Systems and Computers in Japan, Vol. 31, No. 1, 2000
Translated from Denshi Joho Tsushin Gakkai Ronbunshi, Vol. J80-D-II, No. 5, May 1997, pp. 1160–1168

# A Signature Method for Image Data Using Partial Scrambling with the Generalized Peano Scan

Kazuhiro Oka and Kineo Matsui

Department of Computer Science, National Defense Academy, Yokosuka, Japan 239

## SUMMARY

We introduce a signature method which uses partial scrambling to implement the concept of active copyright protection. Active copyright protection prevents the dishonest copying of multimedia by attaching to copyrighted information in the distributed contents a confirming signature, which can be removed only by a user acting in a legitimate process. In this method, the identifying signature, which scrambles a section of the image data, is called the *logmark*. From it, the image data of the document can be identified as the user's. In this paper, the generalized Peano scan serves as the concealing function in the scrambling method, guaranteeing security. We consider the effectiveness of this technique both as a copyright protection method, since by its construction it alters the image data, and in terms of its ability to permit the removal of the signature. © 1999 Scripta Technica, Syst Comp Jpn, 31(1): 20–29, 2000

**Key words**: Active copyright protection; signatures; partial scrambling; generalized Peano scan.

## 1. Introduction

In recent years, with the development of network technology and of multimedia, image data have been frequently utilized. Also, because of the low price of personal computers and the availability of superior user interfaces, the popularization of the general-purpose computers has developed rapidly. Because of the popularity of high-speed digital communications and the high efficiency of compression coding methods, the distribution of multimedia using networks can be expected to become even more popular in the future. However, a new problem has become obvious [1, 2]: copyright infringement entailing the unauthorized reproduction of an author's distributed image data. Digital data can easily be copied on a computer, and anyone can reproduce even a large number of error-free copies of information. Such actions produce maximal disadvantage for a copyrighter and his collaborators. To ameliorate this problem, systems have been constructed [3–5] which guarantee that an author can retain control over his documents.

These methods can generally be classified into three groups. The first class of methods initially inserts information concerning the copyright into the distributed documents; this is the most commonly seen system [5–7]. In models such as those of Refs. 8 and 9, the image data usually are in a layered system. In the second class of methods, the author permits systematic access to the circulating data only to those who possess a decoder for the data. These methods employ a channel systemizer [10, 11], and use the techniques of JPEG and MPEG [12] for additive bitwise transformation. One of these methods [13] identifies the user who produces a dishonest copy. In this method, when a legitimate user transmits data, information on the user to whom the information is being transmitted is inserted into the data during the transmission, so a user who acts dishonestly can be identified. This technique can be regarded as a modification of the methods of the first class.

ln this paper, we present a third method, an expansion of the ideas of the second type. Only part of the image data

CCC0882-1666/00/010020-10
© 1999 Scripta Technica

is systematically changed, so that the rough form of the image data is still preserved, and specific information (in particular, the signature of a document's image data, which identifies the user) is added. The proposed technique is as follows. First the original image data are prepared, and then the signature data are added. The signature data form a 2-valued image of smaller size than the original data, and we store the logmark representing the signature of the author or his delegate. Thus, an arbitrary part of the original image corresponds to the signature data, and only the part corresponding to the logmark or the signature of the original image is scrambled. We use an extension of the generalized Peano scan to determine the scanning order in the region where the scrambling is being done. In this way we display the signature of the image data so that a user knows what is needed to permit the reproduction of the image data in the document. In other words, the user receives the distributed image data including the inserted signature, and can reproduce the original image data by inputting the decryption key after paying a fee.

Section 2 describes the concept of active copyright protection and the proposed specific signature method. In section 3 the partial scrambling method and its decryption method are discussed. Section 4 presents results on the application of this technique to actual image data. Section 5 gives an analysis and evaluation.

## 2. Signatures Using Partial Scrambling

### 2.1. Active copyright protection

Under the articles of copyright rule 21, a receiver must obtain an author's consent to reproduce a document. By reproduction, we mean printing, photography, facsimile, sound recording, television, or other methods of overtly reproducing the document (see articles 1 through 15 of copyright rule 2). According to these regulations, image data which pass through a network should be protected as images and as documents. However, just as with a document, since there is cultural community property, this must be stipulated [1] independent of a limit on the failure of the author to profit. In Refs. 8 and 9, this idea is pursued, and a system is constructed which observes whether a dishonest act has occurred. Namely, we say that a copyright protection scheme is *passive* if it is concerned with identifying dishonesty. However, in such a method, when a signature has been inserted into the distributed image data on a document, the image data may freely be reproduced and the perpetrator cannot be identified. The methods of Refs. 10 and 11 can prevent dishonest copying more actively, but are not able to prevent the distribution of the image data. In Ref. 12 this problem is solved by distributing image data of inferior quality, the images having been intentionally scrambled, but, because of the poor quality of the original images, the user cannot be identified. Since this viewpoint is similar to ours, in this paper, by a partial scramble we implement active copyright protection by adding a signature which identifies the document.

