Vol. 39 No. 2 Transactions of Information Processing Society of Japan Feb. 1998

Regular Paper

Shadow Codes for Representation of Binary Visual Patterns

JULIO TANOMARU,[†] ATSUSHI INUBUSHI[††] and YOSHIZO TAKAHASHI[†]

In this paper, a novel approach to the representation of binary visual patterns is proposed, and the applicability of the method to recognition of handwritten patterns by neural network and conventional classifiers is investigated. The proposed approach has been named shadow codes, because it is based on shadow projections of pixels of the thinned input image onto the bars of a frame surrounding the image. A number of variations of the method can be devised, and the case in which the region of attention consists of a rectangle with orientation given by the principal axes of inertia of the input image is considered in detail. A frame composed by 16 bars classified into three categories is superposed on the attention region containing the thinned input image, and each pixel projects a shadow on the nearest bar of each category. While the determination of the attention region is inherently a translation-invariant process, scaling invariance is achieved by normalizing and quantizing the shadow lengths, resulting in a low-dimensional shadow vector. For a task consisting of the recognition of handwritten numerical characters using both a neural net, namely, a self-organizing map fine-tuned by learning vector quantization, and a conventional classifier, high recognition rates were obtained, confirming the effectiveness of the proposed representation method. Also, comparison with other graphical feature extraction techniques yielding feature vectors of the same dimension indicates that, although compact, shadow codes succeed in preserving information that can be used for recognition.

1. Introduction

Feature extraction and selection are central issues in any pattern recognition system, and this statement is particularly true for patterns of interest in the computer vision field [20]. In this field, practical input patterns are invariably characterized by large amounts of data, usually expressed in the form of rectangular arrays of pixel values, which are either binary, gray-scale, or color vectors. Since the most direct approach of employing raw pixel vectors for recognition is not efficient for large images, there have been many efforts to develop effective, compact data representations for image patterns [4),7)].

A special case of great practical importance concerns apparently simple visual patterns such as alphanumerical characters [13]. For such patterns, a good representation scheme should result in a compact data set, rich enough to allow reliable recognition. It is also desirable that the representation scheme be as robust as possible to variations in the pattern with respect to noise, deformation, thickness, position in the visual field, scale (size), orientation, and so on, to facilitate the task of classification, which is the final stage of the overall recognition system. Furthermore, ideally, the representation scheme should require only simple and quick computations, allowing for efficient and cost-effective real-time implementations on ordinary hardware.

This paper proposes a novel approach to the representation of binary visual patterns in general, and of handwritten characters in particular, for the purpose of pattern recognition. The technical literature presents a number of approaches involving the representation of handwritten characters for either on-line [3),9),24),25),29)] or off-line [8),10),14),19),22),30)] recognition problems. In on-line problems, the existence of a transducer able to codify the dynamical information of the writing as it takes place, typically a tablet digitizer [27], is assumed, whereas in off-line problems the writing is represented after it has been completed, and only static information is available. The problems considered in this paper pertain to the second category.

The basic idea introduced here is dubbed *shadow codes* (SCs), which are feature vectors derived from shadow projections of pixels of the thinned input pattern onto bars of a frame surrounding the pattern inside a compact attention region. The approach requires neither computationally-intensive use of transforms derived from the digital signal processing field [2] nor complex segmentation procedures for pos-

† Faculty of Engineering, The University of Tokushima

†† Systems Engineering Group, Fujitsu Limited

terior structural recognition [5),21),26)], and is totally based on geometrical considerations. It is shown in detail how a simple rectangular frame with orientation obtained from the principal axes of inertia of the input pattern can be used to generate a 16-component SC vector for binary visual patterns. Besides being simple to implement and fast to execute, the proposed representation method is inherently invariant to translation and scaling of the input pattern, and constrains the uncertainty about the orientation of the original pattern to rotation angles that are multiples of 90 degrees.

To verify the effectiveness of the proposed method, experiments in the recognition of handwritten digits were conducted, using neural networks and a simple nearest-neighbor method for the sake of comparison. Results confirmed that, despite its compactness, a 16-component SC retains enough information to allow high recognition rates for simple patterns such as handwritten digits, and also demonstrated the superiority of neural networks for recognizing patterns with great diversity.

The structure of the remainder of this paper is as follows. In Section 2, the concept of shadow codes is introduced, and it is shown in detail how to generate 16-component SCs for binary patterns. Section 3 discusses the invariance properties of SCs, while other characteristics are presented in Section 4. An experimental study of the recognition of handwritten digits by means of neural networks is presented in Section 5, with the results given in Section 6. Finally, Section 7 concludes this paper.

2. Generation of Shadow Codes

Consider an input image pattern represented by binary pixel values in a rectangular grid. It is assumed here that the pattern has been thinned by any of the many methods in the literature [12),31)]. It is also assumed that there is only one input pattern of interest inside the visual field, that is, that auxiliary image partition procedures (external segmentation) have already taken place [18)].

The generation of the SCs for any given binary image basically consists of the following steps: (1) determination of a compact attention region of a specified shape containing the input pattern; (2) superposition of a virtual frame of thin bars on the attention region; (3) determination of the nearest bars for each pixel, and projection of pixel "shadows" on the corresponding bars; and (4) conversion of the shadow features in a real-component vector. Each of these steps is explained below.

2.1 Determination of the Attention Region

The first step in the generation of SCs consists of determining a compact subregion of the visual field containing the input pattern to be represented. The subregion has a shape specified *a priori*, and the procedure for determining it may be thought of as a rudimentary attention mechanism. For this reason, the subregion is called the *attention region*. The simplest shape for the attention region is a rectangular one, which seems to be a natural choice when dealing with simple patterns such as handwritten characters. This assertion is supported by the fact that ZIP codes, as well as data to be entered in document forms, are often required to be written inside rectangular boxes. For the same reason, manuscript forms for oriental languages are usually divided into boxes, and LED and crystal-liquid displays commonly have a rectangular matrix for each character.

