of intersection, a,b...f, of the signature signal 32 with each of the rings is determined to see if they lie within the region formed by the cone POQ. If all of the points of intersection a, b, c, d, e and f lie within the region POQ, then for the segment from O to f of the signature signal 32, the segment is encoded as the value at F. The following table shows the encoded values:

Grid points along 1-ring: 0-7
Grid points along 2-ring: 8-23
Grid points along 3-ring: 24-47
Grid points along 5-ring: 48-87
Grid points along 8-ring: 88-151
Grid points along 13-ring: 152-255

6. If there are some points (a-f) which do not lie in the region defined by the cone POQ, then the method proceeds to the next smaller-size ring, 8-ring. A new cone is then established and points are tested against the new cone.
7. This process continues until we find the largest-size ring that satisfies the cone criteria. In the example shown in Fig. 2, it can be seen graphically that the largest-size such ring is the 3-ring. The points a, b, and c lie in the region defined by the cone POQ for the 3-ring.
8. The segment O-c is then encoded. The origin is shifted to C, the grid point nearest to c. The method returns back to step two (2).

As can be seen from the foregoing, each of the segments of the signature signal 32 can be stored in an 8-bit byte storage location, the canonical microprocessor data size. The signature signal 32 is segmented into a plurality of such 8-bit byte segments. Graphically, this is shown in Fig. 3, wherein the segment S1...S_n comprise the encoded segment signature signals for the signature signal 32, e.g. with S_1 having the value for the grid point "C" etc.

Referring to Figure 4, there is shown a variation of the method of the present invention to encode a segment of the signature signal 32, which is computationally more efficient than the method described heretofore. The method is as follows:

1. The ring set is placed at the origin.
2. A ring closest to the origin is first chosen as the current ring.
3. The smallest size of the cone emanating from the origin to the current ring containing all of the signature points, of the segment of the signature signal, from the origin to the current ring is calculated.
4. The cone of step 3 intersects the current ring, with the intersection having a length L, which is calculated.
5. The length of the intersection calculated from step 4 is compared to a specified tolerance t, (e.g. a multiple of the separation of grid points along the current ring).
6. If the length does not exceed the tolerance, the next outermost ring is chosen as the current ring and the method reverts back to step three (3).
7. If the length exceeds the tolerance, the previous current ring is chosen as the encoding ring. The grid point along the encoding ring which is closest to the signature signal is chosen to encode the signature segment.
8. The origin is shifted to the chosen grid point. The method reverts back to step two (2).

Referring to Fig. 5, there is shown a variation of the method of the present invention. Signature segments S_1 and S_2 are encoded, each in an 8-bit byte field, in accordance with the method of the present invention. However, signature segment S_3 can be an address to a library of encoded signature segments, determined a prior, to be the most likely ones to follow the observed values of signature segments S_1 and S_2. Thus, for example, if a library contains 15 of the commonly-expected types of encoded signature segments, the coding for segment 3 needs to be only 4-bits wide. Four-bits would address all 16 possible cases for the address location within the library. (The 16th case is where the next segment is not one of the segments from the library). Thus, if the encoded signature segment S_3 is one of the encoded signature segments in the library, then the number of bits required to describe the encoded signature segment S_3 is reduced from eight to four bits. This provides for further compression of the signature signal. Of course, if the actual encoded signature segment S_3 is not one of the segments stored in the library, then the actual encoded signature segment S_3 would be used and this would be 8 bits more. This aspect of encoding signature segments as an address within a library of a priori determined signature segments is termed conditional coding. As shown in Fig. 5, S_4 ... S_n can each be either an address in the library of a priori determined conditionally most likely encoded signature segments, or the encoded signature segment itself if the actual encoded signature segment is not one contained within the library.

Referring to Fig. 6, there is shown graphically yet another variation of the present invention. The signature signal 32 can be encoded by a number of different methods (including well-known prior art methods). Thus, for example, the encoded signature segments comprising of S_1 ... S_n is encoded based upon an encoding method E. Using a second different encoding method, E', the identical signature signal 32 is encoded as segments S_1' ... S_k'. If the result of encoding by method E' results in an encoded signature signal requiring less total storage than the encoded signature signal method E, then a single bit (preceding or following the encoded signature signal) is used to distinguish the type of method of encoding which resulted in that encoded signature signal. In the example shown in Fig. 6, using the method E', K bytes of storage are required for the signature signal 32. Using the encoding method E, N bytes of storage are required. If K is less than N, then the actual encoded signature signal would comprise a bit signifying that method E' was used, followed by the K bytes of the encoded signature. Thus, in the example shown in Fig. 6, a bit of "1" followed by $S_1' \ldots S_{k'}$ is used to denote that the method E' was used with the resultant encoded signature signal comprising $S_1' \ldots S_k'$. Conversely, if method E had resulted in fewer bytes of storage requirements, the initial bit would be set to "0", signifying that the method E was used.

One particular aspect of signature verification of POS transactions is the desirability of preventing forgeries. In another variation of the method of the present invention, an internal clock within the computer 18 is used to keep track of the timing for each of the encoded segments. Thus, as the human signature is captured by the electronic tablet 24, a real-time clock is used to time the writing of each of the segments or group of segments, of the human signature. Thereafter, as each segment is encoded, a timing signal associated with that signature segment or group of segments is also stored. Thus, referring to Fig. 7, there is shown a signature encoded by the method of the present invention comprising of segments $S_1 \ldots S_n$. For each of the signature segments S_i, there is a timing signal T_i associated therewith. Both the encoded signature segment and the timing signal for all of the segments are encoded and combined with the transaction signal to form the record signal. As part of signature verification, another aspect of the present invention can be to compare the timing relationship of the writing of each of the segments to the stored data with regard to the timing of writing of each of the segments. This will be in addition to the comparison of the segments of the signatures themselves. This would further foil incidence of forgery.

As can be seen from the foregoing, the method and apparatus of the present invention provides for a simplified and economic solution to the problem of total electronic POS transaction involving a human signature for verification of this transaction, as well as for request of extension of credit by a credit company. Although the invention has been described with respect to a human signature, the invention is also applicable any other form of handwritten information. Thus, as used herein, the term "human signature" includes all types of handwritten information. Further, although the invention has been described with respect to a POS retail transaction, the invention also has applicability in any type of transaction, involving a human signature (including handwritten information), such as signed securities exchange transaction, signed electronic fund transfer or electronic teller transaction, or signed proof of delivery transaction.

Although the apparatus is shown as employing a general-purpose computer 18 with the program attached as Exhibit A hereto, the apparatus to accomplish the method of the present invention can also be a dedicated hardware including a DSP signal processor with appropriate circuitry. Finally, although the Fibonacci series 1,2,3,5,8,13... is used for the construction of the spacing of the rings, the invention is not so limited. Any spacing of the rings can be used so long as the total number of grid points along the perimeter of all of the rings is less than or equal to 256, in the case of a computer with an 8-bit byte. Similarly, the method can be extended to computers that use memory lengths other than 8-bit bytes (such as 16 bits, or 32 bits).