Since ancient times, for pictures, handwriting, and engravings, people have made carved seals and signatures which identify the author. Not only did this prove their authorship, but in addition the existence of a signature added to the value of the work, thus ensuring a profit for the author. From this point of view, in the application to digital image data, the location of the signature and its method of inclusion are problems. First, the signature cannot be deleted by the user alone, and, when it is deleted, the original image data must be restored to the same location. Therefore, it is preferable to construct the signature at an arbitrary position. Also, when the receiver has the proper authorization from the author, in order to be able to restore the original image data it is necessary to preserve in advance the information contained by the pixels in the signature part of the distributed image data. We consider similar issues in the next section, describing a specific signature implementation method.

### 2.2. The signature method

First, we discuss the terminology and variables used in this paper (see Fig. 1). The original image data form a $w_0 \times h_0$ set of color or black-and-white pixels, and we assume that the signature data form a 2-valued $w_s \times h_s$ array of pixels. Here $w_0 \geq w_s$ and $h_0 \geq h_s$. Therefore, the actual signature is embedded at a *signature region location*. Let $(x_0, y_0)$ be the coordinates at its upper left corner. Thus, we assume that $x_0 \leq w_0 - w_s$ and $y_0 \leq h_0 - h_s$. Also, the section where the signature data are constructed is called the *signature part.*

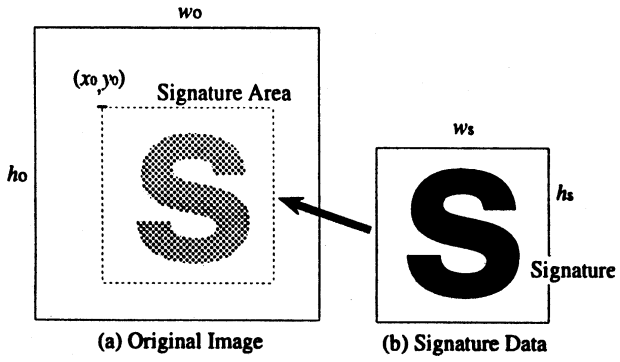



Fig. 1. Definition of terms and variables.

The author or his associates can reconstruct the signature data of arbitrary size representing his own signature and the 2-valued logmark. Next, choose $x_0$ and $y_0$ arbitrarily for the signature region. We will take the XOR with random numbers only in the section which corresponds to the signature region of the original image data. The section which corresponds to the signature region of the original image data resembles noise, and as a result the signature of the original image data is apparent. However, in this method, in restoring the original image data, the random numbers which perform the signature on the image data are destroyed, so it becomes necessary to consider the key for the signature data as well as $x_0$ and $y_0$. Thus, the efficiency of key delivery is not bad, and the indicating signature is easily deduced from the signature data. Therefore, while the scanning order of the signature part is arbitrarily selected by the author, it is desirable that the scanning order be constructed in random fashion. Similarly, the scanning method for the signature part in this paper is determined arbitrarily by a technique extending the generalized Peano scan, which increases security. The details are provided in the next section. This technique is applied in several rectangular regions of differing sizes. Within each particular region, the scanning order is determined by a scanning key, and the scanning order of the signature part can be kept a secret between the author and the users who have received the key.

## 3. Scrambling with the Generalized Peano Scan

### 3.1. Extension of the generalized Peano scan

The Peano scan is a technique for scanning along a continuous curve which passes through every point of the image data exactly once; its practical application is to scanning image data which are scattered along the Peano curve. Research on the Peano scan, which has been active since 1890, has been stimulated from various viewpoints. In recent years, there have been improvements in the scanning algorithm [14], larger dimensions [15, 16], generalizations of the size and shape [16, 17], practical applications of image data compression using various methods [19], and practical applications to image data scrambling [15, 18].

The generalized Peano scan [16] is obtained from the Peano scan by iteratively partitioning arbitrary rectangular regions, and generalizing the Peano scan to arbitrary rectangular regions. Its distinctive characteristic is that it can partition whatever regions are necessary, since the number of its partitions is not always precisely uniform. We introduce three types of partitions, binary through quaternary, but they are not different in theory. If we use these scanning methods, the scanning order on the signature part can be determined. We now explain the generalized Peano scan. First, let $c$ denote a cell in a rectangular region of the scan. The cell can be classified by its scan direction and by the parities of its lengths in all directions, and we represent this by $c(w, h, u, v)$. Here $w$ and $h$ denote the width and height of the cell, and $(u, v)$ is the vector denoting the scan direction. Next, we partition this cell recursively, using the following methods.