2.1.1 Euclidean Rectangular Region

For an attention region with a rectangular shape, the most straightforward choice would probably be the smallest rectangle involving the pattern, with the orientation given by the Euclidean axes. To determine such a rectangle, one needs only to scan the input image for the highest and lowest coordinates of the black pixels.

Despite the simplicity of the approach, choosing the rectangle oriented according to the coordinate axes does not seem to be a good strategy, since it is based on the arbitrary choice of the orientation of the axes themselves, whereas the characteristics of the pattern itself, such as its centroid and pixel distribution, are ignored. An example of a rectangular frame obtained by this approach is given in **Fig. 1** (a).

2.1.2 Inertial Rectangular Region

A better (less biased) choice for the rectangular attention region is to select the smallest rectangle surrounding the pattern, with the orientation determined by the pattern's principal axes of inertia. Consider a binary image represented by its pixels $f(x,y) \in \{0,1\}$. The moment of order $(p+q)$ [15)] is defined as

$$m_{pq} = \sum_x \sum_y x^p y^q f(x,y). \qquad (1)$$

If the coordinates are computed with respect to

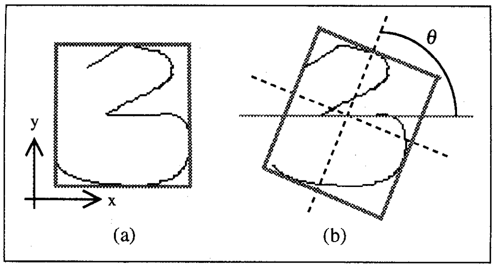


Fig. 1 Rectangular attention region surrounding the input pattern. (a) Euclidean orientation, (b) rectangle based on the principal axes of inertia.

the centroid of the image, the equation above results in the central moment of order $(p+q)$, that is,

$$M_{pq} = \sum_x \sum_y (x - x_C)^p (y - y_C)^q f(x, y), \tag{2}$$

where (x_C, y_C) are the coordinates of the centroid.

With respect to an axis passing through its centroid and making an angle θ with the x-axis, the corresponding central moment of inertia is given by

$$M_\theta = \sum_x \sum_y f(x, y)(y \cos\theta - x \sin\theta)^2. \tag{3}$$

The angle θ that minimizes (maximizes) the moment of inertia M_θ gives the orientation of the principal axes of inertia of the image. From Eqs. (1) and (3), it can be shown that

$$M_\theta = m_{02} \cos^2\theta + m_{20} \sin^2\theta - m_{11} \sin 2\theta. \tag{4}$$

Solving for $\partial M_\theta / \partial \theta = 0$ results in

$$\theta = \frac{1}{2} \tan^{-1}\left(\frac{2m_{11}}{m_{20} - m_{02}}\right) \pm \frac{n\pi}{2}, \tag{5}$$

where $n \in Z$. After determining the principal axes of inertia, the coordinates of the vertices of the rectangular attention region can be easily found by scanning the image pixels. This approach is shown in Fig. 1 (b), and is a more impartial choice than the rectangle oriented according to the Euclidean axes.

2.2 The Shadow Bar Frame

Having determined the rectangular attention region, the next step is to superpose on it a frame consisting of thin, virtual bars. These bars are called *shadow bars* because, as explained below, shadows of the pixels are to be projected onto them, and are classified into a few categories, here called bar groups. It is assumed that the frame is shaped, sized, and oriented in such a way as to match the attention region perfectly.

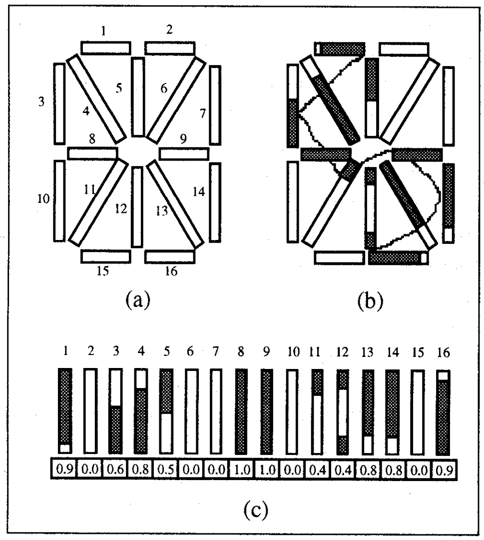


Fig. 2 Example of generation of the shadow vector of a binary pattern. (a) 16-bar frame, (b) a binary pattern and corresponding shadows, (c) normalized shadow vector.

For a rectangular attention region, a shadow bar frame that has produced encouraging results is the one illustrated in **Fig. 2** (a), in which there are 16 straight bars classified into three groups: horizontal, vertical, and diagonal ones. For the sake of visualization, the bars in the figure are wide and do not touch, but in fact they have zero width and do touch in the concurrent points. That is, for instance, the bars b_1, b_3, and b_4 intersect at the top-left corner, and so on. The equations of the segments defining each bar can be easily obtained from the coordinates of the attention region.

2.3 Shadow Projections

In the following step, the nearest bar of each group is first determined for each (black) pixel of the input pattern in the attention region. Next, each pixel projects perpendicular shadows on each of the those bars. For the 16-bar frame described above, each pixel must project a total of three shadows, on the nearest horizontal, vertical, and diagonal bars.