EXHIBIT A

SIGNATURE COMPRESSION SOFTWARE

CODE.C

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <ctype.h>
#include "clisp.h"
#include "tekscreen.h"
#include "sig.h"

static int g_initialized = FALSE;

static int r[RINGS+1] = {0, 1, 2, 3, 5, 8, 13};
static int sr[RINGS+1], nr[RINGS+1], ar[RINGS+1],
           br[RINGS, cr[RINGS+1], dr[RINGS+1],
           er[RINGS+1], fr[RINGS+1], gr[RINGS+1];

static int seg_hits[256], seg_tot=0;
static int ring_hits[RINGS+1];

static
code_setup(scale)
        int scale;
{
        int i;

        sr[0] = nr[0] = ar[0] = br[0] = 0;
        cr[0] = dr[0] = er[0] = fr[0] = 0;
        gr[0] = 0
        for(i=1; i<=256; i++) {
                seg_hits[i] = 0;
        }
        for(i=1; i<=RINGS; i++) {
                ring_hits[i] = 0;
                sr[i] = scale * r[i];
                nr[i] = nr[i-1] + 8 * r[i];
                ar[i] = 2 * r[i] + nr[i-1];
                br[i] = 4 * r[i] + nr[i-1];
                cr[i] = 6 * r[i] + nr[i-1];
                dr[i] = ar[i] - r[i];
                er[i] = ar[i] + r[i];
                fr[i] = br[i] + r[i];
                gr[i] = cr[i] + r[i];
        }
}

EncodePath(lst, fname, scale)
        char *lst;
        char *fname;
```

EXHIBIT A

SIGNATURE COMPRESSION SOFTWARE

CODE.C

```
        int scale;
{
        int i, n, ring;
        char *ax;
        unsigned char c;
        long tot, old[256];
        FILE *fp;
        PATH *p;

        if(!g_initialized) {
                code_setup(scale);
                g_initialized = TRUE;
        }
        if(!NetFOpen(&fp, fname, "put")) {
                return(FALSE);
        }
        fwrite(&scale, sizeof(int), 1, fp);
        foreach(ax, lst) {
                p = (PATH *)car(ax);
                fwrite(&p->npts, 1, sizeof(int), fp);
                n = p->pts[0].x; fwrite(&n,
                    sizeof(int), 1, fp);
                n = p->pts[0].y; fwrite(&n,
                    sizeof(int), 1, fp);
                for(i=1; i < p->npts; i++) {
                        c = encode_point(scale,
                            p->pts+i, p->pts+i-1);
                        fwrite(&c, sizeof(char), 1, fp);
                        seg_hits[c]++;
                        seg_tot++;
                }
        }
        fclose(fp);
        if(getenv("NOSTAT")) return;
        if((fp = fopen("rings.dat", "a+t")) != NULL) {
                for(i=1; i<=RINGS; i++)ring_hits[0]
                    += ring_hits[i];
                fprintf(fp, "%10d : %10d %10d %10d %10d
                 %10d %10d\n", ring_hits[0],
                   ring_hits[1], ring_hits[2],
                   ring_hits[3], ring_hits[4],
                   ring_hits[5], ring_hits[6]);
                fclose(fp);
        }
        if(!NetFOpen(&fp, "segs.dat", "browse")) {
                if(NetFOpen(&fp, "segs.dat", "put")) {
                        tot = 0;
                        fwrite(&tot, sizeof(long),
```

EXHIBIT A

SIGNATURE COMPRESSION SOFTWARE

CODE.C

```
                                    1, fp);
                        for(i=0; i<256; i++) {
                                fwrite(&tot,
                                sizeof(long), 1, fp);
                        }
                        fclose(fp);
                }
        }
        if(NetFOpen(&fp, "segs.dat", "database")) {
                fread(&tot, sizeof(long), 1, fp);
                tot +=seg_tot;

                for(i=0; i<256; i++) {
                        fread(old+i; sizeof(long),
                              1, fp);
                        old[i] += seg_hits[i];
                }
                rewind(fp);
                fwrite(&tot, sizeof(long), 1, fp);
                for(i=0; i<256; i++) {
                        fwrite(old+i, sizeof(long),
                               1, fp);
                }
                fclose(fp);
        }
}

DecodePath(lstp, fname)
        char **lstp;
        char *fname;
{
        FILE *fp;
        DOUBLET a, *p, *q;
        PATH *path;
        int i, npts, scale;
        char *s, *ax;
        unsigned char c;

        *lstp = NULL;
        if(!NetFOpen(&fp, fname, "browse")) {
                return(FALSE);
        }
        npts = 0;
        if(!fread(&scale, sizeof(int), 1, fp))  {
                fclose(fp);
                return(FALSE);
        }
        if(!g_initialized) {
```

EXHIBIT A

SIGNATURE COMPRESSION SOFTWARE

CODE.C

```
                code_setup(scale);
                g_initialized = TRUE;
        }
        while(fread(&npts, sizeof(int), 1, fp)) {
                path = (PATH *)calloc(1, sizeof(PATH));
                path->npts = npts;
                path->pts = (DOUBLET *)calloc(npts,
                            sizeof(DOUBLET));
                append(lstp, path);
                if(!fread(path->pts, sizeof(DOUBLET),
                    1, fp)) {
                        fclose(fp);
                        return(FALSE);
                }
                for(i=1; i<npts; i++) {
                        if(!fread(&c, sizeof(char),
                          1, fp)) {
                                fclose(fp);
                                return(FALSE);
                        }
                        decode_point(scale, path->pts+1,
                          path->pts+i-1, c);
                }
        }
        fclose(fp);
        return(TRUE);
}

static
encode_point(scale, p, q)
        DOUBLET *p, *q;
{
        int n;
        int i, x, y, ax, ay, m;

        x = p->x - q->x;
        y = p->y - q->y;
        x /= scale;
        y /= scale;
        ax = ABS(x);
        ay = ABS(y);
        m = MAX(ax, ay);
        if(!x && !y) {
                Message("encoding error");
        }
        for(i=1; i<=RINGS; i++) {
                if(m == r[i]) {
                        ring_hits[i]++;
```

EXHIBIT A

SIGNATURE COMPRESSION SOFTWARE

CODE.C

```
                        if(m == ax) {
                            if(x > 0) {
                                n = (y <= 0) ?
                                    nr[i-1]
                                    -y : nr[i]
                                    -y;
                            } else {
                                n = br[i] +y;
                            }
                            return(n);
                        } else {
                            n = (y < 0) ? ar[i]
                                - x : cr[i] + x;
                            return(n);
                        }
                }
        }
        return(-1);
}

static
decode_point(scale, p, lastp, n)
        int scale;
        DOUBLET *p, *lastp;
        int n;
{
        int i, x, y;

        for(i=1; i<=RINGS; i++) {
                if(n >= nr[i-1] && n < nr[i]) {
                        break;
                }
        }
        if(n >= dr[i] && n <= er[i]) {
                y = -r[i];
                x = ar[i] - n;
        } else if (n >= fr[i] && n <= gr[i]) {
                y = r[i];
                x = n - cr[i];
        } else if (n >= er[i] && n <= fr[i]) {
                x = -r[i];
                y = n - br[i];
        } else if (n <= dr[i]) {
                x = r[i];
                y = nr[i] - n;
        } else {
                Message("encode error");
        }
```

EXHIBIT A

SIGNATURE COMPRESSION SOFTWARE

CODE.C