a. Recurrent binary partition procedure

$$
\begin{aligned}
c(even,even,1,0) &\rightarrow c(odd,even,1,1)+c(odd,even,1,-1)\\
c(even,odd,1,0) &\rightarrow c(odd,odd,1,1)+c(odd,odd,1,-1)\\
&\rightarrow c(even,odd,1,1)+c(even,odd,1,-1)\\
c(odd,even,1,1) &\rightarrow c(even,even,1,0)+c(odd,even,1,1)\\
c(odd,odd,1,1) &\rightarrow c(even,odd,1,0)+c(odd,odd,1,1)\\
c(even,odd,1,1) &\rightarrow c(even,odd,1,0)+c(even,odd,1,1)
\end{aligned}
$$

b. Recurrent ternary partition procedure

$$
\begin{aligned}
&c(even,odd,1,1)\\
&\quad\rightarrow c(even,odd,1,1)+c(even,odd,1,-1)+c(even,odd,1,1)\\
&\quad\rightarrow c(odd,odd,1,1)+c(odd,odd,1,-1)+c(even,odd,1,1)\\
&c(odd,odd,1,1)\\
&\quad\rightarrow c(odd,odd,1,1)+c(odd,odd,1,-1)+c(odd,odd,1,1)\\
&\quad\rightarrow c(odd,odd,1,1)+c(even,odd,1,-1)+c(even,odd,1,1)\\
&c(odd,even,1,1)\\
&\quad\rightarrow c(odd,even,1,1)+c(odd,even,1,-1)+c(odd,even,1,1)
\end{aligned}
$$

c. Recurrent quaternary partition procedure

$$
\begin{aligned}
c(odd,even,1,0) &\rightarrow c(even,even,0,1)+c(even,odd,1,0)\\
&\quad + c(even,odd,1,0)+c(even,even,0,-1)\\
c(even,even,1,0) &\rightarrow c(even,even,0,1)+c(even,even,1,0)\\
&\quad + c(even,even,1,0)+c(even,even,0,-1)
\end{aligned}
$$

Here *odd* denotes a suitable odd number and *even* an even number; $\rightarrow$ denotes the result of a partitioning procedure, and + denotes the adjunction of a starting point of a cell to an already existing corner of a cell in the partitioning. In this procedure, if $w < h$ we may transpose the height and width dimensions of the cell, so we may assume that the cell is not long and slender. However, we exclude cells which are impossible to transpose, such as $c(even, even, 1, 0)$ and $c(even, odd, 1, 0)$. Examples of the resulting scans are shown in Fig. 2.

We can increase the security even more if we further extend the generalized Peano scan and can conceal the key used in the scanning method of the region. We now state the details. In the above recurrent partitioning procedure, we consider combinations of four types:

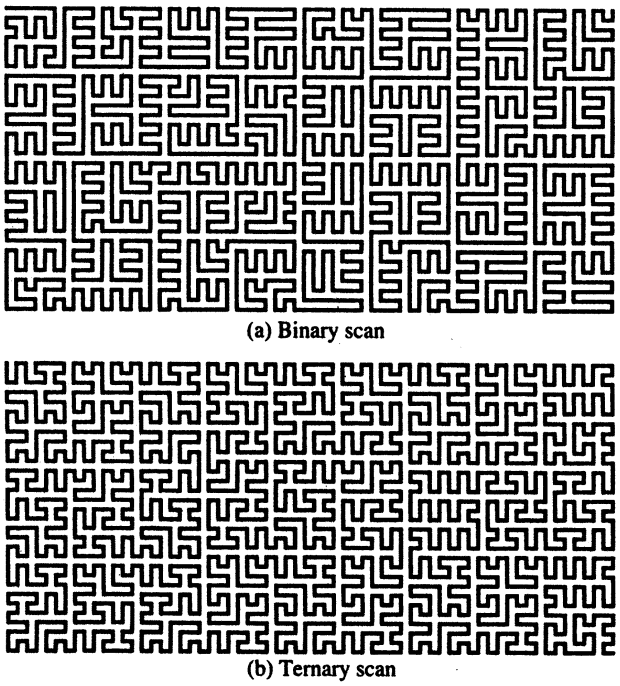



Fig. 2. Examples of generalized Peano scan.

$$even \rightarrow even+even \quad (1)$$

$$even \rightarrow odd+odd \quad (2)$$

$$odd \rightarrow odd+even \quad (3)$$

$$odd \rightarrow even+odd \quad (4)$$

so a large number of partitioning methods exist. For example, if we want to use (2) to partition and the width is 7, then three types of partitioning method are possible: 2 + 5, 4 + 3, and 6 + 1, as shown in Fig. 3. For the partitioning of the

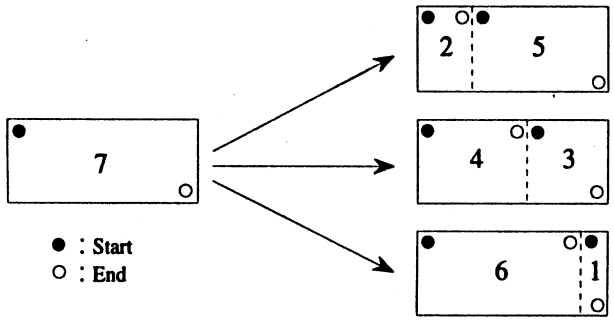



Fig. 3. Types of division.