An example of shadow projection is shown in Fig. 2 (b), where the shadows are represented as the darkened parts of the bars, resulting in a pattern of dark and white parts for each bar. It

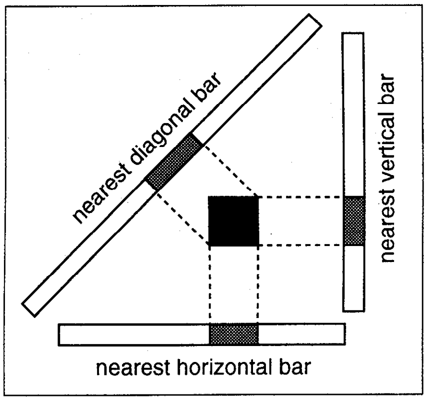
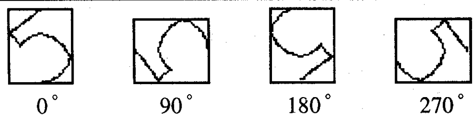


Fig. 3 The lengths of shadows of a given pixel are not discrete, but continuous values.



0°	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
90°	10	3	15	11	8	4	1	12	5	16	13	9	6	2	14	7
180°	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
270°	7	14	2	6	9	13	16	5	12	1	4	8	11	15	3	10

Fig. 4 Shadow codes for rotations multiple of 90° can be easily obtained by simply reordering the components of the code vector. For simplicity, only the bar labels are shown.

is important to note that although the number of pixels in the input pixels is a discrete quantity, the lengths of the shadows of each pixel are continuous values. That is, depending on the projection angle, the length of a pixel shadow varies, as illustrated in **Fig. 3**.

2.4 Conversion into Shadow Codes

In the final stage of SC generation, the shadow patterns of the bars are converted into real numbers, resulting in a vector. For this, a number of approaches could be devised.

A simple coding procedure is to measure the total length of the shadowed part of each bar, regardless of whether or not the parts are contiguous, and divide the result by the length of the corresponding bar. This gives a SC vector with 16 real-valued components between 0.0 and 1.0, as shown in Fig. 2 (c), where the lengths of the bars have been normalized.

3. Invariance Characteristics of Shadow Codes

This section discusses several characteristics of SCs with rectangular attention regions and the coding procedure described above, which make them especially attractive for invariant pattern recognition.

3.1 Translation and Scaling Invariance

Since it is assumed that the original image has been segmented in such a way that there is only one input pattern of interest in the visual field, the determination of the attention region containing the pattern is a translation-invariant procedure. That is, the attention region will have the same content even if the input pattern is moved in the visual field.☆

The sizes of regions are normalized by dividing the lengths of the shadowed parts by the whole length of each bar. In other words, SCs are also invariant to the actual size (scale) of the input pattern.

3.2 Shadow Codes of Rotated Patterns

Assume that the input pattern is surrounded by a rectangular attention region with the orientation given by the principal axes of inertia. Although the basic orientation of the input pattern, that is, its rotation angle with respect to the coordinate axes, can be determined from the directions of its principal axes of inertia, there may be an error that is a multiple of 90 degrees.

The reason for this can be easily understood by considering handwritten numerals, for instance. The numeral '4' can be written with its width greater than its height, or vice-versa. It can also be upside down and, because of the rectangular symmetry, it would not be possible to tell which side of the rectangular attention region is the "correct" bottom.

However, the SC vector proposed above takes advantage of the symmetries of the attention region. Since the proposed shadow bar frame is symmetric with respect to rotation by multiples of 90 degrees, it is simple to obtain the SC vector of rotated input patterns simply by relabeling the components of the vector, that is, by permutating its components. This is illustrated in **Fig. 4**, which shows how the SCs of a

☆ It may be argued that translation-invariance is a direct consequence of the proposed method for extraction of the rectangular extraction region, and not of the shadow projections themselves. Here, however, the determination of the attention region is seen as part of the representation process, and not as an independent, pre-processing procedure.

pattern rotated by 90, 180, or 270 degrees in a counterclockwise direction can be obtained by relabeling the components of the SC vector of the original pattern.

This property may be exploited for extending the concept of SCs to allow for rotation-invariance. In fact, experiments on the recognition of randomly rotated handwritten digits have been performed by using a scrambler network conceptually based on Widrow's ideas [28], with encouraging results [23].

4. Other Characteristics

4.1 Data Compression

From the description above, it is clear that the proposed SCs are capable of greatly compressing data. For instance, no matter how many pixels the original image contains, the SC vector described in the previous section always consists of a 16-component real vector. Of course, it remains to be demonstrated whether the compressed data retains enough information to allow reliable recognition, but this will be left for the next two sections of this paper.

As a concrete example, consider the experiments described in Sections 5 and 6, where each input image consisted of 2,500 bits (50×50 grid). In this situation, instead of memorizing the 16 real values that make up the shadow codes, a more parsimonious approach is to save the integer shadow lengths before normalization. This results in a total of 19 bytes of information (16 for the shadow lengths and 3 for the bar lengths), or 152 bits, representing a compression rate of 93.9%.

4.2 Generalized Shadow Codes

The central idea behind SCs is the quantization of shadow projections of pixels of the input image on bars of a frame matching the attention region. This is a very general idea, since it is possible to devise many different shapes and orientations for the attention region, other bar frames that differ not only in the shape but also on the number and categories of their bars, as well as other shadow quantization procedures.

In fact, the concept of SCs may be considered as an extension of the idea proposed by Burr [1], originally based on the shapes of luminous bars in LED displays for pocket calculators. However, Burr was not concerned with pattern representation, but only with practical experiments involving pattern recognition by neural nets, and did not consider symmetry and normalization in his frame. Furthermore, although a rectangular frame was used, only the rectangle with the orientation given by the Euclidean axes was considered by Burr, which proved enough for his small-scale experiments.

In the proposed SCs, symmetry plays an important role, as does normalization and the determination of an orientation for the region of attention that is independent of the coordinate system employed by the user. With such features in mind, the authors are studying a few other variations and extensions of SCs, and the results will be reported elsewhere. Variations include the use of a circular region of attention with polar coordinates, and extensions for dealing with gray-scale images.