```
        p->x = lastp->x + x * scale;
        p->y = lastp->y + y * scale;
}

#idef TEST
#define C_PGREEN                        C_MAGENTA
main(argc, argv)
        int argc;
        char **argv;
{
        char *lst = NULL, *fname;

        fname = (argc > 1 ) ? argv[1] : "tmp.cmp";
        init_screen();
        lst = NULL;
        if(DecodePath(&lst, fname)) {
                DrawOffsetPath(lst, NULL, C_WHITE);
                FreePath(lst);
                InputKeyboard();
        } else {
                Message("Cannot read file
                        \"%s\"", fname);
        }
        restore_screen();
}
init_screen()
{
        EgaOpen(1);
        EgaWait(10);
        EgaBackground(1, 0);
        EgaReadWritePage(0, 0);
        EgaVisualPage(0);
        EgaSetFont(F_STD);
        SetupQuestion(C_LRED, C_WHITE, C_LRED,
                C_DBLUE, C_DBLUE);
        SetupMessage(C_LRED, C_WHITE, C_LRED);
        EgaSetColorMap(C_PGREEN, 1, 3, 2);
        EgaSetFong(F_SMALL);
        ResetWindow();
}

restore_screen()
{
        EgaClose();
        EgaWait(1);
}
#endif
```

EXHIBIT A

SIGNATURE COMPRESSION SOFTWARE

SIG.C

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <string.h>
#include <ctype.h>
#include "clisp.h"
#include "sig.h"

static int g_scale = 1; /*assume integer scale
                          from rgp -> sgp */
static int r[RINGS+1] = {0, 1, 2, 3, 5, 8, 13};
DOUBLET s_pts[MAXP];

static int sr[RINGS+1];
                                        /* scaled r's
                                        (reconstruction grid in
                                        sample coordinates) */
static float ss;                        /* scale squared */
static float cone_tol = CONE_TOLERANCE;
                                        /* min theta resulting in
                                        "bad" intersections on
                                        each ring */
static float xi[RINGS+1];

static float eps;         /* relative fp tolerance */
static big_cone(), filter_pts(), intersect_path_ring(),
                intersect_seg_ring(),
                last_intersection(), long_seg(),
                nearest_rgp(), new_cone(), ring_point(),
                setup();

CompressPath(lst, scale, fname)
        char    **lst;
        int             scale;
        char    *fname;
{
        int i, n, total_pts, lifts;
        char *ar, *s, *flst = NULL;
        DOUBLET pt;
        PATH *p, *fp;

        FilterPath(lst, &flst, scale);
        EncodePath(flst, fname, scale);
        FreePath(flst);
}


FilterPath(lst, flstp, scale)
```

EXHIBIT A

SIGNATURE COMPRESSION SOFTWARE

SIG.C

```
        char *lst, **flstp;
        int     scale;
{
        char *ax;
        PATH *p, *fp;
        setup(scale);
        *flstp = NULL;
        foreach(ax, lst) {
                p = (PATH *)car(ax);
                filter_pts(&fp, p);
                append(flstp, fp);
        }
}

#define SQUARE(a) ((a) * (a))
static
setup(scale)
        int scale;
{
        int i;
        float tmp;
        char *s;

        g_scale = scale;
        tmp = g_scale * cone_tol;
        ss = SQUARE(tmp);
        for(sr[0] = 0, xi[0] = 0, i=1; i<=RINGS; i++) {
                sr[i] = g_scale * r[i];
                xi[i] = tmp / (sqrt(2 * (SQUARE(sr[i]
                        - 1) + SQUARE(sr[i]))));
        /*      xi[i] = tmp / (sqrt(ss + SQUARE(sr[i])));
                        */
        }
        eps = REL_TOL * g_scale;
}

static
filter_pts(fpathp, path)
        PATH **fpathp, *path;
{
        int x, i, j, k, n, check, ring, kmax,
                newlink, npts;
        int colinear, cmax, o_ndx, tlen;
        float tmp;
        DOUBLET tau, o, rho, *p, *q, phi, c;
        FDOUBLET vmin, vmax;

        kmax = path->npts-2;
```

EXHIBIT A

SIGNATURE COMPRESSION SOFTWARE

SIG.C

```
npts = 0
k = 0; /* index into unfiltered array */
n = 0; /* index into filtered array */
o_ndx = 0;
s_pts[0] = path->pts[0];
o = s_pts[0];
SET_PT(vmin, s_pts[0];
nearest_rgp(&vmin);
SET_PT(s_pts[0], vmin);
while(k < kmax) {
for(; ; k++) {
        VSub(&rho, path->pts+k, &o);
        tlen = VAbs(&rho);
        if(tlen >= 1)break;
}
PollKeyboard();
SET_PT(vmin, rho);
Nmlize(&vim);
vmax = vmin;    /* initialize cone angle at 0 */
newlink = FALSE;
colinear = FALSE;
cmax = 0;
tau.x = tau.y = 0
while(k < kmax && !newlink) {   /* look for big
                                   cone */
        k++;
        VSub(&rho, path->pts+k, path->pts+k-1);
        if(VAbs(&rho) == 0 {
                continue;
        }
        if(DOT(rho, tau) < 0) {
                for(ring=1; ring <= RINGS &&
                   sr[ring] <= tlen; ring++);
                ring--;
                if(newlink = ring > 1) {
                        continue;
                }
        }
        PollKeyboard();
        ring = 0;
        check = FALSE;
        VSub(&tau, path->pts+k, &o);
        if((tlen = VAbs(&tau)) > sr[RINGS]) {
                newlink = TRUE;
                long_seg(&ring, o_ndx, path,
                   &vmin, &vmax, &o);
                continue;
        }
```

EXHIBIT A

SIGNATURE COMPRESSION SOFTWARE

SIG.C

```
                if(check = new_cone(&vmin, &vmax,
                      &tau)) {
                    if(colinear) {
                        cmax = MAX(cmax, tlen);
                        for(ring=1; ring <= RINGS
                           && sr[ring] <= cmax;
                           ring++);

                        ring--;
                        newlink = ring > 1;
                        colinear = !newlink;
                    } else  {
                        newlink = big_cone(&ring,
                          &vmin, & vmax, tlen);
                    }
                } else {
                    colinear = TRUE;
                        cmax = MAX(cmax, tlen);
                    }
                }
                if(newlink) {
                    for(n++;!intersect_path_ring
                      (path, s_pts+n, &o, o_ndx,
                       ring, &x) && ring; ring--);
                    k = x;          /* end of segment
                                    containing
                                    intersection */
                    o_ndx = k;
                    o = s_pts[n];
                    PollKeyboard();
                }
            }
            k = o_ndx;
            while(k < path->npts &&
                        last_intersection(s_pts+n+1,
                          path, s_pts+n, &k)) {
                PollKeyboard();
                n++;
            }
            npts = n+1;
            *fpathp = (PATH *)calloc(1, sizeof(PATH));
            (*fpathp)->pts = (DOUBLET *)calloc(npts,
               sizeof(DOUBLET));
            (*fpathp)->npts = npts;
            for(i=0, p=(*fpathp)->pts, q=s_pts; i<npts;
               i++, p++, q++) {
                p->x = q->x; p->y = q->y;
            }
```