cell, we could use any of three types of scan. Using this property and the scanning order of the cells, the author can choose arbitrarily. We now describe the decision method. In the case of binary partitioning, the partitioning method differs according to the quotient and the remainder obtained when we divide the width of the cell we want to partition by 2. Similarly, in the case of ternary partitioning, the partitioning method differs according to the quotient and the remainder obtained when we divide the width of the cell we want to partition by 3. In particular, in each case, we make a table of partitioning rules in advance, prepare several partitioning methods, and store the partitioning rules table. Then when the author arbitrarily chooses his scanning key, he can choose his scanning order uniformly from the partitioning rules of the partitioning rules table. Tables 1 and 2 give examples of the partitioning rules table for binary and ternary partitions. Here $m$ is a natural number satisfying the conditions of the table. In our example of an actual use of the above method, for $w$ = 7, the relevant cell is

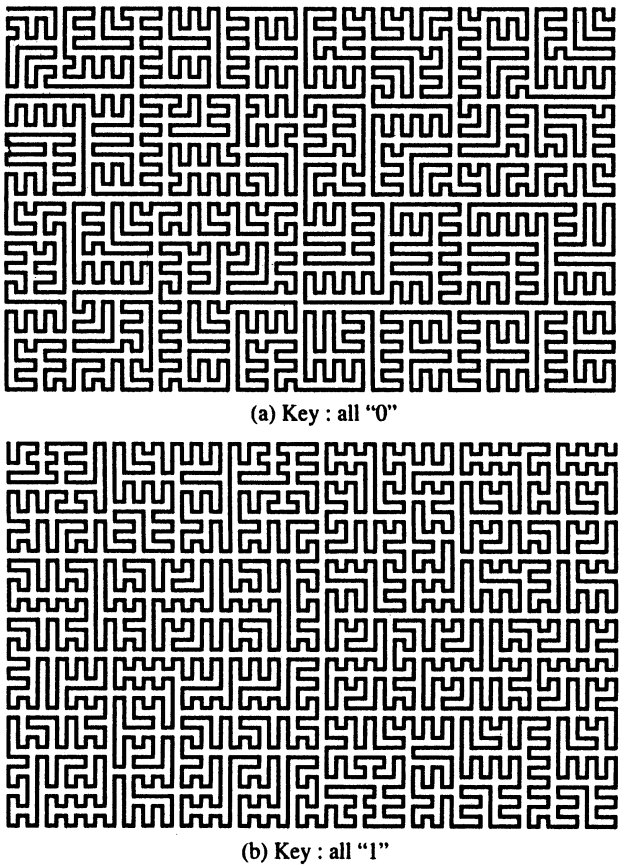
Table 1. Dividing rule list (binary division)

| Cell | | | Partitioning | |
|---|---|---|---|---|
| Cell type | $\lfloor width/2 \rfloor$ | $width \pmod 2$ | Key value = 0 | Key value = 1 |
| c(even,even,1,0) | even | 0 | 2m-1 : 2m+1 | 2m+1 : 2m-1 |
| | odd | 0 | 2m+1 : 2m+1 | 2m+3 : 2m-1 |
| c(even,odd,1,0) | even | 0 | 2m-1 : 2m+1 | 2m+1 : 2m-1 |
| | odd | 0 | 2m : 2m+2 | 2m+2 : 2m |
| c(odd,even,1,1) | even | 1 | 2m : 2m+1 | 2m+2 : 2m-1 |
| | odd | 1 | 2m : 2m+3 | 2m+2 : 2m+1 |
| c(odd,odd,1,1) | even | 1 | 2m : 2m+1 | 2m+2 : 2m-1 |
| | odd | 1 | 2m : 2m+3 | 2m+2 : 2m+1 |
| c(even,odd,1,1) | even | 0 | 2m : 2m | 2m+2 : 2m-2 |
| | odd | 0 | 2m : 2m+2 | 2m+2 : 2m |

Table 2. Dividing rule list (ternary division)

| Cell | | | Partitioning | |
|---|---|---|---|---|
| Cell type | $\llcorner width/3 \lrcorner$ | $width \pmod 3$ | Key value = 0 | Key value = 1 |
| $c(even,odd,1,1)$ | *even* | 0 | $2m : 2m : 2m$ | $2m+1 : 2m-1 : 2m$ |
| | *odd* | 1 | $2m+1 : 2m+1 : 2m+2$ | $2m+2 : 2m : 2m+2$ |
| | *even* | 2 | $2m+1 : 2m+1 : 2m$ | $2m+2 : 2m : 2m$ |
| $c(odd,odd,1,1)$ | *odd* | 0 | $2m+1 : 2m+1 : 2m+1$ | $2m+1 : 2m : 2m+2$ |
| | *even* | 1 | $2m+1 : 2m : 2m$ | $2m+1 : 2m+1 : 2m-1$ |
| | *odd* | 2 | $2m+1 : 2m+2 : 2m+2$ | $2m+1 : 2m+3 : 2m+1$ |
| $c(odd,even,1,1)$ | *odd* | 0 | $2m+1 : 2m+1 : 2m+1$ | $2m+1 : 2m-1 : 2m+3$ |
| | *even* | 1 | $2m+1 : 2m-1 : 2m+1$ | $2m+1 : 2m+1 : 2m-1$ |
| | *odd* | 2 | $2m+1 : 2m+1 : 2m+3$ | $2m+1 : 2m+3 : 2m+1$ |

$c(odd, even, 1, 1)$ and $m = 1$: if the key value is 0, the partition is 2 + 5; if the key value is 1, the partition is 3 + 4. In Fig. 4 we describe two Peano scans created using different keys. This scanning method makes it possible to conceal



(a) Key : all "0"

(b) Key : all "1"

Fig. 4. Peano scan created by different keys.

the scanning order, so that the method described in the next section can be applied to the scan of the signature part.