4.3 Fast Computation

SCs can be computed easily and fast by the proposed method, particularly when the shape of the attention region and the subregions defined by the shadow bars are simple, as in the case of the rectangular attention region with a 16-bar frame. In this case, instead of computing the distance from each pixel to each bar in order to determine the closest bar of each of the three groups, the rectangle may be divided into several bar domain regions determined by the shadow bars, as indicated by the dashed lines in **Fig. 5**, in such a way that the nearest bars can be found by simply determining in which domains a given pixel falls.

For instance, in Fig. 5 (a) there are three dashed lines dividing the rectangle into six horizontal bar domains. A pixel falling within the domain marked "1" projects its horizontal shadow onto the bar b_1; the vertical shadow of the pixel will be projected onto b_3 or b_5, de-

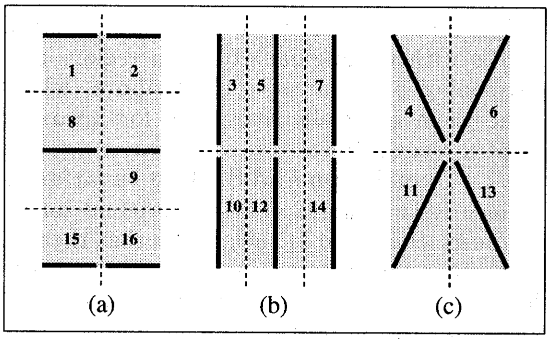


Fig. 5 Bar domains for the (a) horizontal, (b) vertical, and (c) diagonal bars. The horizontal, vertical, and diagonal bars that receive the shadows of a given pixel can be promptly determined by verifying in which bar domains the pixel falls. The numbers indicate corresponding bar labels.

pending on whether the pixel falls in domain "3" or "5"; finally, as the pixel necessarily falls in domain "4", its diagonal shadow is projected onto bar b_4. Since the equations describing the six dashed lines can be easily determined from the coordinates of the vertices of the attention region, it is straightforward to obtain the sets of equations defining each bar domain.

5. Application of SCs to Handwritten Digit Recognition by Neural Networks

In order to investigate their effectiveness, the proposed SCs were applied to the task of recognizing handwritten digits given as thinned black-and-white images.

5.1 Recognition System

The overall dataflow of a general recognition system using SCs is shown in **Fig. 6**, where the data processing is illustrated in a pipeline fashion, with the objectives, input, and output of each stage clearly shown.

5.2 Training and Test Data

To generate data for training and test, about a dozen people were asked to write single digits from '0' to '8' of any size inside a 50 × 50 rectangular grid, using a computer mouse. Since one of the ultimate goals of the current research is to generate a rotation-invariant recognition system, the numeral '9' was not considered in the experiments described here. To distinguish a handwritten '6' from a '9' in a rotation-invariant system, it would be necessary to implement a top-down contextual level in the recognition system, and this is beyond the scope of this paper.

In the experiments described here, 140 patterns for each digit were used for training (total of 1,260 training patterns) while testing was carried out with 100 patterns for each digit. Samples of the training and testing sets of patterns are shown in **Fig. 7**. As the figure suggests, a wide range of possible handwritten digits have been included in both the training and testing pattern sets.

At first, SC vectors as described in Section 2 were generated by using a rectangular attention region with the orientation given by the principal axes of inertia. Preliminary recognition results were good for all patterns, except for those corresponding to the digit '1'. Analysis of the SC values indicated the following problem: since the digit '1' is usually written in such a way that it resembles a single vertical bar, the thinned image of the digit often practically corresponds to one of its principal axes of inertia, as computed through Eq. (5). In consequence, the region of attention shrinks, and almost all the bars become completely shadowed, resulting in shadow values that are very close to each other and to 1.0.

To correct such a problem, a heuristic modification was adopted. Instead of using the "true" principal axes of inertia of the input pattern,

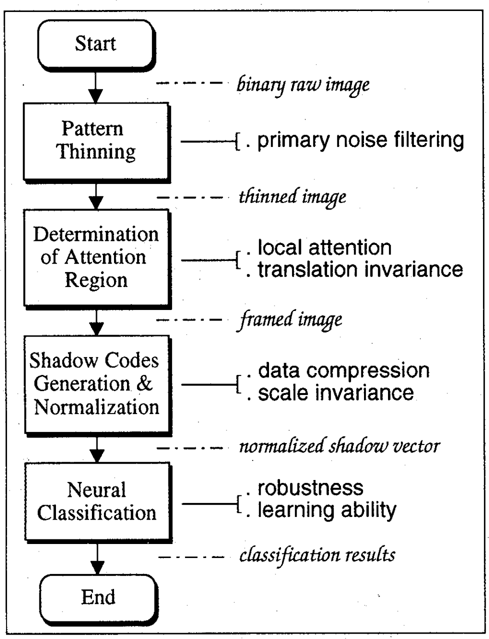


Fig. 6 Dataflow diagram for the overall recognition system.

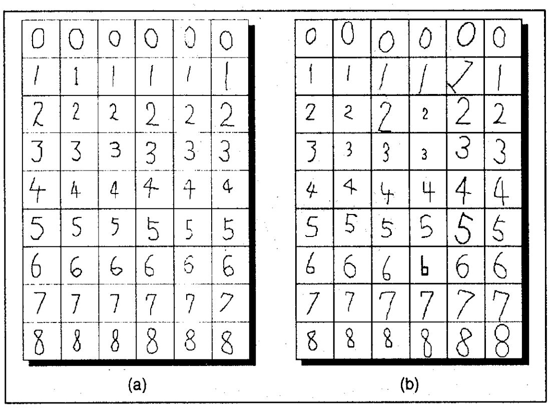


Fig. 7 Samples of training (a) and testing (b) patterns used in the experiments. In the recognition experiments, the test patterns were translated and scaled (amplified or reduced) randomly.

the axes were rotated artificially by 45 degrees, and only then were the SCs generated. It is clear that, even with this modification, the orientation of the region of attention is independent of the user's coordinate system, and does not imply any change to the rest of the SCs computation procedure.