EXHIBIT A

SIGNATURE COMPRESSION SOFTWARE

SIG.C

```
}

static
new_cone(vmin, vmax, tau)
        FDOUBLET *vmin, *vmax;
        DOUBLET *tau;
{
        float tmp;

        if((tmp = CROSS(*vmin, *tau)) >= eps) {
                if(CROSS(*vmax, *tau) >= eps) {
                        SET_PT(*vmax, *tau);
                        Nmlize(vmax);
                        return(TRUE);
                }
        } else if (tmp < -eps) {
                SET_PT(*vmin, *tau);
                Nmlize(vmin);
                return(TRUE);
        }
        return(FALSE);
}

static

big_cone(ringp, vmin, vmax, tlen)
        int *ringp;
        FDOUBLET *vmin, *vmax;
        int tlen;
{
        float sin_theta, tmp, s;
        FDOUBLET pmin, pmax;
        int i, newlink, j;

        sin_theta = CROSS(*vmin, *vmax);/* positive */
        *ringp = 0;
        for(i=1, newlink=FALSE; i<=RINGS &&
          (sr[i] <= tlen) && !newlink; i++) {
                if(sin_theta >= xi[i]) {
                        ring_point(&pmin, vmin, i);
                        ring_point(&pmax, vmax, i);
                        if(SFDist(&pmin, &pmax) >= ss) {
                                newlink = TRUE;
                                *ringp = i;
                        }
                }
        }
        return(newlink);
```

EXHIBIT A

SIGNATURE COMPRESSION SOFTWARE

SIG.C

```
}

static
long_seg(ringp, k, path, vmin, vmax, o)
        int *ringp, k;
        PATH *path;
        FDOUBLET *vmin, *vmax;
        DOUBLET *o;
{
        FDOUBLET pmin, pmax, vmn, vmx;
        DOUBLET p;
        int i, j;
        float tmp;

        for(*ringp=1, i=RINGS; i>1; i--) {
                vmn = *vmin; vmx = *vmax;
                intersect_path_ring(path, &p, o, k,
                   i, &j);
                VSub(&p,  &p, o);
                new_cone(&vmn, &vmx, &p);
                if(CROSS(vmn, vmx) < xi[i]) {
                        *ringp = i;
                        return;
                }
                ring_point(&pmin, &vmn, i);
                ring_point(&pmax, &vmx, i);
                if(SFDist(&pmin, &pmax) <= ss) {
                        *ring = i;
                        return;
                }
        }
        return;
}

static
ring_point(p, v, ring)
        FDOUBLET *p, *v;
        int ring;
{
        float sx, sy;

        /*      should check for 0       */
        if((sx = ABS(v->x)) >= (sy = ABS(v->y))) {
                p->x = sr[ring];
                if(sx != v->x) {
                        p->x = -p->x;
                }
                p->y = p->x * v->y / v->x;
```

EXHIBIT A

SIGNATURE COMPRESSION SOFTWARE

SIG.C

```
            } else {
                    p->y = sr[ring];
                    if(sy != v->y) {
                            p->y = -p->y;
                    }
                    p->x = p->y * v->x / v->y;
            }
    }

    static
    last_intersection(p, path, o, kp)
            DOUBLET *p, *o;
            PATH *path;
            int *kp;
    {
            int k, i, n, ndx;
            DOUBLET rho, pt, v;
            FDOUBLET vmin, vmax, vmn, vmx, pmin, pmax;

            k = ndx = *kp;
            *kp = path->npts-1;
            for(i=ndx;i < path->npts;i++) {
                    VSub(&rho, path->pts+i, o);
                    if(VAbs(&rho) >= 1) break;
            }
            SET_PT(vmin, rho);
            Nmlize(&vmin);
            vmax = vmin;
            for(i=1, n=0; i<=RINGS; i++) {
                    if(!intersect_path_ring(path, &pt,
                       o, ndx, i, &k)) {
                            break;
                    }
                    VSub(&v, &pt, o);
                    vmn = vmin; vmx = vmax;
                    new_cone(&vmn, &vmx, &v);
                    ring_point(&pmin, &vmn, i);
                    ring_point(&pmax, &vmx, i);
                    if(SFDist(&pmin, &pmax) >= ss) {
                            break;
                    }
                    n = i;
                    *p = pt;
                    *kp = k;
            }
            return(n > 0);
    }
```

EXHIBIT A

SIGNATURE COMPRESSION SOFTWARE

SIG.C

```
static
intersect_path_ring(path, p, o, ndx, ring, endp)
        PATH *path;
        DOUBLET *p, *o;
        int ndx, ring, *endp;
{
        int n, k, tmp;
        DOUBLET a;

        if(!ring)return(FALSE);
        for(k = ndx, n=path->npts; k<n; k++) [
                VSub(&a, path->pts+k, o);
                if((tmp = VAbs(&a) - sr[ring]) >= 0) {
                        if(!intersect_seg_ring(p,
                           o, path->pts+k-1, path->
                           pts+k, ring))
{
                                return(FALSE);
                        }
                        *endp = (tmp == 0) ? k+1 : k;
                        return(TRUE);
                }
        }
        return(FALSE);
}

static
intersect_seg_ring(p, o, inp, outp, ring)
        DOUBLET *p, *o, *inp, *outp;
        int ring;
{
        FDOUBLET v, a;
        FDOUBLET t, tmp;
        float tt;

        v.x = outp->x - inp->x;
        v.y = outp->y - inp->y;
        if(v.x == 0) {
                tmp.x = inp->x;
                tmp.y = v.y > 0 ? o->y + sr[ring]
                    : o->y - sr[ring];
        } else if (v.y == 0) {
                tmp.x = v.x > 0 ? o->x + sr[ring]
                    : o->x - sr[ring];
                tmp.y = inp->y;
        } else {
                a.x = inp->x - o->x;
                a.y = inp->y - 0->y;
```