### 3.2. Scanning method of the signature part

The scanning method described in the previous section can scan arbitrary cells, but cannot scan completely because of the presence of the logmark and the signature of the signature part. However, if we regard a set of several cells of the signature part as constituting a complex form, then scanning becomes possible. This justifies the partitioning process of the signature part. We call this procedure *regional partitioning*. If the protection provided by the scanning key of the cell is sufficiently strong, it is important to partition the largest possible cells. The actual procedure is carried out by exhaustive search. In Fig. 5 we show an example of the result of regional partitioning.

In this process, the standard output is the length in all directions and the relative coordinates $(x_i, y_i)$ of each cell in the partition, and we use the key to remove the signature. It is possible in the scan of the signature part to combine the scanning method of the previous section with the regional partitioning process. In Fig. 6 we illustrate the result of a scan using this technique. In the next section, we merge the combined procedure with our proposed technique.

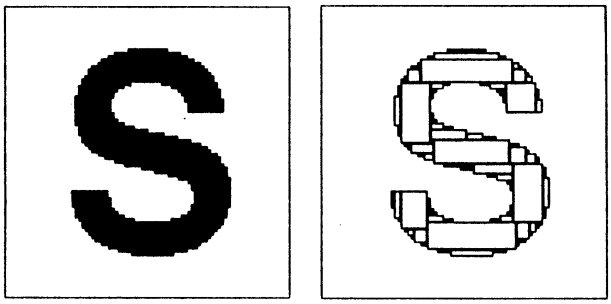
### 3.3. Signatures and the reproduction procedure

Combining the variable key with the proposed technique, we proceed as follows.

**a**. $(x_0, y_0)$: the header address of the signature region

**b**. $(x_i, y_i)$: the relative address of a cell

**c**. $c_i(w_i, h_i, u_i, v_i)$: the type and size of the cell



(a) Signature Data (b) divided areas

Fig. 5. Area dividing.

**d**. $k_i$: the scanning key of the cell

**e**. *key*: the keys generated by the random numbers, for $i = 1, 2, \ldots$.

[the signature procedure (binary partition)]

Step 1: prepare the signature data for the original image data

Step 2: the regional partitioning procedure

2-1 $i \leftarrow 1$

2-2 search the cell for the largest possible increase in the signature part

2-3 output $x_i$, $y_i$, and $c_i(w_i, h_i, u_i, v_i)$

2-4 delete $c_i$ from the signature part

2-5 $i \leftarrow i + 1$, GOTO 2-2

Step 3: the signature procedure

3-1 input $x_0$, $y_0$, and the key

3-2 $i \leftarrow 1$

3-3 read $x_i$, $y_i$, and $w_i$, $h_i$, $u_i$, $v_i$ and input $k_i$

3-4 $w \leftarrow w_i$, $h \leftarrow h_i$

3-5 IF $w < h$ and transposition is possible THEN transpose the cells

compute $m$ from $w$, $k_i$, and the partition rules table;

IF partition is impossible THEN generate 8 bits randomly and XOR the bits with the pixel value

ELSE $w \leftarrow w'$ (the width of the cell on the left side); GOTO 3-5 (recursion)

$w \leftarrow w''$ (the width of the cell on the right side); GOTO 3-5 (recursion)

3-6 $i \leftarrow i + 1$, GOTO 3-3

3-7 the signature is complete; output the image data

[the procedure for reproduction of the original image data]

Step 1: receive the image data and purchase the key.

Step 2: perform step 3 of the signature procedure; output the original image data.

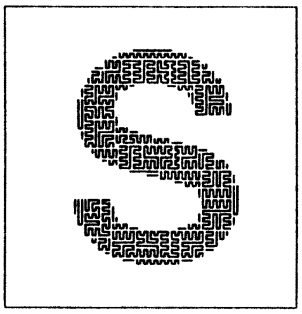
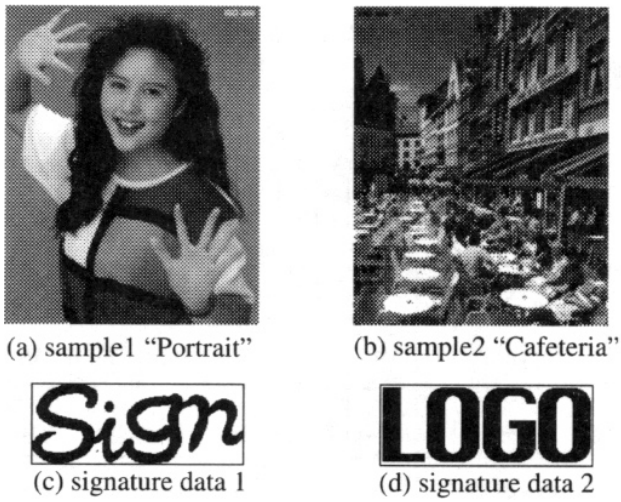


Fig. 6. Scanning sequence for signature.