5.3 Neural Network and Conventional Classifiers

Because of their learning capability and robustness, neural networks were chosen for the classification stage of the recognition system. Since the proposed SCs are very compact, small-scale neural networks can be used for the classifier, in contrast to the usual approach, where pixel values are fed directly to the input neurons, which results in large, slow, and difficult-to-train neural networks even for small input patterns [1),6),28)]. When the proposed 16-component SC vecto is used, the number of input nodes of the classifier is also 16. Furthermore, only simple computations are required, and the system is not limited by computational resources, as in Perantonis and Lisboa [16)].

In this paper, the self-organizing map (SOM) proposed by Kohonen [11)] was used for classification, as illustrated in **Fig. 8**. Besides its self-organizing capability, a characteristic of this neural net is that the resulting clusters can be seen topologically. A disadvantage is that, because it is an unsupervised learning paradigm, the results are usually worse than those obtained by supervised models, such as the multilayer perceptron trained by backpropagation [17)]. However, this problem can be rectified by using the Learning Vector Quantization (LVQ) paradigm, a supervised method also developed by Kohonen [11)], to refine the results obtained by SOM. The LVQ paradigm was not employed directly from the beginning, for two reasons: (a) observation of the clusters resulting from applying the shadow codes to an unsupervised neural network offers hints on how much discriminatory information the codes preserve, and (b) the topological map provided by SOM shows graphically the relationships between clusters, indicating the ones most likely to produce fuzzy results.

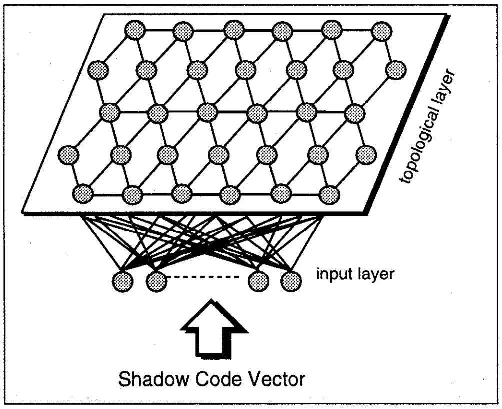


Fig. 8 Self-Organizing Map used as a classifier for the SC vectors. The number of inputs is determined by the dimension of the SC vector, while the number of units and organization of the topological layer must be determined experimentally.

Another advantage of the adopted neural networks is their short training time, which is usually only a few seconds on a workstation. With large databases for training, using more popular neural nets such as the multilayer perceptron can be a frustratingly slow experience. Experiments with a simple nearest-neighbor classifier, here referred to as the conventional classifier, were also carried out to allow comparison of results.

First, in order to test the hypothesis that the selection of the orientation of the rectangular region should be independent of the user's system of coordinates, as well as to verify whether the proposed SCs retained enough information for recognition, experiments using subsets of both the training and test patterns were conducted. The training patterns were used to train a 9×10-node SOM with a learning rate of 0.3 for 5,000 iterations, and the resulting SOM was then fine-tuned by using the LVQ algorithm.

The results are shown in **Table 1**, where Set 1 and Set 2 refer to the training and testing pattern sets, respectively. Furthermore, the columns marked "Euclidean axes" indicate that rectangular attention regions with Euclidean orientation were used, while rectangles with their orientation given by the principal axes of inertia were used in the remaining columns. Overall, high recognition rates were obtained, indicating that SCs compress data greatly while preserving enough information to allow recognition. From Table 1, it is clear that the best results were obtained by using the principal axes of inertia to determine the orientation of the rectangular region, confirming the expectations that the best representation is achieved when only the inherent characteristics of each pattern are considered, independently of the coordinate

Table 1 Recognition results for training (Set 1, 1,260 patterns) and testing (Set 2, 900 patterns) patterns. High recognition rates confirm that SCs preserve enough information to allow reliable pattern classification. The advantage of choosing the rectangular attention region with the orientation given by the pattern's principal axes of inertia is also clear.

Numeral	Recognition Rate %			
	Euclidean Axes		Principal Axes	
	Set 1	Set 2	Set 1	Set 2
0	96.7	100.0	100.0	100.0
1	72.4	42.9	100.0	100.0
2	96.7	85.7	100.0	100.0
3	96.7	85.7	100.0	100.0
4	100.0	100.0	93.3	100.0
5	93.3	100.0	100.0	100.0
6	96.7	100.0	100.0	100.0
7	97.7	100.0	100.0	100.0
8	100.0	100.0	100.0	100.0

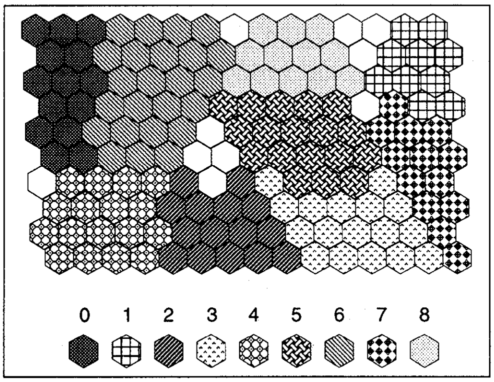


Fig. 9 Training results obtained by a SOM with 150 units. The clusters are well organized, suggesting that SCs preserve enough information for recognition. White units did not "fire" for any training pattern.