EXHIBIT A

SIGNATURE COMPRESSION SOFTWARE

SIG.C

```
                t.x = (((v.x > 0) ? sr[ring] :
                   -sr[ring]) - a.x) /v.x;
                t.y = (((v.y > 0) ? sr[ring] :
                   -sr[ring]) - a.y) /v.y;
                tt = MIN(t.x, t.y);
                if(tt < 0) {
                        t.x = -t.x;
                        t.y = -t.y;
                        tt = MIN(t.x, t.y);
                        if(tt < -eps {
                                return(FALSE);
                                  /* parameter error ! */
                        }
                }
                tmp.x = a.x + tt * v.x + o->x;
                tmp.y = a.y + tt * v.y + o->y;
        }
        nearest rgp(&tmp);
        SET_PT(*p, tmp);
        return(TRUE);
}
static
nearest_rgp(p)
        FDOUBLET *p;
{
        int i, s;
        FDOUBLET q;

        q.x = (int)(p->x + .5);
        q.y = (int)(p->y + .5);
        s = g_scale>>1;
        i = (int)(q.x) % g_scale;
           /* use mask if scale is power of 2 */
        p->x = (i <= s) ? q.x - i : q.x + g_scale - i;
        i = (int)(q.y) % g_scale;
        p->y = (i <= s) ? q.y = i : q.y + g_scale - i;
}
```

EXHIBIT A

SIGNATURE COMPRESSION SOFTWARE

SIG.H

```
#ifndef TRUE
#define TURE 1
#define FALSE 0
#endif
#include "path.h"

#define RINGS 6
#define CONE_TOLERANCE 1.5
#define REL_TOL 1.0e-7

#define MAXP 5000
extern DOUBLET s_pts[];
```

EXHIBIT A

SIGNATURE COMPRESSION SOFTWARE

PATH.H

```
#define DOUBLET struct doublet
DOUBLET {
        int x,y
};

#define FDOUBLET struct fdoublet
FDOUBLET {
        float x,y;
};

#define PATH struct path
PATH {
        DOUBLET *pts;
        int     npts;
};

#define ABS(a)   (((a) > 0) ? (a) : -(a))
#define DOT(p, q) ((p).x * (q).x + (p).y * (q).y)
#define CROSS(p. q) ((p).x * (q).y - (p).y * (q).y)
#define MAX(a, b) (((a) > (b)) ? (a) : (b))
#define MIN(a, b) (((a) < (b)) ? (a) : (b))
#define SET_PT(p, q) (p).x = (q).x, (p).y = (q).y

float SFDist();
long SDist();
```

WHAT IS CLAIMED IS:

1. A method for electronically processing a transaction having a human signature for verification of said transaction, said method comprising the steps of:

electronically capturing said transaction at said point of transaction to form a transaction signal;

electronically capturing said human signature at said point of transaction to form a signature signal;

compressing said signature signal to form a compressed signature signal;

combining said compressed signature signal with said transaction signal to form a record signal; and

processing said record signal.

2. The method of Claim 1 wherein said processing step further comprises storing said record signal.

3. The method of Claim 1 wherein said processing step further comprises transmitting said record signal.

4. The method of Claim 1 wherein said processing step further comprises verifying said record signal.

5. The method of Claim 1 further comprising the steps of

generating a paper copy of said transaction at said point of transaction;

forming a human signature on said paper copy at said point of transaction while simultaneously performing the step of electronically capturing said human signature.

6. The method of Claim 1 further comprising the step of:

remotely generating a print copy of said human signature from said record signal.

7. The method of Claim 6 wherein said generating is performed also remotely in time.

8. The method of Claim 1 wherein said transaction is a retail sales transaction.

9. The method of Claim 1 wherein said transaction is signed securities exchange transaction.

10. The method of Claim 1 wherein said transaction is a signed electronic fund transfer transaction.

11. The method of Claim 1 wherein said transaction is an electronic teller transaction.

12. The method of Claim 1 wherein said transaction is a package delivery transaction.

13. The method of Claim 1 wherein said compressing step further comprises:

dividing said signature signal into a plurality of signature segment signals; and encoding each signature segment signal in accordance with the Modified Ring Encoding Method.

14. The method of Claim 13 wherein said encoding step further comprises:

selecting N rings in said Modified Ring Encoding Method with each ring having a number of grid points, and where the total number of grid points is less than or equal to 256.

15. The method of Claim 14 wherein said rings are sized by the Fibonacci series of

1, 2, 3, 5, 8, 13

16. The method of Claim 1 wherein said compressing step further comprises:

dividing said signature signal into a plurality of signature segment signals;

compressing each signature segment signal into a compressed segment signal;

forming a library of compressed segment signals that are a priori determined; and

encoding one of said signature segment signals as an address to one of said compressed segment signals from said library.

17. The method of Claim 1 wherein said compressing step further comprises:

compressing said signature signal by a first compressing method to form a first compressed signature signal;

compressing said signature signal by a second compressing method to form a second compressed signature signal;

comparing said first compressed signature signal to said second compressed signature signal;

selecting the smaller size of the two first and second compressed signature signal as the compressed signature signal and using an additional bit to distinguish the two methods.

18. The method of Claim 1 wherein said capturing step further comprises:

electronically capturing the segments of said human signature;

clocking the timing of the writing of said segments of said human signature; and

forming a signature signal based upon said segments and said timing.

19. An apparatus for electronically processing a transaction having a human signature for verification of said transaction, said apparatus comprising:

means for electronically capturing said transaction at said point of transaction to form a transaction signal;

means for electronically capturing said human signature at said point of transaction to form a signature signal;

means for compressing said signature signal to form a compressed signature signal;

means for combining said compressed signature signal with said transaction signal to form a record signal; and means for processing said record signal.

20. The apparatus of Claim 19 further comprising:

means for generating a paper copy of said transaction at said point of transaction, wherein said human signature can be formed over said paper copy while simultaneously being electronically captured by said capturing means.

21. The apparatus of Claim 19, wherein said compressing means further comprises:

means for dividing said signature signal into a plurality of signature segment signals; and

means for encoding each signature segment signal in accordance with a variation of the Modified Ring Encoding Method.

22. The apparatus of Claim 21 wherein said encoding means further comprises:

means for selecting N rings in said Modified Ring Encoding Method with each ring having a number of grid points along its perimeter, and where the total number of grid points is less than or equal to 256.

23. The apparatus of Claim 22, wherein said rings are sized by the Fibonacci series of

1, 2, 3, 5, 8, 13

24. The apparatus of Claim 19, wherein said compressing means further comprises:

means for dividing said signature signal into a plurality of signature segment signals;

means for compressing each signature segment signal into a compressed segment signal;

means for forming a library of a priori determined compressed segment signals; and

means for encoding one of said compressed segment signal as an address to one of said compressed segment signals from said library.

25. The apparatus of Claim 19, wherein said compressing means further comprises:

means for compressing said signature signal by a first compressing method to form a first compressed signature signal;

means for compressing said signature signal by a second compressing method to form a second compressed signature signal;

means for comparing said first compressed signature signal to said second compressed signature signal;

means for selecting the smaller size of the two first and second compressed signature signal as the compressed signature signal and using an additional bit to distinguish the two methods.

26. The apparatus of Claim 19, wherein said compressing means further comprises:

means for electronically capturing the segments of said human signature;

means for clocking the timing of the writing of said segments of said human signature; and

means for forming a signature signal based upon said segments and said timing.

27. A method of compressing a signature signal comprising the steps of:

dividing said signature signal into a plurality of signature segment signals;

encoding each signature segment in accordance with a variation of the Modified Ring Encoding Method.