## 4. Experimental Results

We performed experiments, applying this technique to actual image data. The original image data used in the experiment and the signature data are shown in Fig. 7. Figures 7(a) and 7(b) are reduced and transformed from the standard image data published by the Japan Standards Society, 256 × 320 pixels, at 8 bits per pixel. The signature data of Figs. 7(c) and 7(d) are 210 × 80 pixels at 1 bit per pixel.

First, we perform the regional partitioning procedure on the signature data of Figs. 7(c) and 7(d) and generate a copy of the key. The results are shown in Fig. 8. The results of applying the signatures to Fig. 7 are shown in Fig. 9. The leading address of the signature region is at $(x_0, y_0) = (23, 160)$, and we have used an all-zero scanning




(a) sample1 "Portrait" (b) sample2 "Cafeteria"

(c) signature data 1 (d) signature data 2

Fig. 7. Original images and signature data.

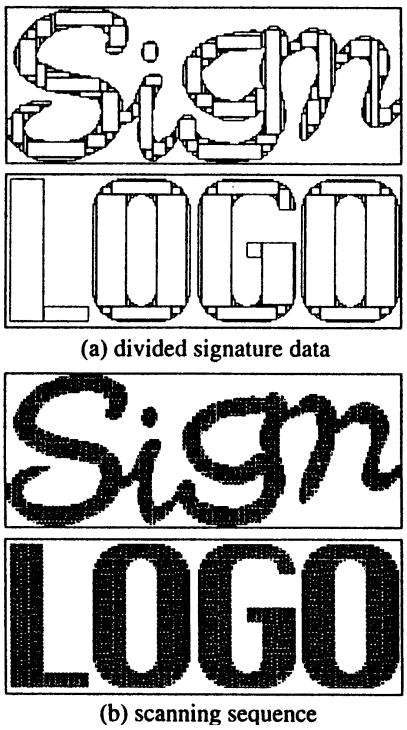



Fig. 8. Divided signature data and scanning sequence.

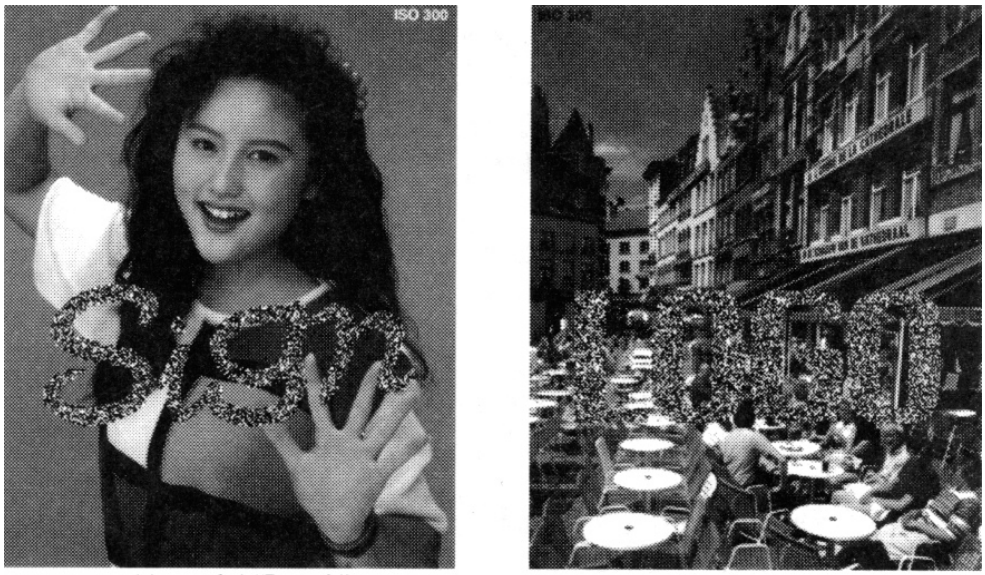
key. From these results we see that it is possible to voluntarily display the signature of the image data to the user. The appearance of the signature also plays the role of leaving the image data almost completely intelligible.