Table 2 Average and standard deviation of the bar shadow values for all the training patterns. For convenience, all the values in the table have been multiplied by 100. High values of the standard deviation with respect to the average values indicate diversity of data.

bar #		Numeral 0	1	2	3	4	5	6	7	8
1	avg	75	50	78	76	42	74	41	64	74
1	std	10	10	9	10	16	16	22	17	11
2	avg	50	0	10	20	4	9	4	26	9
2	std	13	0	10	14	8	13	13	20	15
3	avg	73	51	69	58	75	66	73	57	76
3	std	10	9	14	15	18	15	14	16	16
4	avg	36	89	56	46	73	54	54	35	67
4	std	9	6	15	16	18	15	14	16	16
5	avg	42	46	82	85	64	51	26	77	82
5	std	13	12	8	8	24	16	21	17	11
6	avg	19	3	26	41	34	35	23	32	31
6	std	6	4	15	9	19	15	16	12	15
7	avg	55	0	7	27	17	20	16	13	17
7	std	14	0	10	12	14	13	15	23	16
8	avg	40	49	26	36	77	82	81	41	81
8	std	12	11	14	14	17	12	19	14	11
9	avg	41	47	62	83	89	84	82	48	81
9	std	12	10	19	13	11	9	10	12	10
10	avg	56	0	4	2	47	14	52	13	11
10	std	12	1	7	4	18	13	20	14	16
11	avg	19	4	17	13	50	27	46	8	37
11	std	6	4	18	11	14	15	11	8	12
12	avg	39	50	82	29	70	41	66	13	82
12	std	11	10	8	16	20	17	24	14	11
13	avg	37	89	43	36	77	45	52	71	51
13	std	8	8	12	11	13	13	13	10	18
14	avg	74	48	43	73	42	72	75	73	77
14	std	9	9	15	11	24	10	9	14	10
15	avg	54	0	10	5	32	6	50	0	23
15	std	12	0	12	7	29	9	15	2	16
16	avg	75	51	42	66	39	71	78	31	77
16	std	9	9	16	13	17	14	10	10	10

dinate axes. Accordingly, all the other results presented hereafter are for only the rectangular attention region with the orientation given by the principal axes of inertia.

Experiments with several sizes and topologies of SOMs were carried out, using the SCs of all the 1,260 training patterns (140 for each numeral). The best results were achieved with a topological layer containing 15×10 units in a rectangular array, a learning rate of 0.5 and a final neighborhood radius of 5 units. The clusters obtained with this 150-unit SOM after 7,000 learning iterations are shown in **Fig. 9**. It can be seen that the clusters are well organized, suggesting that, although SCs are very compact, they preserve enough information to allow reliable recognition. In the figure, white units did not "fire" for any of the training patterns.

To allow comparison, a simple nearest neighbor classifier was devised. For this purpose, first the average and the standard deviation of the shadow values for each bar and for each pattern category, that is, for each numeral, were determined, as shown in **Table 2**. In Table 2, all values have been multiplied by 100, and the bars have been labeled according to Fig. 2 (a). As expected, the values of the standard deviation were significantly high with respect to the average values, indicating the diversity of the data used for training.

Basically, the operation of the conventional classifier is as follows. For a given input pattern, first compute its shadow bar values s_i, $i = 1, 2, \ldots, 16$ and then, for each category or class c, calculate the distance

Table 3 Recognition results for a nearest-neighbor (conventional) and a SOM-LVQ (neural) classifier. A total of 1,260 and 900 handwritten digits were used for training (Set 1) and test (Set 2), respectively. As expected, the neural classifier was able to cope better with the high diversity of the data used in the experiments.

Numeral	Recognition Rate %			
	Conventional		Neural	
	Set 1	Set 2	Set 1	Set 2
0	98.6	96.0	98.6	99.0
1	100.0	98.0	100.0	99.0
2	96.4	89.0	100.0	97.0
3	96.4	84.0	97.9	94.0
4	87.1	87.0	98.6	97.0
5	92.1	84.0	98.6	92.0
6	91.4	90.0	100.0	97.0
7	94.3	97.0	100.0	95.0
8	92.9	99.0	97.1	98.0
Average	94.4	89.3	99.0	96.4

$$d_c = 0.25\sqrt{\sum_{i=1}^{16} (s_i - \bar{s}_i^c)^2}, \tag{6}$$

where $\bar{s}_i^c$ stands for the average shadow value of the i-th bar for the c-th category, as given in Table 2.

For the problem considered here, it is clear that there are nine categories of interest. After computation of the distances $0 \le d_c \le 1$ for all the nine categories, the input pattern is simply attributed to the category with the smallest d_c value. When this procedure does not permit a sharp conclusion, that is, when the smallest d_c value is too close to or equal to another one, classification is done on the basis of the standard deviation for the shadow values, as shown in Table 2.

The recognition results for both the neural and conventional classifiers, using all the 1,260 training and 900 testing patterns, are shown in **Table 3**, where Set 1 and Set 2 denote the training and test pattern sets, respectively. From the results, it can be seen that the neural classifier obtained by combining SOM and LVQ models performed better than the conventional one. This was expected, because of the large variance of the data used for both training and testing. Once again, high recognition rates confirm the ability of the proposed SC vectors to represent simple binary patterns in a compact way, while retaining essential information to allow reliable recognition.

The mistaken recognition results given by the neural network are shown in **Table 4**. All

Table 4 Misrecognition results (confusion matrix) for the neural network with respect to a set of 100 patterns per numeral. All the errors ocurred in neighboring classes of the SOM shown in Fig. 9.

Input Numeral	Misrecognized as 0	1	2	3	4	5	6	7	8	Subtotal
0	–						1			1
1		–						1		1
2			–	2	1					3
3			2	–		3		1		6
4			1		–		2			3
5	2		1		3	–			2	8
6	2				1		–			3
7		2		3				–		5
8						2			–	2

errors occurred in neighboring clusters of the SOM shown in Fig. 9. Furthermore, it was observed that in all cases in which the network misrecognized an input pattern, the correct answer was the second candidate, that is, the class with the second-smallest recognition error.