28. The method of Claim 27 wherein said encoding step further comprises:

selecting N rings in said Modified Ring Encoding method with each ring having a number of grid points along its perimeter and wherein the total number grid points along the perimeter of all of the rings is less than or equal to 256.

AMENDED CLAIMS
[received by the International Bureau on 03 June 1991 (03.06.91); original claim 28 cancelled; original claims 6,17,21,25,27 amended; other claims unchanged (6 pages)]

forming a human signature on said paper copy at said point of transaction while simultaneously performing the step of electronically capturing said human signature.

6. The method of Claim 1 further comprising the step of:

remotely generating a print copy of said human signature and said transaction from said record signal.

7. The method of Claim 6 wherein said generating is performed also remotely in time.

8. The method of Claim 1 wherein said transaction is a retail sales transaction.

9. The method of Claim 1 wherein said transaction is signed securities exchange transaction.

10. The method of Claim 1 wherein said transaction is a signed electronic fund transfer transaction.

11. The method of Claim 1 wherein said transaction is an electronic teller transaction.

12. The method of Claim 1 wherein said transaction is a package delivery transaction.

13. The method of Claim 1 wherein said compressing step further comprises:

dividing said signature signal into a plurality of signature segment signals; and encoding each signature segment signal in accordance with the Modified Ring Encoding Method.

14. The method of Claim 13 wherein said encoding step further comprises:

selecting N rings in said Modified Ring Encoding Method with each ring having a number of grid points, and where the total number of grid points is less than or equal to 256.

15. The method of Claim 14 wherein said rings are sized by the Fibonacci series of

1, 2, 3, 5, 8, 13

16. The method of Claim 1 wherein said compressing step further comprises:

dividing said signature signal into a plurality of signature segment signals;

compressing each signature segment signal into a compressed segment signal;

forming a library of compressed segment signals that are a priori determined; and

encoding one of said signature segment signals as an address to one of said compressed segment signals from said library.

17. The method of Claim 1 wherein said compressing step further comprises:

compressing said signature signal by a first compressing method to form a first compressed signature signal;

compressing said signature signal by a second compressing method different from said first method to form a second compressed signature signal;

comparing said first compressed signature signal to said second compressed signature signal;

selecting the smaller size of the two first and second compressed signature signal as the compressed signature signal and using an additional bit to distinguish the two methods.

18. The method of Claim 1 wherein said capturing step further comprises:

electronically capturing the segments of said human signature;

clocking the timing of the writing of said segments of said human signature; and

forming a signature signal based upon said segments and said timing.

19. An apparatus for electronically processing a transaction having a human signature for verification of said transaction, said apparatus comprising:

means for electronically capturing said transaction at said point of transaction to form a transaction signal;

means for electronically capturing said human signature at said point of transaction to form a signature signal;

means for compressing said signature signal to form a compressed signature signal;

means for combining said compressed signature signal with said transaction signal to form a record signal; and

means for processing said record signal.

20. The apparatus of Claim 19 further comprising:

means for generating a paper copy of said transaction at said point of transaction, wherein said human signature can be formed over said paper copy while simultaneously being electronically captured by said capturing means.

21. The apparatus of Claim 19, wherein said compressing means further comprises:

means for dividing said signature signal into a plurality of signature segment signals; and

means for encoding each signature segment signal.

22. The apparatus of Claim 21 wherein said encoding means further comprises:

means for selecting N rings in said Modified Ring Encoding Method with each ring having a number of grid points along its perimeter, and where the total number of grid points is less than or equal to 256.

23. The apparatus of Claim 22, wherein said rings are sized by the Fibonacci series of

1, 2, 3, 5, 8, 13

24. The apparatus of Claim 19, wherein said compressing means further comprises:

means for dividing said signature signal into a plurality of signature segment signals;

means for compressing each signature segment signal into a compressed segment signal;

means for forming a library of a priori determined compressed segment signals; and

means for encoding one of said compressed segment signal as an address to one of said compressed segment signals from said library.

25. The apparatus of Claim 19, wherein said compressing means further comprises:

means for compressing said signature signal by a first compressing method to form a first compressed signature signal;

means for compressing said signature signal by a second compressing method different from said first compression method to form a second compressed signature signal;

means for comparing said first compressed signature signal to said second compressed signature signal;

means for selecting the smaller size of the two first and second compressed signature signals as the compressed signature signal and using an additional bit to distinguish the two methods.

26. The apparatus of Claim 19, wherein said compressing means further comprises:

means for electronically capturing the segments of said human signature;

means for clocking the timing of the writing of said segments of said human signature; and

means for forming a signature signal based upon said segments and said timing.

27. A method of compressing a signature signal comprising the steps of:

dividing said signature signal into a plurality of signature segment signals;

encoding each signature segment by selecting N rings in a Modified Ring Encoding method with each ring having a number of grid points along its perimeter and wherein the total number grid points along the perimeter of all of the rings is less than or equal to 256.