## 5. Evaluation and Considerations

### 5.1. Evaluating the strength of the signature

This section considers the situation in which a user who receives the signed image data purchases the key and tries to remove the signature. We assume that this technique is thoroughly understood by the user, and that all procedures can be reproduced. If the procedure is robust, all of the key must be determined. Since there is a danger of deducing the variables $x_0$, $y_0$, $x_i$, $y_i$, $w_i$, $h_i$ which are used as key in this technique from the distributed signed image data, we here discuss the subject of removal. Ultimately, if it is not possible to deduce the actual variables, we could try to recover a close approximation. Thus, we consider the strength of the effect of the key $k_i$ on each variable $u_i$, $v_i$. There are only eight values taken on by the variables $u_i$ and $v_i$: (1, 0), (−1, 0), (0, 1), (0, −1), (1, 1), (1, −1), (−1, 1), and (−1, −1), and we assume that the probability of each is 1/8. The choices for $k_i$ vary, depending on the size of each cell and the number of times the cell has been recursively partitioned. If this number of times is $n_i$, then the total variation is $2^{n_i}$. The probability *Prob* that the key used in generating the random number is guessed by chance using this technique is therefore

$$Prob = \frac{1}{8} \times 2^{-\Sigma_i^{n_i}} \times p_{key}$$




(a) sample1 “Portrait” (b) sample2 “Cafeteria”

Fig. 9. Signed images.

Thus, the strength of the proposed signature system depends on the sum of the $n_i$. In other words, the larger the number of cells constituting the signature part, the more it is preferred. As an example, we show the strength of the signature data used in our experiments (sum of the $n_i$):

a. the strength in Fig. 7(c) is $\Sigma n_i = 996$
b. the strength in Fig. 7(d) is $\Sigma n_i = 1425$.

The smaller the numerical values of the dimensions of the signature data in Fig. 7(c), the weaker the strength is. Though the signature part in Fig. 7(c) is long and slender, it is obviously complex. In the discussion of whether this probability is adequate, we should decide by comparing the cost of purchasing the key with the cost of recovering the agreed key.

### 5.2. Evaluating the effect of coding

Next we discuss the effect on the receiver of various kinds of coding of the image data on which the signature technique is performed. In our technique, because scrambling is accomplished using random numbers, if the bit sequence is changed by coding, there is the possibility of failure in the removal of the signature. Conversely, the compression of the associated coding is not entirely effective unless there is good decoding. We consider the remarkable example known as JPEG compression [20]. In the JPEG algorithm, the values in a quantization table produce different compression ratios. Thus, it is easy to change the pixel values if the quantization is coarse, depending on the rapidity of the change of values in the image data. Therefore, we estimate the rapidity of change in the signature part. It follows that the signature cannot be removed completely. By evaluating this quantitatively, we see that the bounds on the noise which remains depend on the quantization table values. The quantization table used is shown in Fig. 10. The values in this quantization table are changed by $1/t$ by coding and decoding; the image data which remain after the removal of the signature, including the limited noise, are shown in Fig. 11. Here the parameter $t$ determined

$$SNR = 20 \times \log_{10}\left(\frac{255}{err}\right) \tag{6}$$

by the quantization table has a direct effect on the compression ratio. It is easy to express the combination of the SN ratio and the compression; in the unsigned case,
where

$$err = \sqrt{\frac{1}{w_o \times h_o}\sum_{i=1}^{w_o \times h_o}\left(org_i - jpg_i\right)^2} \tag{7}$$

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 16 | 11 | 10 | 16 | 24 | 40 | 51 | 61 |
| 12 | 12 | 14 | 19 | 26 | 58 | 60 | 55 |
| 14 | 13 | 16 | 24 | 40 | 57 | 69 | 56 |
| 14 | 17 | 22 | 29 | 51 | 87 | 80 | 62 |
| 18 | 22 | 37 | 56 | 68 | 109 | 103 | 77 |
| 24 | 35 | 55 | 64 | 81 | 104 | 113 | 92 |
| 49 | 64 | 78 | 87 | 103 | 121 | 120 | 101 |
| 72 | 92 | 95 | 98 | 112 | 100 | 103 | 99 |

Fig. 10. Quantizing table for JPEG.

Here the $org_i$ are the pixel values of the original image data and the $jpg_i$ are the pixel values after the encoding and decoding. As we can see in Fig. 11, if the size of the quantization step is small, the noise generation can be controlled, causing a simultaneous decrease in the compression ratio. If the bits in the neighborhood of the most significant bit are changed, considerable noise may appear. This point will need improvement in the future. To further enhance our design, we would like to find a way to modify the pixel values in the algorithm which removes the signature.

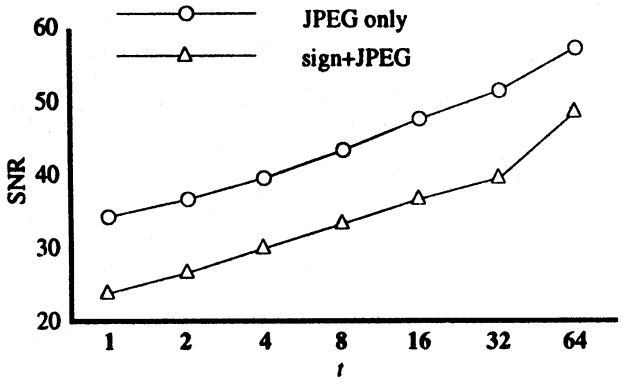



Fig. 11. SN ratio for $t$.

### 5.3. Application to color image data

In this paper we have considered only original image data which are black and white, but exactly the same procedure can be applied to colored image data. For example, these techniques could be applied to the brightness values of the colored image data; results almost identical to the case of black-and-white image data would be obtained.