Finally, having verified that SCs do preserve information for recognition, it remains to compare them with other character extraction methods. To make such a comparison fair, the other methods should be graphically based and should result in 16-dimensional vectors, as in the case of SCs. Two methods were considered here, namely, the stroke density function (SDF) and the the gross mesh pattern (M feature) [10].

For the SDF method, the rectangular attention region is first divided into eight approximately equal horizontal bands, and for each band the average density of black pixels is taken as a feature. The same procedure is then repeated in the vertical direction. For the M feature, the rectangular region is divided in 16 approximately equal subregions, and for each subregion the ratio between the number of black pixels in the subregion and the total number of black pixels in the whole image is computed. Both procedures result in 16-dimensional, real-valued vectors.

The results for Sets 1 and 2 obtained by using the same SOM-LVQ procedure with a 15×10 grid are shown in **Table 5**. It is clear that the proposed SCs yielded a superior recognition rate for both Sets 1 and 2, indicating that SCs are more successful in preserving feature information. It should also be noted that the results in Table 5 were obtained by beginning from the framed images (see Fig. 6).

Table 5 Comparison of results with those given by two other feature extraction methods using a SOM-LVQ classifier. In all cases, the network input consisted of 16-dimensional vectors. It is clear that SCs yielded a superior recognition rate.

Numeral	Recognition Rate (%)					
	SCs		SDF		M feature	
	Set 1	Set 2	Set 1	Set 2	Set 1	Set 2
0	98.6	99.0	100.0	94.0	98.6	79.0
1	100.0	99.0	97.9	91.0	97.9	89.0
2	100.0	97.0	95.7	93.0	96.4	81.0
3	97.9	94.0	98.6	89.0	95.7	88.0
4	98.6	97.0	97.1	87.0	95.0	76.0
5	98.6	92.0	98.6	91.0	97.9	91.0
6	100.0	97.0	99.3	89.0	93.6	86.0
7	100.0	95.0	98.6	91.0	92.1	85.0
8	97.1	98.0	100.0	90.0	95.7	82.0
Average	99.0	96.4	98.4	90.6	95.9	84.1

6. Conclusion

This paper has introduced a compact and simple coding method for binary visual patterns. The method is based on projections of pixel shadows onto bars of a frame surrounding the pattern to be represented, generating codes that were dubbed shadow codes. For a binary image of any size, the proposed approach easily generates a fixed-length real-valued feature vector, achieving a high degree of data compression.

The proposed SC representation method was applied to the problem of recognizing handwritten numerical characters, using both a neural network and a conventional classifier. High recognition rates were obtained, confirming the effectiveness of the method. Comparison with other graphical feature extraction techniques yielding feature vectors of the same dimension indicates that, although compact, shadow codes succeed in preserving information that can be used for recognition. Furthermore, all the processing steps for computing SCs are fast, and because of the compactness of the code, the proposed approach seems to be suitable for real-time implementation.

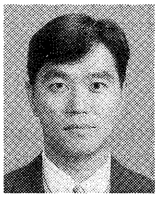
In order to attain higher recognition rates, several improvements are being considered, including the construction of a larger training set, use of different shadow bar frames, and the inclusion of a few more components in the code vector, such as information about the ratio between height and width, pixel density for each bar domain, and so on, and hybrid methods combining the SC method with other feature extraction methods. In addition, current research is focusing on exploiting the symmetries of the SCs to design a neural net recognition system that is not only invariant to translation and scaling, but also to rotation of the input patterns in the visual field. Several approaches are being tested, and results will be reported elsewhere.

References

1) Burr, D.: Experiments on Neural Net Recognition of Spoken and Written Text, *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol.36, No.7, pp.1162–1168 (1988).
2) Casasent, D. and Psaltis, D.: Position, Rotation, and Scale Invariant Optical Correlation, *Applied Optics*, Vol.15, No.7, pp.1795–1799 (1976).
3) Chen, K.-J., Li, K.-C. and Chang, Y.-L.: A System for On-Line Recognition of Chinese Characters, *Comput. Processing of Chinese and Oriental Languages*, Vol.3, pp.309–318 (1988).
4) Fu, K.S.: Pattern Recognition for Automatic Visual Inspection, *IEEE Computer*, Vol.15, No.2, pp.34–40 (1982).
5) Fu, K.S.: *Syntatic Pattern Recognition and Applications*, Prentice-Hall (1982).
6) Fukushima, K. and Wake, N.: Handwritten Alphanumeric Character Recognition by the Neocognitron, *IEEE Trans. Neural Networks*, Vol.2, No.3, pp.355–365 (1991).
7) Gonzalez, R. and Wintz, P.: *Digital Image Processing*, Addison-Wesley (1987).
8) Hagita, H., Umeda, M. and Masuda, I.: Classification of Handprinted Chinese Characters by Three Features, *IEICE Trans. Inf. & Syst.*, Vol.J63-D, No.12, pp.1096–1102 (1980).
9) Hamanaka, M., Yamada, K. and Tsukumo, J.: On-Line Japanese Character Recognition Based on a Flexible Pattern Matching Method Using Normalization-Cooperative Feature Extraction, *IEICE Trans. Inf. & Syst.*, Vol.E77-D, No.7, pp.825–831 (1994).
10) Kimura, F. and Shridhar, M.: Handwritten Numeral Recognition Based on Multiple Algorithms, *Pattern Recognition*, Vol.24, No.10, pp.969–983 (1991).
11) Kohonen, T.: The Self-Organizing Map, *Proc. IEEE*, Vol.78, No.9, pp.1464–1480 (1990).
12) Lam, L., Lee, S.-W. and Suen, C.: Thinning Methodologies – A Comprehensive Survey, *IEEE Trans.Pattern Anal.Machine Intell.*, Vol.14, No.9, pp.869–885 (1992).
13) Mantas, J.: An Overview of Character Recognition Methodologies, *Pattern Recognition*, pp.425–430 (1986).
14) Mori, S., Yamamoto, K. and Yasuda, M.: Research on Machine Recognition of Handprinted