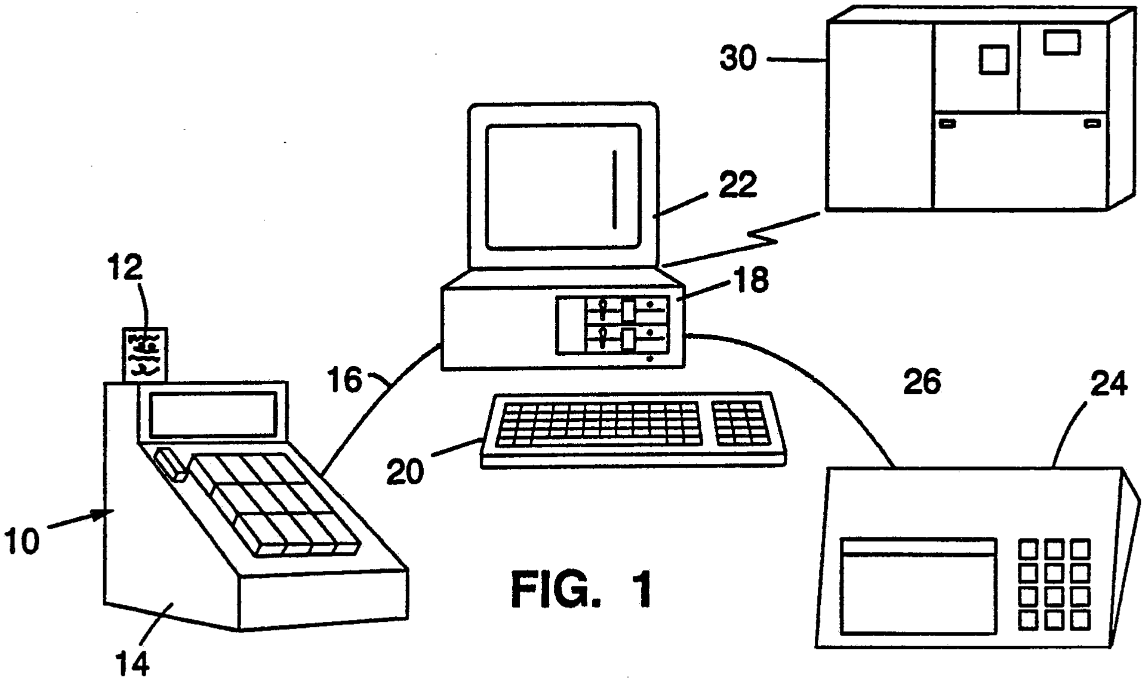


FIG. 1

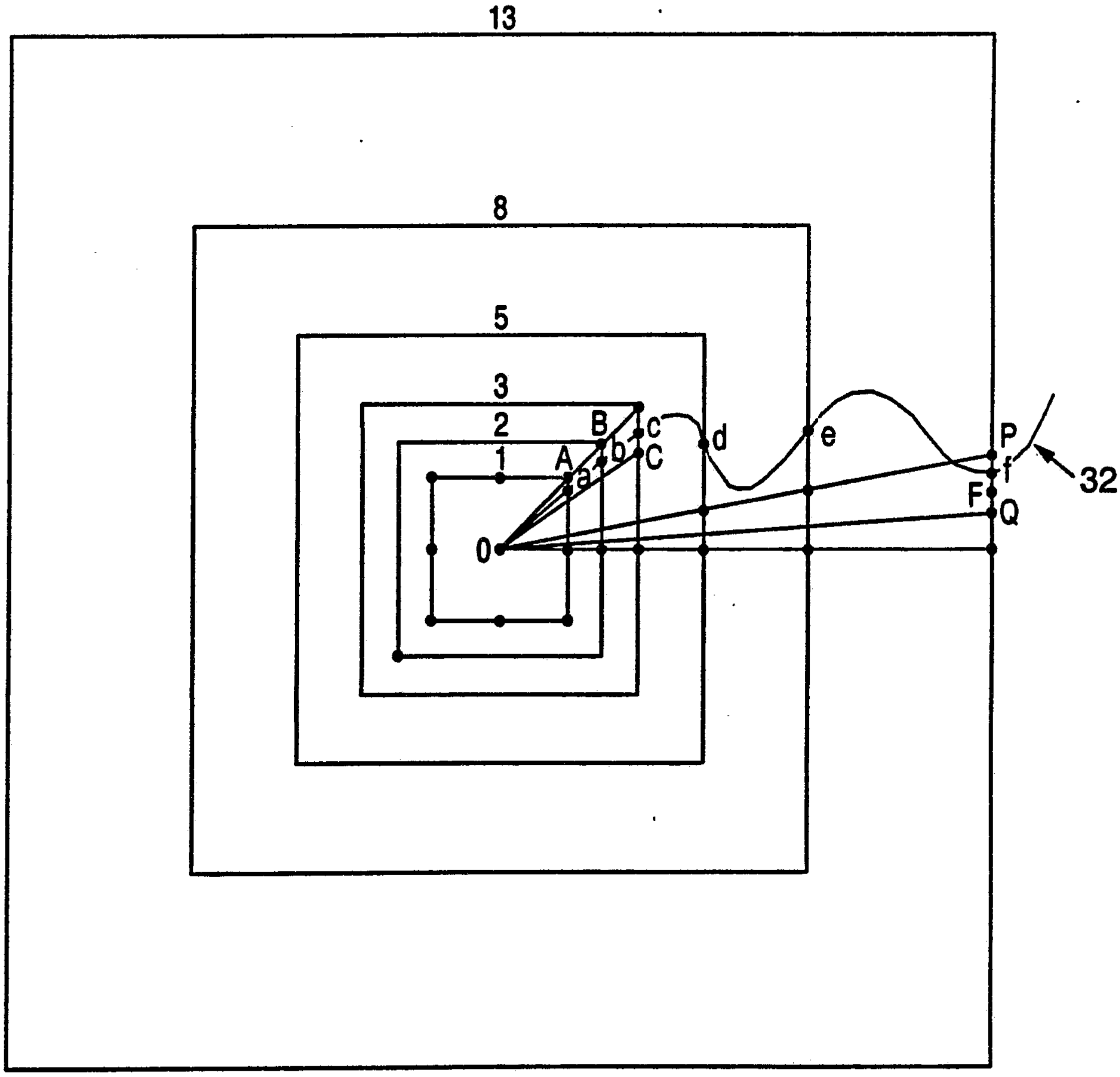
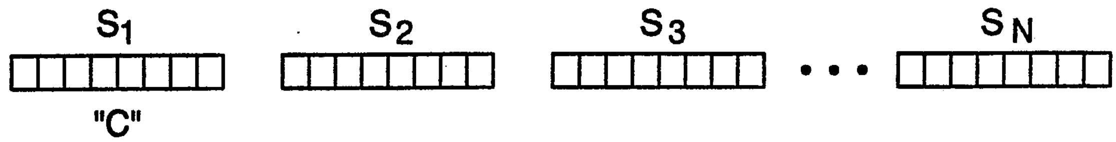
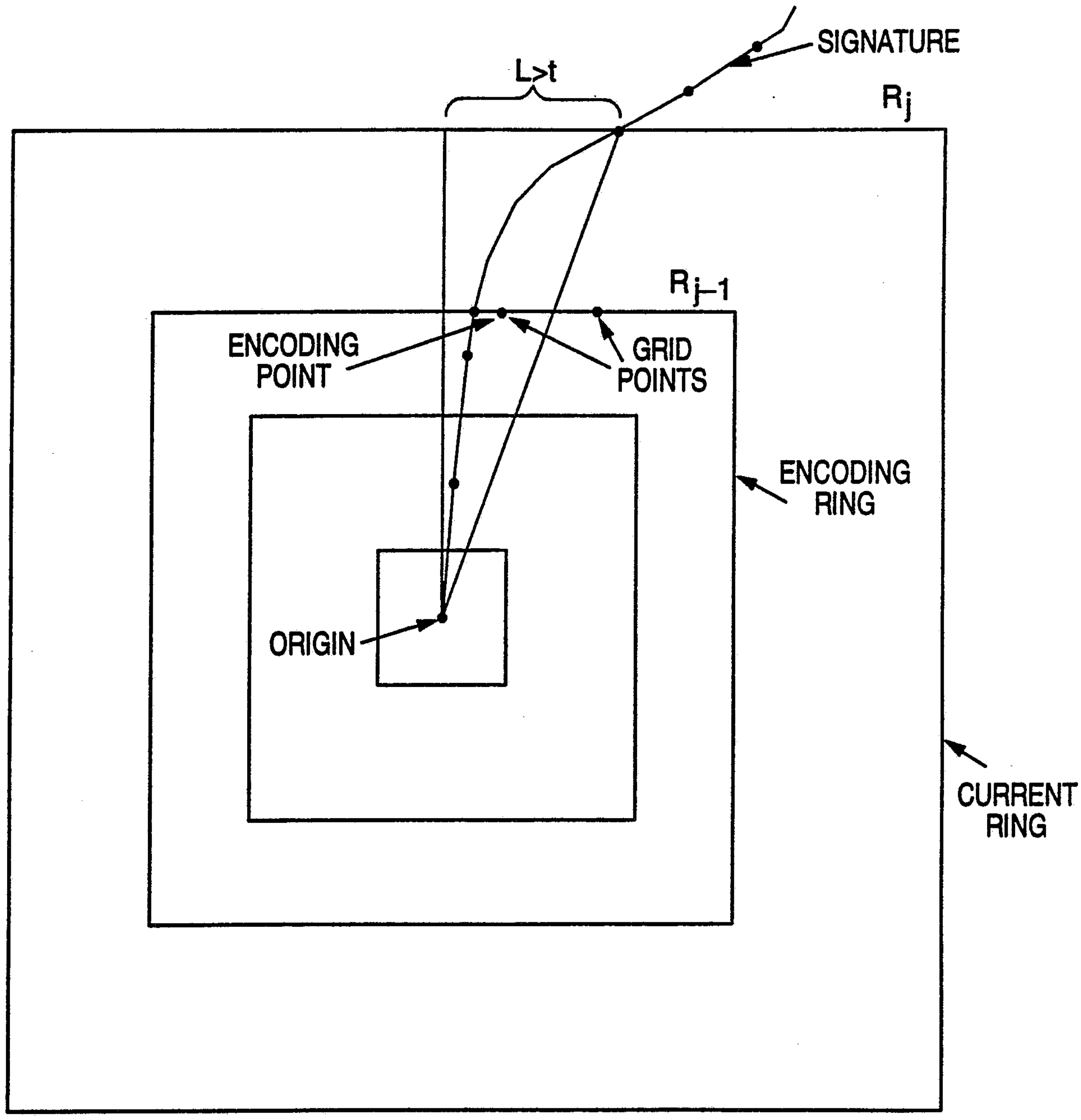