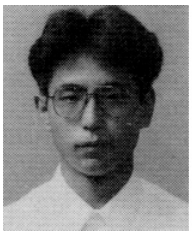
## 6. Conclusion

In this paper, we have described, from the point of view of active copyright protection, a signature method for the copyright protection of image data which, as its specific method of implementation, employs partial scrambling using the generalized Peano scan. Since, using this technique, the signature for the disseminated image data is always apparent, it is possible to identify the author. Also, in contrast to the case in which all of the image data are scrambled, we can easily verify the general form of the image data. Furthermore, a receiver who has purchased the legitimate key from the author is able to remove the signature. In such a way we could construct a system with cash payments, which must be legalized, in which image data with a signature inserted are reproduced and distributed. But there is a problem. In this paper we have not discussed the situation in which a user who has received the legitimate key collaborates with dishonest users to reproduce the image data with the signature removed. Our technique is unable to deal with this case. Therefore, another technique to act as a countermeasure will probably need to be used simultaneously. In the future, research will be necessary to construct a practical system, one which can deal with the inferior image quality which results from compression.

## REFERENCES


1. Handa M. A copyright system approach to turning points. Tsubu; 1994. (in Japanese)
2. Nakayama S. Multimedia and copyright. Iwanami; 1996. (in Japanese)
3. Kitamura M. A system constructed for copyrighting. Inf Proc Res Rep 95, 1995(37);AVM-8:129–134. (in Japanese)
4. Takashima Y, Ishii S, Yamanaka K. A copyright protection system using PCMCIA cards. SCIS95 1995, B5.5. (in Japanese)
5. Matsuya H, Inaba, Wakasugi K, Kasahara M. A method for the construction of a copyright protection function for image data in documents. Inf Theory Results, ISEC94-58, 1995. (in Japanese)
6. Koike H, Matsumoto T, Imai H. A copyright protection system for digital image data. SCIS93-13C, 1993. (in Japanese)
7. Komatsu N, Tominaga H. A proposal for digital watermarks in the communication of document image data and its practical application to signatures. Inf Theory Pap 1989;J72-B-I:208–218. (in Japanese)
8. Matsui K. The deep-level coding of image data. Morikita; 1993. (in Japanese)
9. Oka K, Nakamura Y, Matsui K. Embedding signatures in image data hardcopy using density-pattern methods. Inf Theory Pap 1996;J79-D-II:1624–1626. (in Japanese)
10. Matsumoto T. An approach to protection based on the coding of digital documents. Inf Theory Results ISEC95-27, 1995. (in Japanese)
11. Matsumoto T, Uchiyama H, Koyama T. An approach to protection based on the coding of digital documents (II). SCIS96-9D, 1996. (in Japanese)
12. Fujii H, Yamanaka Y. A system for the scrambling of digital image data for copyright protection. SCIS96-9A, 1996. (in Japanese)
13. Suzuoki M, Watanabe H, Take T. A technique for the copyright protection of digital image data and a technique for identifying users who copy dishonestly. Inf Theory Results ISEC95-47, 1996. (in Japanese)
14. Kamata S, Eason RO, Kawaguchi E. An implementation of a Hilbert scanning algorithm and its application to data compression. Trans IEICE Inf Syst 1993;E76-D:420–428.
15. Stevens RJ, Lehar AF, Preston FH. Manipulation and presentation of multidimensional image data using the Peano scan. IEEE Trans PAMI 1983;5:520–526.
16. Nagae T, Yasui I, Nagahashi H. The generalized Peano scan and its practical application to a halftone process. TV Inst Results 16 1992;16:25–30. (in Japanese)
17. Agui T, Nagae T, Nakajima M. Generalized Peano scans for arbitrarily sized arrays. Trans IEICE 1991;E74:1337–1342.
18. Kamata S. A study of light-and-dark image data compression using Hilbert curves. Inf Theory Results IE95-123, 1996. (in Japanese)
19. Komiya Y, Komatsu N, Kudoh H. An image data scrambling technique using SFC. PCSJ93 1993(4-2): 77–78. (in Japanese)
20. Image Data Electronics Institute edition. New handbook of image data electronics. Corona; 1993. (in Japanese)

## **AUTHORS** (from left to right)



**Kazuhiro Oka** (student member) graduated in 1992 from the National Defense Academy in electrical engineering. Currently he is studying operations research in the Theoretical Engineering Research Department of the same university. His principal research interest is in the copyright protection of image data.

**Kineo Matsui** (member) graduated in 1961 from the National Defense Academy in electrical engineering. In 1965 he did special research in electronics engineering at Kyushu University. In 1981 he became a professor in the Electrical Engineering Department of the National Defense Academy, and in 1989 a professor in the Information Engineering Department. His principal research interests include coding theory, the coding of image data, and information security. He is the author of a book, *The Deep-Level Coding of Image Data*, published by Morikita. He is a member of the Information Processing Society, the Image Data Electronics Society, and the TV Society of Japan.