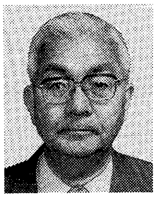
Characters, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol.6, pp.386–405 (1984).
15) Ozaki, H. and Taniguchi, K.: *Image Processing: From the Basics to Applications*, 2nd edition, Kyoritsu (1983).
16) Perantonis, S. and Lisboa, P.: Translation, Rotation, and Scale Invariant Pattern Recognition by High-Order Neural Networks and Moment Classifiers, *IEEE Trans. Neural Networks*, Vol.3, No.2, pp.241–251 (1992).
17) Rumelhart, D.E., Hinton, G. and Williams, R.: Learning Internal Representations by Error Propagation, *Parallel Distributed Processing*, Rumelhart, D. and McClelland, J. (Eds.), Vol.1, pp.318–362, MIT Press (1986).
18) Sato, M. and Kida, H.: A Method of Character Extraction from Printed Documents Including Variable Character Spacing, IEICE Technical Report, No.IE88-138 (1989).
19) Sayre, K.: Machine Recognition of Handwritten Words: A Project Report, *Pattern Recognition*, Vol.5, pp.213–228 (1973).
20) Simon, J.: *From Pixels to Features*, North-Holland (1989).
21) Siy, P. and Chen, C.: Fuzzy Logic for Handwritten Numeral Character Recognition, *Fuzzy Models for Pattern Recognition*, Bezdek, J. and Pal, S. (Eds.), pp.321–325, IEEE Press (1992).
22) Suen, C.Y.: Character Recognition by Computer and Applications, *Handbook of Pattern Recognition and Image Processing*, Young, T. and Fu, K. (Eds.), pp.569–586, Academic Press (1986).
23) Tanomaru, J. and Inubushi, A.: A Compact Representation of Binary Patterns for Invariant Recognition, *Proc. 1995 IEEE Int. Conf. Syst., Man, and Cybernetics*, Vol.5, pp.1550–1555 (1995).
24) Tappert, C.C.: Adaptive On-Line Handwriting Recognition, *Proc. 7th Int. Conf. Pattern Recognition*, pp.1004–1007 (1984).
25) Tappert, C., Suen, C. and Wakahara, T.: The State of the Art in On-Line Handwriting Recognition, *IEEE Trans. Pattern Anal. Machine Intell.*, Vol.12, No.8, pp.787–808 (1990).
26) Tou, J. and Gonzalez, R.: Recognition of Handwritten Characters by Topological Feature Extraction and Multilevel Categorization, *IEEE Trans. Comput.*, Vol.C-21, pp.776–785 (1972).
27) Ward, J. and Phillips, M.: Digitizer Technology: Performance Characteristics and the Effects on the User Interface, *IEEE Comput. Graphics Appli.*, pp.31–44 (1987).
28) Widrow, B., Winter, R. and Baxter, R.: Layered Neural Nets for Pattern Recognition, *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol.36, No.7, pp.1109–1118 (1988).
29) Yoshida, K. and Sakoe, H.: Online Handwritten Character Recognition for a Personal Computer System, *IEEE Trans. Consumer Electron.*, Vol.28, pp.202–209 (1982).
30) Yoshimura, M., Shimizu, T. and Yoshimura, I.: A Recognition System for Japanese Zip Codes Using Arc Features, *IEICE Trans. Info. & Syst.*, Vol.E77-D, No.7, pp.810–816 (1994).
31) Zhang, T. and Suen, C.: A Fast Parallel Algorithm for Thinning Digital Patterns, *Comm. ACM*, Vol.27, No.3, pp.236–239 (1984).

(Received June 12, 1996)
(Accepted November 5, 1997)



Julio Tanomaru was born in Brazil in 1964. He graduated in Electronics Engineering from the Aeronautics Institute of Technology (ITA), Brazil, in 1986, worked for two years in a major Brazilian telecommunications company, and later came to Japan, where he received the M.Eng. and Dr.Eng. degrees in computer sciences from the University of Tokushima, Japan, in 1992 and 1996, respectively. Since 1992 he has been an assistant professor at the at the Department of Information Science and Intelligent Systems at the University of Tokushima. His current research interests include applications of evolutionary algorithms, neural networks and fuzzy reasoning for pattern recognition, nonlinear system idenfication, and combinatorial optimization. He is a member of the IEEE Control and SMC societies, ISCA, IPSJ, SICE, and iSCIe.

Atsushi Inubushi was born in Osaka in 1972 and received the B.Eng. and M.Eng. degrees from the University of Tokushima in 1995 and 1997, respectively, doing research on the recognition of visual patterns using neural networks. Since 1997 he has been working with the Totalizator Systems Engineering Division of Fujitsu Limited, where he is currently involved in the development os information systems for sports competitions.



Yoshizo Takahashi was born in Kobe in 1932. He received his B.Eng. and Dr.Eng. degrees from Kyoto University in 1955 and 1964, respectively. From 1955 to 1977 he was employed at Toshiba Inc., and since 1978 he has been a professor at the Department of Information Science and Intelligent Systems at the University of Tokushima. His current research interests include architectures and algorithms for parallel computers. He is a member IPSJ, IEICE of Japan, Japan Society for Software Science and Technology, ACM, and IEEE Computer Society.