FIG. 2



S1
S2
S3
SN
"C"

FIG. 3



SIGNATURE
L>t
Rj
Rj-1
ENCODING POINT
GRID POINTS
ENCODING RING
ORIGIN
CURRENT RING

FIG. 4

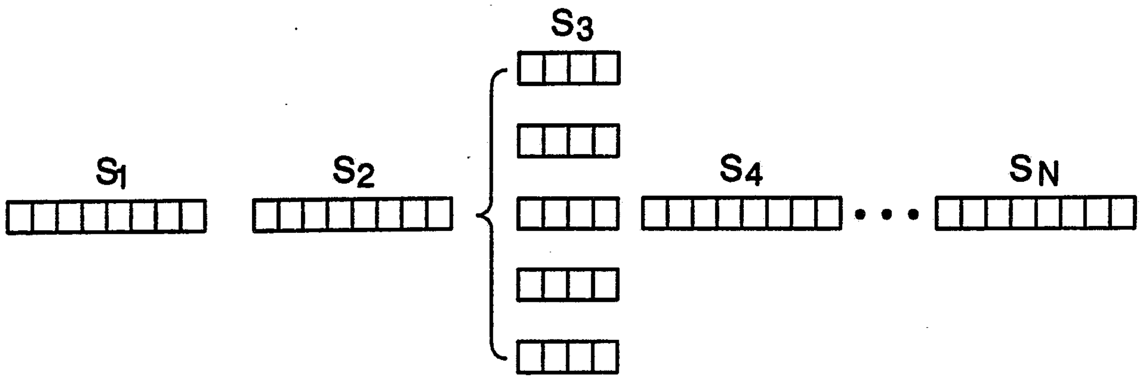


FIG. 5

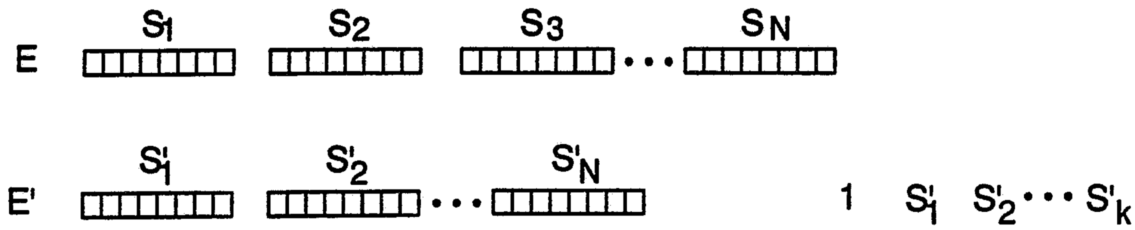


FIG. 6

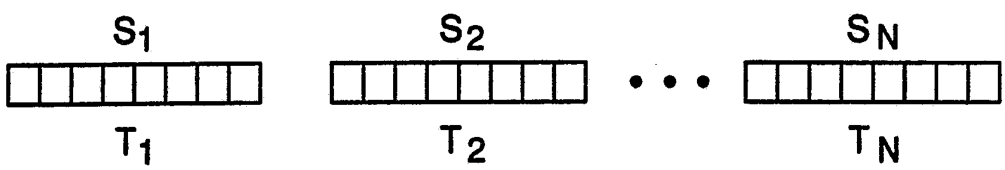


FIG. 7

INTERNATIONAL SEARCH REPORT

International Application No. PCT/US91/00072

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) [6]

According to International Patent Classification (IPC) or to both National Classification and IPC

U.S. CL. 382/3 Int.Cl.(5) G06K 9/00

II. FIELDS SEARCHED

Minimum Documentation Searched [7]

Classification System	Classification Symbols
U.S.	382/3, 56,21; 358/261.3

Documentation Searched other than Minimum Documentation to the Extent that such Documents are Included in the Fields Searched [8]

III. DOCUMENTS CONSIDERED TO BE RELEVANT [9]

Category *	Citation of Document, [11] with indication, where appropriate, of the relevant passages [12]	Relevant to Claim No. [13]
Y	US,Y, 4,752,965 (Dunkley et al.) 21 June 1988, see column 3, lines 4-23 and lines 56-66 and column 2 lines 13-22	1-12,18-20 and 26
Y	US,Y, 4,020,463 (Himmel) 26 April 1977, see column 1 paragraph 2 and column 2, paragraph 1	1-12,18-20 and26
Y	IEEE Transactions on Inf. Theory, Vol.IT-31 No. 1 issued January 1985. NEUHOFF, "A Rate and Distortion Analysis of Chain Codes for Line Drawings".	13-15,21-23, 27,28,and 35
Y	US,Y, 4,922,545 (Endoh et al.) 1 May 1990, see column 2, lines 30-50	16 and 24
Y	US,Y, 4,087,788 (Johannesson) 2 May 1978, see column 1, lines 25-51.	17 and 25

* Special categories of cited documents: [10]

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

IV. CERTIFICATION

Date of the Actual Completion of the International Search	Date of Mailing of this International Search Report
14 March 1991	16 APR 1991
International Searching Authority	**Signature of Authorized Officer**
ISA/US	Michael R. Cammarata

Form PCT/ISA/210 (second sheet) (Rev.11-87)